1989/90 Data Book

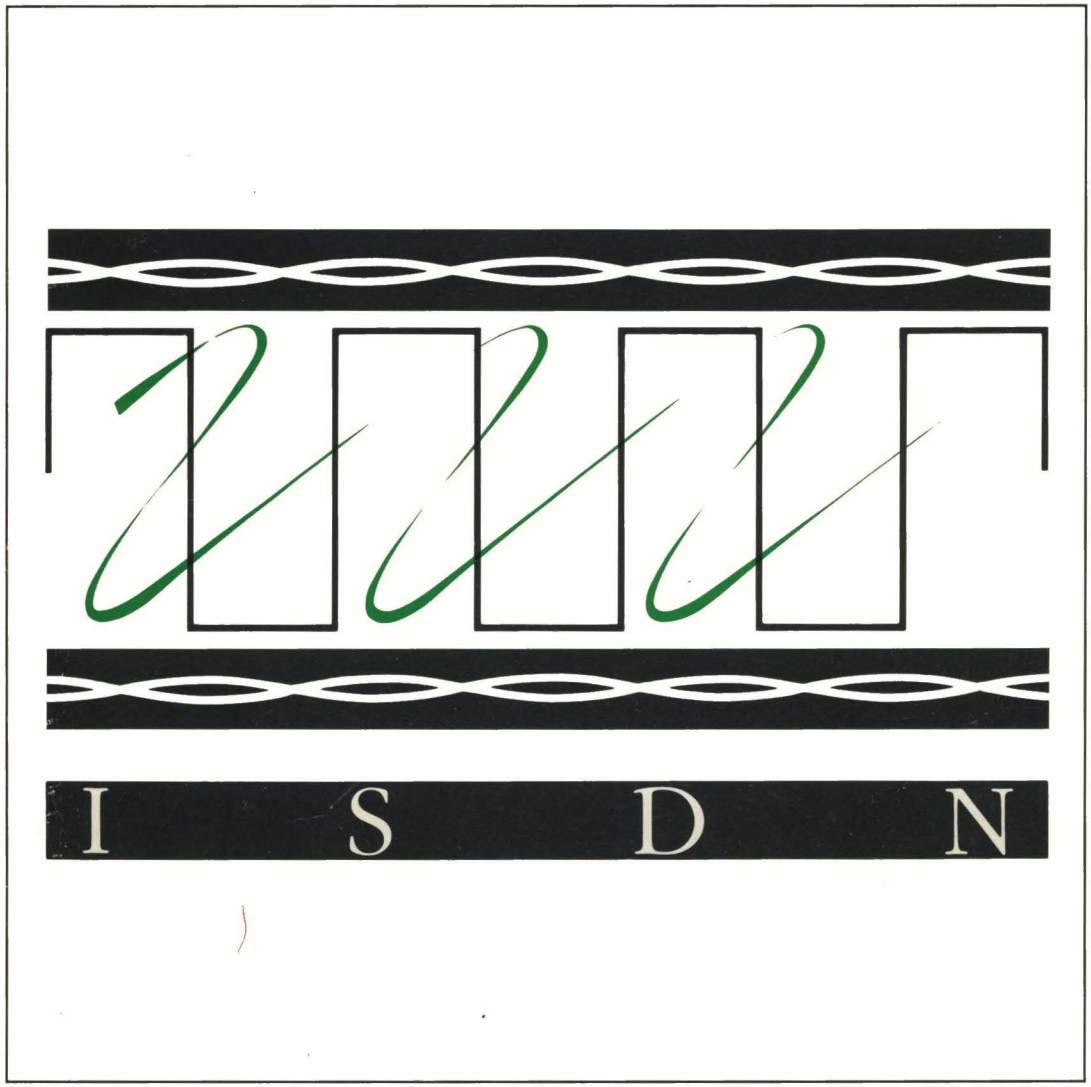
ISDN

CMOS from AMD

ISDN

1989/90 Data Book

Advanced Micro Devices



Advanced
Micro
Devices

ISDN
Data Book

901 Thompson Place, P.O. Box 3453, Sunnyvale, California 94088-3000

(408)732-2400 TWX: 910-339-9280 TELEX: 34-6306

Table of Contents

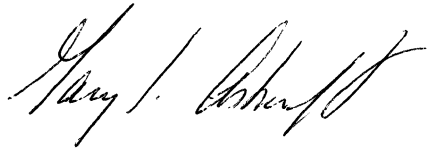
Introduction

ISDN, the Integrated Services Digital Network, is the major component of an evolving World Network™ that is linking telephones and computers around the globe. This development is molding an intricate infrastructure of the world's telephones as well as the multitudes of existing computers and data networks.

Advanced Micro Devices' family of ISDN devices provides comprehensive solutions for the new services that offer simultaneous transmission and reception of voice and data over a common network. AMD is a leading participant in the development and advancement of ISDN as a major worldwide technology.

This Data Book is your comprehensive guide to Advanced Micro Devices' extensive line of ISDN products. Whether your application is terminal equipment, network terminators, line terminators, PABXs, concentrators or central office line cards, our products provide a cost-effective, complete solution with both highly integrated silicon and high-performance software that is compliant with the CCITT standards.

Advanced Micro Devices' commitment in 1982 to becoming a leader in ISDN is a reality today. We continue to apply substantial resources to maintain that leadership position—now and for the future. AMD's technical staff excels at offering support at every level. Our trained field applications engineers, specialists in communications, are backed by system experts in the factory. Call your local AMD sales office or the authorized representative listed in the back of this publication for further information about ISDN.



Gary Ashcraft
Vice President
Communication Products Division

Advanced
Micro
Devices

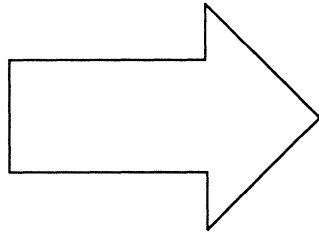


Table of Contents

Chapter 1 **Terminal Products**

Preliminary

Am79C30A/32A

Digital Subscriber Controller (DSC)
ISDN Data Controller (IDC)

Advanced Micro Devices

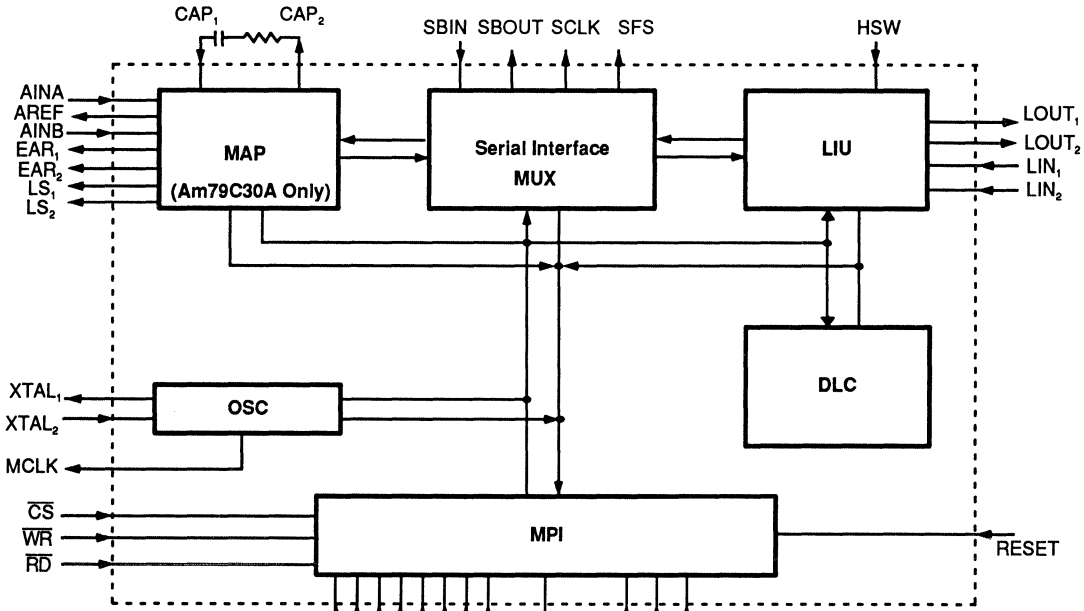
DISTINCTIVE CHARACTERISTICS

- **Combines CCITT I.430 S/T Interface transceiver, D-channel LAPD processor, and audio processor (DSC only) in a single chip**
- **Interrupt-driven microprocessor interface**
- **CMOS technology, TTL compatible**
- **'S' or 'T' Interface Transceiver**

  Level 1 Physical Layer Controller
  Supports point-to-point, short or extended passive bus configurations
  Multiframe support
- **D-channel Processing Capability**
  Flag generation/detection
  CRC generation/checking
  Zero insertion/deletion
  Four 2-byte address detectors
  Random number generation
  8-byte transmit and receive FIFOs
- **Audio Processing Capability (DSC only)**

  Dual audio inputs
  Earpiece and loudspeaker drivers
  Filter/codec with A/mu selection
  Programmable gain and equalization filters
  Programmable sidetone level
  Programmable DTMF, single tone, and ringer tone generation

BLOCK DIAGRAM



LIU Line Interface Unit
MUX Multiplexer
MAP Main Audio Processor
DLC Data Link Controller
MPI Microprocessor Interface
OSC Oscillator and Timing Circuitry

Publication# 09893 Rev. D Amendment /0
Issue Date: March 1989

GENERAL DESCRIPTION

The Am79C30A Digital Subscriber Controller (DSC) and Am79C32A ISDN Data Controller (IDC), shown in the block diagram, provide the Terminal Equipment access to the ISDN. The Am79C30A/32A is compatible with the CCITT I-Series recommendations at the 'S' reference point allowing the user of the device to design TEs which conform to the international ISDN standards.

The Am79C30A/32A provides a 192 kbps full duplex digital path between the TE located in the subscriber's premises and the NT or PABX line card over 4-wires. The Am79C30A/32A separates the bit stream into the B1- (64 kbps), B2- (64 kbps) and D- (16 kbps) channels. The B-channels are routed to different sections of the Am79C30A under user control. The D-channel is partially processed in the Am79C30A/32A and passed to the microprocessor for further processing.

The transmission rate of 192 kbps provides a 48-bit frame every 250 µs for framing and maintenance. The frame structure provides for frame synchronization and multiple terminal contention resolution as described in the CCITT I-series recommendations. Both point-to-point and point-to-multipoint connections are supported.

The Am79C30A can be used as a voice telephone, a digital data terminal, or a voice and data terminal. The Am79C32A can be used as a digital data terminal.

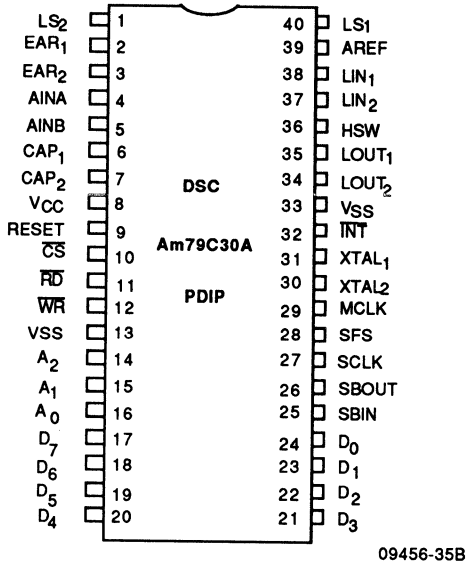
The audio processor in the Am79C30A, shown in the block diagram, uses Digital Signal Processing (DSP) to implement the codec and filter functions. The audio processor interfaces to a speaker, an earpiece, and two separate audio inputs. In the receive and transmit paths the user may program gain or alter the frequency response. The audio processor is not available in the Am79C32A.

A serial port gives the user access to the B-channels of the Am79C30A/32A multiplexer. This serial port may be used by data terminals and provides, with additional circuitry, access to the CCITT 'R' reference point.

The Am79C30A/32A is controlled via an interrupt driven microprocessor bus interface by an external microprocessor. Using this interface, the microprocessor processes the D-channel information and programs the Am79C30A/32A accordingly. This includes programming a multiplexer within the Am79C30A/32A to route the B-channels as specified by the D-channel control information. The microprocessor can interrogate and program the Am79C30A/32A via its mode, status, and error registers.

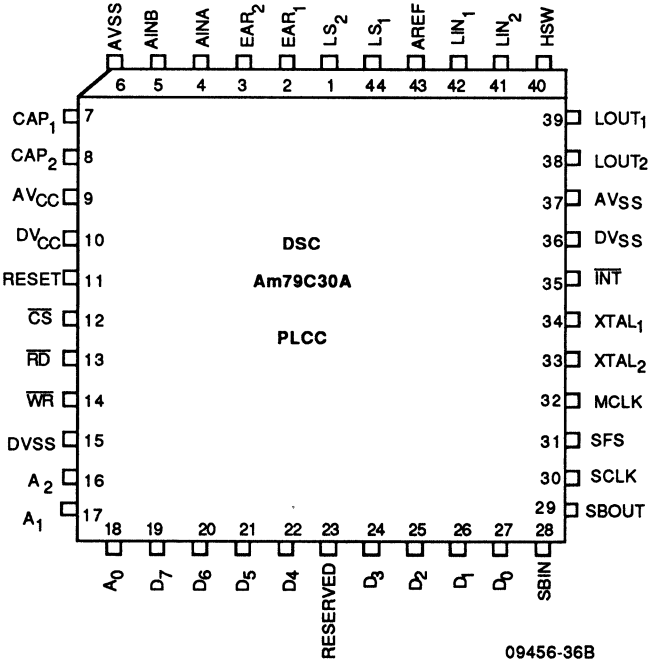
CONNECTION DIAGRAMS
Top View

40 Pin Dual-In-Line Package



09456-35B

44 Pin Plastic Leaded Chip Carrier (PLCC)

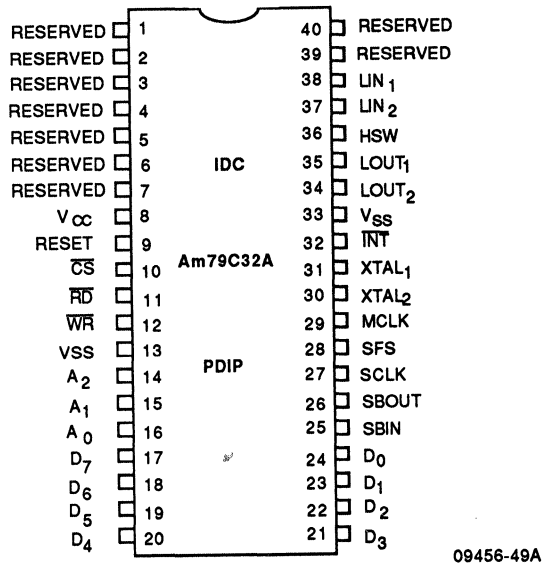


09456-36B

Figure 2. Am79C30A DSC Connection Diagrams

CONNECTION DIAGRAMS
Top View

40 Pin Dual-In-Line Package



44 Pin Plastic Leaded Chip Carrier (PLCC)

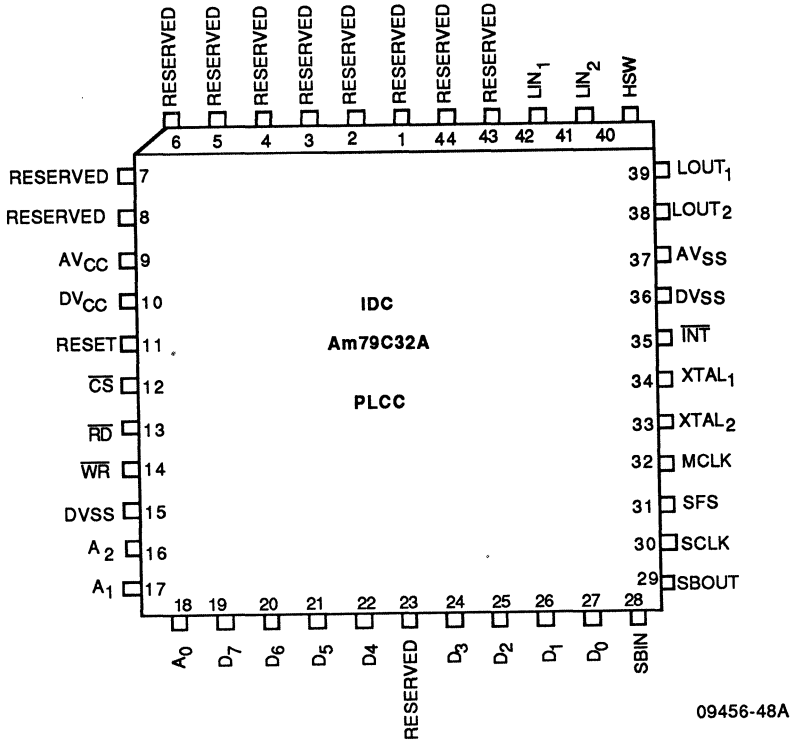
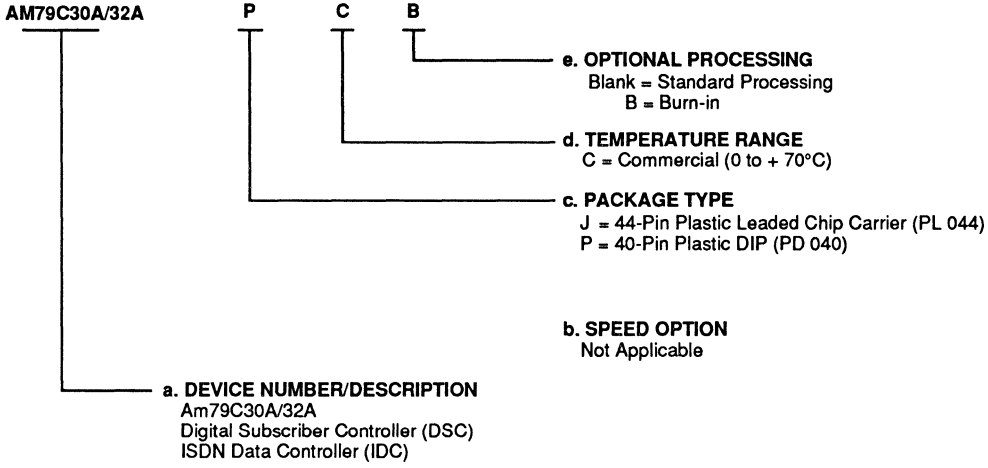


Figure 3. Am79C32A IDC Connection Diagrams

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM79C30A/32A	PC,JC,PCB,JCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

All signal levels are TTL compatible unless otherwise stated.

Line Interface Unit (LIU)

HSW
Hook-Switch (Input)

The HSW signal indicates if the hookswitch is on or off hook. This signal may be generated with a mechanical switch wired to ground with a pull-up resistor to V_{CC}. Any change in the HSW state causes an interrupt.

LIN1, LIN2
Subscriber Line Input (Differential Inputs)

The LIN1 and LIN2 inputs interface to the subscriber ('S' reference point) via an isolation transformer. LIN2 is the positive input, LIN1 is the negative input. These pins are not TTL compatible.

LOUT1, LOUT2
Subscriber Line Output (Differential Outputs)

The LOUT1 and LOUT2 line driver output signals interface to the subscriber line at the 'S' reference point via an isolation transformer and resistors. LOUT2 is the positive 'S' interface driver (that is, sources current during a high mark) and LOUT1 is the negative 'S' interface driver (that is, sources current during low mark). For multipoint applications, all TE's must maintain the same polarity on the 'S' interface. These pins are not TTL compatible.

Multiplexer (MUX)

SBIN
Serial Channel (Input)

The data rate on SBIN is 192 kbps. SBIN consists of three 64 kbps serial channels. Data bytes are received MSB first.

SBOUT
Serial Channel (Output)

The data rate on SBOUT is 192 kbps. SBOUT consists of three 64 kbps serial channels. Data bytes are transmitted MSB first.

SCLK
Serial Clock (Output)

SCLK is a 192 kbps synchronization clock which defines the position of the serial bits in the SBOUT and SBIN channels. Data at the SBIN input must be valid on the rising edge of SCLK. The data on the SBOUT pin changes on the falling edge of SCLK. SCLK powers up tri-stated, and is enabled when a MUX connection is programmed.

SFS
Serial Channel Frame Sync. (Output)

SFS is an 8 kHz signal which identifies the beginning of each frame by a low to high transition. The192 kbps data stream on SBIN and SBOUT is referenced to SFS. SFS powers up tri-stated, and is enabled when a MUX connection is programmed.

Main Audio Processor (MAP)

All MAP pins are analog, and hence not TTL compatible.

AINA, AINB
Analog (Inputs)

These analog inputs allow for two separate analog (audio) inputs to the transmit path of the codec/ filter. Input signals on either of these pins must be referenced to AREF.

AREF
Analog Reference (Output)

This is a nominal 2.4 V reference voltage output for biasing the analog inputs. Note that AREF is only available when the MAP is active.

CAP1, CAP2
Capacitor/Resistor (CAP1, Input; CAP2, Output)

An external resistor and capacitor are connected in series between these pins. These components are needed for the integrator in the Analog to Digital Converter (ADC).

EAR1, EAR2
Earpiece Interface (Differential Outputs)

EAR1 and EAR2 are the outputs from the receive path of the filter codec. These differential outputs can directly drive 600 ohms.

LS1, LS2
Loudspeaker Interface (Differential Outputs)

LS1 and LS2 are push-pull outputs which can directly drive a 50 ohm loudspeaker.

Microprocessor Interface (MPI)

A2-A0
Address Line (Inputs)

A2, A1, and A0 signals select source and destination registers for read and write operations on the data bus.

$\overline{CS}$
Chip Select (Input)

$\overline{CS}$ must be low to read or write to the Am79C30A/32A. Data transfer occurs over the bidirectional data lines (D7-D0).

D7-D0
Data Bus (Bidirectional with High Impedance State)

The eight bidirectional data bus lines are used to exchange information with the microprocessor. D0 is the least significant bit (LSB) and D7 is the most significant bit (MSB). A high on the data bus line corresponds to a

logic '1' and low corresponds to a logic '0'. These lines act as inputs when both $\overline{WR}$ and $\overline{CS}$ are active and as outputs when both $\overline{RD}$ and $\overline{CS}$ are active. When $\overline{CS}$ is inactive or both $\overline{RD}$ and $\overline{WR}$ are inactive, the D0-D7 pins are in a high impedance state.

$\overline{INT}$
Interrupt (Output)

An active low output on the $\overline{INT}$ pin informs the external microprocessor that the Am79C30A/32A needs interrupt service. $\overline{INT}$ is updated once every 125 µs. The $\overline{INT}$ pin remains active until the Interrupt Register (IR) is read or the Am79C30A/32A is reset.

RESET
Reset (Input)

Reset is an active high signal which causes the Am79C30A/32A to immediately terminate its present activity and initialize to the reset condition. When reset returns low, the Am79C30A/32A enters the idle mode.

$\overline{RD}$
Read (Input)

The active low read signal is conditioned by $\overline{CS}$ and indicates that internal information is to be transferred onto the data bus. A number of internal registers are user accessible. The contents of the accessed register are transferred onto the data bus after the high to low transition of the $\overline{RD}$ input.

$\overline{WR}$
Write (Input)

The active low write signal is conditioned by $\overline{CS}$ and indicates that external information on the data bus is to be transferred to an internal register. The contents of the data bus are loaded on the low to high transition of the $\overline{WR}$ input.

Oscillator (OSC)

MCLK
Master Clock (Output)

The MCLK output is available for use as the system clock for the microprocessor. It is derived from the 12.288 MHz crystal via a programmable divider in the Am79C30A/32A which provides the following MCLK output frequencies: 12.288, 6.144, 4.096, and 3.072 MHz.

XTAL1, XTAL2
External Crystal (Output/Input)

XTAL1 and XTAL2 are connected to an external parallel resonant crystal for the on-chip oscillator. XTAL2 can also be connected to an external source instead of a crystal, in which case XTAL1 should be left disconnected. The frequency must be 12.288 MHz, ± 80 ppm.

Power Supply Pins

PLCC Packages

AVcc	+5V analog power supply ±5% (PLCC only)
AVss	Analog ground (PLCC only)
DVss	Digital ground (PLCC only)
DVcc	+5V digital power supply, ±5% (PLCC only)

DIP Packages

Vcc	+5V power supply, ±5% (DIP only)
Vss	Ground (DIP only)

Note: For best performance, decoupling capacitors should be installed between Vcc and Vss as close to the chip as possible. Do not use separate supplies for analog and digital power and ground connections.

1

OPERATIONAL DESCRIPTION

In order to specify the functions of the Am79C30A/32A, the device has been divided into blocks as shown in the block diagram. Each of the blocks listed below is defined separately in the Functional Description.

LIU Line Interface Unit
MUX Multiplexer
MAP Main Audio Processor (Am79C30A only)
DLC Data Link Controller
MPI Microprocessor Interface
OSC Oscillator and Timing Circuitry

User Accessible Registers/Buffers

The microprocessor interface is used to program and control the operation of the Am79C30A/32A. The following registers/buffers are user accessible in each of the blocks listed above and are described in the respective Functional Description Sections.

Registers	No.	Mnemonic
Line Interface Unit (LIU)		
LIU Status Register	1	LSR
LIU Priority Register	1	LPR
LIU Mode Registers	2	LMR
Multiframe Register	1	MF
Multiframe S-Bit/Status Buffer	1	MFSB
Multiframe Q-Bit Buffer	1	MFQB
Multiplexer (MUX)		
MUX Control Registers	4	MCR
Main Audio Processor (MAP) (Am79C30A only)		
X Filter Coefficient Registers	16	X
R Filter Coefficient Registers	16	R
GX Gain Coefficient Registers	2	GX
GR Gain Coefficient Registers	2	GR
GER Gain Coefficient Registers	2	GER
Sidetone Gain Coefficient Registers	2	STGR
Frequency Tone Generator Registers	2	FTGR
Amplitude Tone Generator Registers	2	ATGR
MAP Mode Registers	2	MMR
Data Link Controller (DLC)		
First Received Byte Address Registers	4	FRAR
Second Received Byte Address Registers	4	SRAR
Transmit Address Register (16 bit)	1	TAR
D-channel Receive Byte Limit Register (16 bit)	1	DRLR
D-channel Receive Byte Count Register (16 bit) (2 byte FIFO)	1	DRCR
D-channel Transmit Byte Count Register (16 bit)	1	DTCR
Random Number Generator Registers	2	RNGR
D-channel Mode Registers	4	DMR
D-channel Status Registers	2	DSR
Address Status Register (2 byte FIFO)	1	ASR
D-channel Error Register (2 byte FIFO)	1	DER
Microprocessor Interface (MPI)		
Initialization Register	1	INIT
Command Register	1	CR
Interrupt Register	1	IR
Data Register	1	DR
D-channel Transmit Buffer (8 byte FIFO)	1	DCTB
D-channel Receive Buffer (8 byte FIFO)	1	DCRB
Bb Transmit Buffer	1	BBTB
Bb Receive Buffer	1	BBRB
Bc Transmit Buffer	1	BCTB
Bc Receive Buffer	1	BCRB

Note: See the Microprocessor Interface section for register addressing.

Initialization

The initialization procedure is controlled via the Initialization Register (INIT) which is accessed by the microprocessor as defined in the Microprocessor Interface section. This Initialization Register (INIT) has the following format:

Initialization Register (INIT), Read/Write

Bit # 7	6	5	4	3	2	1	0	Control	Function
X	X	X	X	X	X	0	0	Power Mode Selection	Idle Mode (default)
X	X	X	X	X	X	0	1		Active Mode (voice & data)
X	X	X	X	X	X	1	0		Active Mode (Data only)
X	X	X	X	X	X	1	1		Reserved
X	X	X	X	X	0	X	X	Interrupt Selection	Enable $\overline{INT}$ pin (default)
X	X	X	X	X	1	X	X		Disable $\overline{INT}$ pin
X	X	0	0	0	X	X	X	Clock Divider Selection	Divide by 2 (default)
X	X	0	0	1	X	X	X		Divide by 1
X	X	0	1	0	X	X	X		Divide by 4
X	X	0	1	1	X	X	X		Divide by 2
X	X	1	0	0	X	X	X		Divide by 3
X	X	1	0	1	X	X	X		Divide by 2
X	X	1	1	0	X	X	X		Divide by 2
X	X	1	1	1	X	X	X		Divide by 2
X	1	X	X	X	X	X	X	Abort Selection	Receive abort
X	0	X	X	X	X	X	X		No Receive abort (default)
1	X	X	X	X	X	X	X		Transmit abort
0	X	X	X	X	X	X	X		No Transmit abort (default)

Reset

The Am79C30A/32A can be reset by driving the reset pin high. When power is first supplied to the Am79C30A/32A, a reset must be asserted. This initializes the Am79C30A/32A to its default values as defined in the subsequent sections. After reset, the Am79C30A/32A enters the idle mode.

Idle Mode Operation

To conserve power the Am79C30A/32A can be placed in the idle mode. This can be done by either asserting the RESET signal or by clearing bits 0 and 1 in the INIT register. When the Am79C30A/32A is in the idle mode, and there is no activity on any of the external interfaces, the state of the output pins are:

Pin Name	State following RESET	Idle Mode
D7-D0	High impedance	High impedance
MCLK	6.144 MHz	As programmed (see Initialization section)
$\overline{INT}$	Logical '1'	Logical '1' can be driven low by internal conditions explained below
SBOUT	High impedance	High impedance
SFS	High impedance	8 kHz
SCLK	High impedance	192 kHz period
LS1, LS2, EAR1, EAR2, AREF	High impedance	High impedance
LOUT1, LOUT2	High impedance	High impedance

SFS and SCLK will be high impedance after a reset, and will become active upon the first write to any MCR registers.

The minimum idle mode power consumption is met if bits 3 and 5 of the INIT register are set to a logical '0' and bit 4 is set to a logical '1', thus ensuring that the MCLK output is 3.072 MHz.

In the idle mode the MAP, SBP, MUX and DLC blocks are disabled and the OSC, MPI and LIU blocks are active. The programmed state of the Am79C30A/32A is maintained (hence the programmable registers retain their data) and the Am79C30A/32A can be programmed while in the idle mode. The DLC read only registers are reset. The microprocessor can access any of the Am79C30A/32A registers but the idle mode power consumption specification does not apply when the microprocessor is accessing the Am79C30A/32A.

If the receiver is enabled, the LIU responds automatically to activity when the Am79C30A/32A is in the idle mode. Thus, if the receiver is enabled and the LIU either detects activity on the 'S' interface or receives an activation request from the microprocessor, the LIU follows the CCITT activation procedure and interrupts the microprocessor when activation has been achieved. The default value of LIU Mode Register 1 bit 6 disables the receiver. The Am79C30A/32A idle mode can be terminated in the following three ways:

1) When the LIU achieves activation on the 'S' interface, the Am79C30A/32A interrupts the microprocessor (depending on the programmed state of the Am79C30A/32A's LIU Mode Register 1). This interrupt "wakes up" the microprocessor if it is in its idle mode. The microprocessor then programs the Am79C30A/32A to the proper mode via bits 0 and 1 in the INIT register. (See LIU section.)

2) The Am79C30A/32A has a hookswitch (HSW) detector with debounce circuitry to accommodate the requirements of a mechanical switch. When the HSW changes state, an interrupt is generated to the microprocessor if the HSW interrupt is enabled. The microprocessor can then read the status of the HSW via the LIU Status Register and power-up the Am79C30A/32A as required.

3) At any time, the microprocessor can terminate the idle mode by writing to bit 0 and bit 1 of the INIT Register.

MCLK Options

Upon reset the Am79C30A/32A crystal frequency of 12.288 MHz is divided by two and output as the MCLK. The microprocessor can alter this output frequency through the INIT register as per its timing requirements. It should be noted that the MCLK is never stopped. The frequencies available at the MCLK output pin are:

1) 12.288 MHz (divide by 1)
2) 6.144 MHz (divide by 2)
3) 4.096 MHz (divide by 3)
4) 3.072 MHz (divide by 4)

Receive and Transmit Abort Commands

The microprocessor has the option via INIT register bits 6 and 7 to abort the receive and transmit D-channel packets. When the microprocessor sets one of these bits, the Am79C30A/32A aborts the respective operation. The frame abort sequence is defined in greater detail later. (See Data Link Controller section.)

Interrupt Handling

The Am79C30A/32A generates either no interrupt or only one interrupt every 125 µsec. Once asserted, $\overline{\text{INT}}$ remains active until the microprocessor responds by interrogating the Am79C30A/32A's Interrupt Register. Reading the Interrupt Register in response to an activated $\overline{\text{INT}}$ pin deactivates the $\overline{\text{INT}}$ pin, and clears the interrupt register.

If an interrupt causing event occurs while the Interrupt Register (IR) is being read by the microprocessor, the effect of the event is held until the microprocessor has completed it's read cycle. A reset clears all interrupt causing conditions.

The $\overline{\text{INT}}$ pin remains active until the Interrupt Register is read or reset is driven active. The Interrupt Register is cleared by reading the Interrupt Register, or by reset.

Interrupt Register (IR), Read Only

The Interrupt Register has the following format:

Bit	Interrupt generated/action required		Interrupt Mask
0	D-channel transmit threshold interrupt/load D-channel Transmit Buffer		DMR1 bit 0
1	D-channel receive threshold interrupt/read D-channel Receive Buffer		DMR1 bit 1
2	D-channel status interrupt/read DSR1		
	Source	Cause	
	DSR1 bit 0	Valid Address (VA) or End of Address (EOA)	DMR3 bit 0
	DSR1 bit 1	When a closing flag is received or a receive error occurs	DMR1 bit 3
	DSR1 bit 6	When a closing flag is transmitted	DMR3 bit 1
3	D-channel error interrupt/read DER and DSR2 bit 2		
	Source	Cause	
	DER bit 0	Current received packet has been aborted	DMR2 bit 0
	DER bit 1	Non-integral number of bytes received	DMR2 bit 1
	DER bit 2	Collision abort detected	DMR2 bit 2
	DER bit 3	FCS error	DMR2 bit 3
	DER bit 4	Overflow error	DMR2 bit 4
	DER bit 5	Underflow error	DMR2 bit 5
	DER bit 6	Overrun error	DMR2 bit 6
	DER bit 7	Underrun error	DMR2 bit 7
	DSR2 bit 2	Receive packet lost	DMR3 bit 6
4	Bb or Bc byte available or buffer empty interrupt/read or write Bb or Bc buffers		MCR4 bit 3
5	LIU status interrupt/read LSR		
	Source	Cause	
	LSR bit 3	Change of state to F3	LMR2 bit 3
	LSR bit 4	Change of state from/to F7	LMR2 bit 6
	LSR bit 5	Change of state from/to F8	LMR2 bit 4
	LSR bit 7	HSW change of state	LMR2 bit 5
6	D-Channel status interrupt/read DSR2		
	Source	Cause	
	DSR2 bit 0	Last byte of received packet	DMR3 bit 2
	DSR2 bit 1	Receive byte available	DMR3 bit 3
	DSR2 bit 3	Last byte transmitted	DMR3 bit 4
	DSR2 bit 4	Transmit buffer available	DMR3 bit 5
7	Multiframe interrupt/read MFSB		
	Source	Cause	
	MFSB bit 5	S-data available	MF bit 1
	MFSB bit 6	Q-bit buffer empty	MF bit 2
	MFSB bit 7	Multiframe change of state (In/Out of sync.)	MF bit 3

Bits 0, 1 and 4 of the IR, if set, advise the microprocessor that the respective buffer is ready for reading or writing. If bit 0 is set due to an empty buffer, the D-Channel Transmit Buffer must be serviced within 375 µs. If bit 1 is set and the D-Channel Receive Buffer is full, the buffer must be serviced within 425 µs. This is to prevent erroneous data transfers causing transmitter underrun and receiver overrun errors. If bit 4 is set then the Bb or Bc buffers must be accessed within 122.4 µs. This is to prevent erroneous data transfers. Only one interrupt is used to signal accessibility for both B-channels of the 'S' interface. Since the data transfer must occur synchronously to the 'S' interface, any data access to either Bb or Bc or both must be made within the 122.4 µs limit. Note: even though only a single interrupt is issued, either or both 'S' interface B-channels must be serviced.

IR bits 2, 3, 5, 6, and 7, if set, indicate that a bit has been set in the associated status or error register. All of the interrupts generated by the Am79C30A/32A can be individually disabled.

DMR1, DMR2, DMR3, LMR2, MCR4, and MF control the mask conditions which affect the $\overline{INT}$ pin. The $\overline{INT}$ pin is activated only by interrupts which are not disabled and the Interrupt Register reflects the status of enabled interrupts. The $\overline{INT}$ pin can be disabled by setting INIT register bit 2 to a logical '1'.

The Am79C30A/32A has facilities that allow the microprocessor to read the status registers (status update is inhibited during status read) or the IR at any time during functional operation.

Test Facilities

The Am79C30A/32A provides three types of test facilities:

1) The ability to suppress the framing signals from the Am79C30A/32A to the network termination. This simulates a loss of synchronization and is programmed into LMR1 bits 2 and 3.

2) Three types of D-channel loopbacks, two on the 'S' interface for maintenance purposes and one on the microprocessor interface for local testing. Normally, the Am79C30A/32A compares its transmitted D-channel bits to its received E-channel bits. Any difference is interpreted as an error condition which halts transmission. However, in these loopbacks, the Am79C30A/32A ignores any such difference, thereby allowing transmission to proceed. These loopback test modes are selected via LMR2 bits 0,1 and 2. See LIU Registers section for more details.

3) Two MAP loopbacks, one analog and one digital. The analog loopback enables an analog signal to be sent in on AINA or AINB and out on EAR1/EAR2 or LS1/LS2. This loopback is selected by connecting Ba to Ba in the multiplexer. The digital loopback tests the digital portion of the MAP by routing the data from the MUX through to the interpolator and then feeding it into the decimator. The ADC and DAC are disconnected from the loopback path. This loopback is selected by setting bit 7 of MMR1 to a logical '1' (See Figure 9).

FUNCTIONAL DESCRIPTION

Microprocessor Interface (MPI)

The Am79C30A/32A can be connected to any general purpose 8-bit microprocessor via the MPI. The MCLK from the Am79C30A/32A can be used as the clock for the microprocessor. The MPI is an interrupt driven interface containing all the circuitry necessary for access to the internal programmable registers, status registers, coefficient RAM, and transmit/receive buffers.

MPI External Interface

The MPI has the following external connections:

Name	Direction	Function
D7-D0	Bidirectional	Data Bus
A2, A1, & A0	Inputs	Address Line
$\overline{RD}$	Input	Read enable
$\overline{WR}$	Input	Write enable
$\overline{CS}$	Input	Chip Select
RESET	Input	Initialization
$\overline{INT}$	Output	Interrupt

Directly Accessed Registers/Buffers

Register	Mnemonic
Command Register	CR
Interrupt Register	IR
Data Register	DR
D-channel Status Register 1	DSR1
D-channel Error Register (2 byte FIFO)	DER
D-channel Transmit Buffer (8 byte FIFO)	DCTB
D-channel Receive Buffer (8 byte FIFO)	DCRB
Bb Transmit Buffer	BBTB
Bb Receive Buffer	BBRB
Bc Transmit Buffer	BCTB
Bc Receive Buffer	BCRB
D-channel Status Register 2	DSR2

The 8-bit bidirectional data bus (D7-D0) is used to communicate with these registers. The selection of which register(s) is (are) accessed is controlled by the $\overline{CS}$, $\overline{RD}$, $\overline{WR}$, A2, A1, and A0 signals from the microprocessor to the Am79C30A/32A in the following manner (see Figure 4):

Register Selection

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A2	A1	A0	Register(s) Accessed
0	1	0	0	0	0	Command Register (CR), write only
0	0	1	0	0	0	Interrupt Register (IR), read only
0	1	0	0	0	1	Data Register (DR), write only
0	0	1	0	0	1	Data Register (DR), read only
0	0	1	0	1	0	D-channel Status Register 1 (DSR1), read only
0	0	1	0	1	1	D-channel Error Register (DER), read only (2 byte FIFO)
0	1	0	1	0	0	D-channel Transmit Buffer (DCTB), write only (8 byte FIFO)
0	0	1	1	0	0	D-channel Receive Buffer (DCRB), read only (8 byte FIFO)
0	1	0	1	0	1	Bb channel Transmit Buffer (BBTB) write only
0	0	1	1	0	1	Bb channel Receive Buffer (BBRB), read only
0	1	0	1	1	0	Bc channel Transmit Buffer (BCTB), write only
0	0	1	1	1	0	Bc channel Receive Buffer (BCRB), read only
0	0	1	1	1	1	D-channel Status Register 2 (DSR2), read only
1	X	X	X	X	X	No access (X = logical '0' or '1')

Note: The $\overline{RD}$ and $\overline{WR}$ signals must never both be low under normal operating conditions.

Indirectly Accessed Registers

To read from or write to any of the indirect data registers a command is first written to the Command Register (CR). Depending on the command, one (or more) data byte(s) is then transferred to or from the selected register(s) as follows:

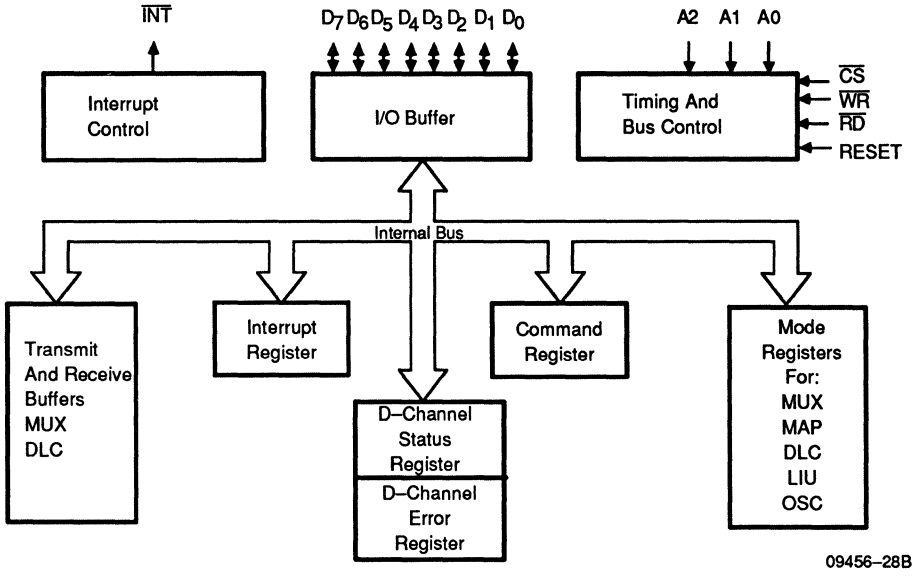
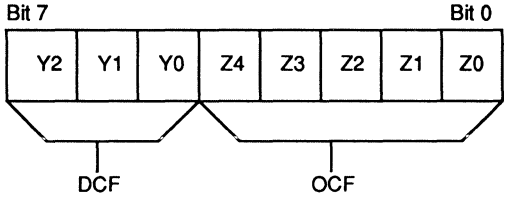


Figure 4. MPI Block Diagram

Command Register (CR), Write only

The Command Register (CR) is used to hold the index for the indirectly accessed registers. The CR is one byte wide and divided into two fields, the destination code field (DCF) and the operational code field (OCF):



The 'Y' bits in the DCF determine to which block the OCF is related. The 'Y' bits are defined below:

Am79C30A/32A block	Y2	Y1	Y0
RESERVED	0	0	0
INIT	0	0	1
MUX	0	1	0
MAP (Am79C30A only)	0	1	1
DLC	1	0	0
LIU	1	0	1
RESERVED	1	1	0
RESERVED	1	1	1

The 'Z' bits of the OCF contain the operation code used to address a specific register within a DCF block. For each DCF there is a different set of OCFs. Each of the OCFs and their associated data registers are defined in the following sections. When the OCF indicates a read or write operation is possible, then the appropriate access is achieved by asserting either the read or write signals to the Am79C30A/32A.

Registers within certain groups can be quickly accessed by using internal circuitry which automatically increments the indirect address (index value) contained in the OCF. The CR is first loaded with the index value, then the data bytes are transferred in sequence between the Am79C30A/32A and the microprocessor via the DR. For example, operation number 5 in the MPI-LIU definition allows operations 2 to 4 to be performed sequentially without reloading the CR. Whenever the CR register is loaded, any previous commands are automatically terminated.

In the following tables the "bytes transferred" numbers next to the OCFs are the number of bytes which are read or written to the DR after the CR has been loaded.

MPI-INIT Definition

The INIT register is used by the microprocessor to select the MCLK output frequency and the power-up/idle states of the Am79C30A/32A. This register is accessed via the 'Z' bits in the CR as follows:

INIT OPERATION (DCF = 001)	OCF Z4	Z3	Z2	Z1	Z0	Bytes Transferred
R/W INIT register	0	0	0	0	1	1

MPI-LIU Definition

The LIU contains the following registers:

LIU Registers	No.	Mnemonic
LIU Status Register	1	LSR
LIU Priority	1	LPR
LIU Mode Registers	2	LMR1, LMR2
Multiframe Register	1	MF
Multiframe S-bit Buffer	1	MFSB
Multiframe Q-bit Buffer	1	MFQB

These registers are accessed via the 'Z' bits in the CR as follows:

LIU OPERATION (DCF = 101)	OCF Z4	Z3	Z2	Z1	Z0	Bytes Transferred
1. Read LSR	0	0	0	0	1	1
2. R/W LPR	0	0	0	1	0	1
3. R/W LMR1	0	0	0	1	1	1
4. R/W LMR2	0	0	1	0	0	1
5. Perform Operations 2-4	0	0	1	0	1	3
6. Read/Write MF	0	0	1	1	0	1
7. Read MFSB	0	0	1	1	1	1
8. Write MFQB	0	1	0	0	0	1

MPI-MUX Definition

The MUX contains four microprocessor read/write control registers MCR1, MCR2, MCR3, and MCR4. These registers are accessed via the 'Z' bits in the CR as follows:

MUX OPERATION (DCF = 010)	OCF Z4	Z3	Z2	Z1	Z0	Bytes Transferred
1. R/W MCR1	0	0	0	0	1	1
2. R/W MCR2	0	0	0	1	0	1
3. R/W MCR3	0	0	0	1	1	1
4. R/W MCR4	0	0	1	0	0	1
5. Perform Operations 1-4	0	0	1	0	1	4

When more than one byte is being transferred due to OCF command 5, MCR1 is always accessed first, followed by MCR2, then MCR3, and finally MCR4.

MPI-MAP Definition (Am79C30A only)

The MAP registers are:

MAP Registers	No.	Mnemonic
X Filter Coefficient Registers	16	X
R Filter Coefficient Registers	16	R
GX Gain Coefficient Registers	2	GX
GR Gain Coefficient Registers	2	GR
GER Gain Coefficient Registers	2	GER
Sidetone Gain Coefficient Registers	2	STGR
Frequency Tone Generator Registers	2	FTGR
Amplitude Tone Generator Registers	2	ATGR
MAP Mode Registers	2	MMR

These registers are accessed via the OCF 'Z' bits in the CR.

MAP OPERATION (DCF = 011)	OCF Z4	Z3	Z2	Z1	Z0	Bytes Transferred
1. R/W X Filter Coefficients	0	0	0	0	1	16
2. R/W R Filter Coefficients	0	0	0	1	0	16
3. R/W GX Gain Coefficients	0	0	0	1	1	2
4. R/W GR Gain Coefficients	0	0	1	0	0	2
5. R/W GER Gain Coefficients	0	0	1	0	1	2
6. R/W STG Coefficients	0	0	1	1	0	2
7. R/W FTGR1 & FTGR2	0	0	1	1	1	2
8. R/W ATGR1 & ATGR2	0	1	0	0	0	2
9. R/W MMR1	0	1	0	0	1	1
10. R/W MMR2	0	1	0	1	0	1
11. Perform Operations 1-10	0	1	0	1	1	46

MPI-DLC Definition

The indirectly accessible DLC registers are:

Registers	No.	Mnemonic
First Received Byte Address Registers	4	FRAR
Second Received Byte Address Registers	4	SRAR
Transmit Address Register (16 bit)	1	TAR
D-channel Receive Byte Limit Register (16 bit)	1	DRLR
D-channel Receive Byte Count Register (16 bit) (2 word FIFO)	1	DRCR
D-channel Transmit Byte Count Register (16 bit)	1	DTCR
Random Number Generator Registers	2	RNGR
D-channel Mode Registers	4	DMR
Address Status Register (2 byte FIFO)	1	ASR

DLC OPERATION (DCF = 100)	OCF Z4	Z3	Z2	Z1	Z0	Bytes Transferred
1. R/W FRAR1, 2 & 3	0	0	0	0	1	3
2. R/W SRAR1, 2 & 3	0	0	0	1	0	3
3. R/W TAR	0	0	0	1	1	2
4. R/W DRLR	0	0	1	0	0	2
5. R/W DTCR	0	0	1	0	1	2
6. R/W DMR1	0	0	1	1	0	1
7. R/W DMR2	0	0	1	1	1	1
8. Perform Operations 1-7	0	1	0	0	0	14
9. Read DRCR	0	1	0	0	1	2
10. R/W RNGR1	0	1	0	1	0	1
11. R/W RNGR2	0	1	0	1	1	1
12. R/W FRAR4	0	1	1	0	0	1
13. R/W SRAR4	0	1	1	0	1	1
14. R/W DMR3	0	1	1	1	0	1
15. R/W DMR4	0	1	1	1	1	1
16. Perform Operations 12-15	1	0	0	0	0	4
17. Read ASR	1	0	0	0	1	1

Microprocessor Access Guide for Indirect Registers

Block Accessed	Operation#	Register Name	Y & Z HEX Coding	Byte Sequence
MPI-INIT	1	INIT	21	One byte transferred
MPI-LIU	1	LSR	A1	One byte transferred
	2	LPR	A2	One byte transferred
	3	LMR1	A3	One byte transferred
	4	LMR2	A4	One byte transferred
	5	Perform 2-4	A5	LPR, LMR1, LMR2
	6	MF	A6	One byte transferred
	7	MFSB	A7	One byte transferred
	8	MFQB	A8	One byte transferred
MPI-MUX	1	MCR1	41	One byte transferred
	2	MCR2	42	One byte transferred
	3	MCR3	43	One byte transferred
	4	MCR4	44	One byte transferred
	5	Perform 1-4	45	MCR1, MCR2, MCR3, MCR4
MPI-MAP	1	X Coeff.	61	h0 LSB,h0 MSB..h7 MSB
(see Fig. 9)	2	R Coeff.	62	h0 LSB,h0 MSB..h7 MSB
	3	GX Coeff.	63	LSB, MSB
	4	GR Coeff.	64	LSB, MSB
	5	GER Coeff.	65	LSB, MSB
	6	STG Coeff.	66	LSB, MSB
	7	FTGR1, FTGR2	67	FTRG1, FTGR2
	8	ATGR1, ATGR2	68	ATGR1, ATGR2
	9	MMR1	69	One byte transferred
	10	MMR2	6A	One byte transferred
	11	Perform 1-10	6B	46 bytes loaded 1-10
MPI-DLC	1	FRAR1, 2 & 3	81	FRAR1, FRAR2, FRAR3
	2	SRAR1, 2 & 3	82	SRAR1, SRAR2, SRAR3
	3	TAR	83	LSB, MSB
	4	DRLR	84	LSB, MSB
	5	DTCR	85	LSB, MSB
	6	DMR1	86	One byte transferred
	7	DMR2	87	One byte transferred
	8	Perform 1-7	88	14 bytes loaded 1-7
	9	DRCR	89	LSB, MSB
	10	RNGR1 (LSB)	8A	One byte transferred
	11	RNGR2 (MSB)	8B	One byte transferred
	12	FRAR4	8C	One byte transferred
	13	SRAR4	8D	One byte transferred
	14	DMR3	8E	One byte transferred
	15	DMR4	8F	One byte transferred
	16	Perform 12-15	90	FRAR4, SRAR4, DMR3, DMR4
	17	ASR	91	One byte transferred

*Note: LSB=Least Significant Byte; MSB=Most Significant Byte

Line Interface Unit (LIU)

The LIU connects to the four wire 'S' interface through a pair of isolation transformers, one for the transmit and one for the receive direction, as shown in Figure 5.

The receiver section of the LIU consists of a differential receiver, circuitry for bit timing recovery, circuitry for detecting high marks and low marks, and a frame recovery circuit for frame synchronization. The receiver converts the received pseudo-ternary coded signals to binary before delivering them to the other blocks of the Am79C30A/32A. It also performs collision detection (E and D bit comparison) per the CCITT recommendations so that several TEs can be connected to the same 'S' interface.

The transmitter consists of a binary to pseudo-ternary encoder and a differential line driver which meets the CCITT recommendations for the 'S' interface.

The Am79C30A/32A can establish multiframe synchronization, receive S-bits, and transmit Q-bits synchronized to the received frame.

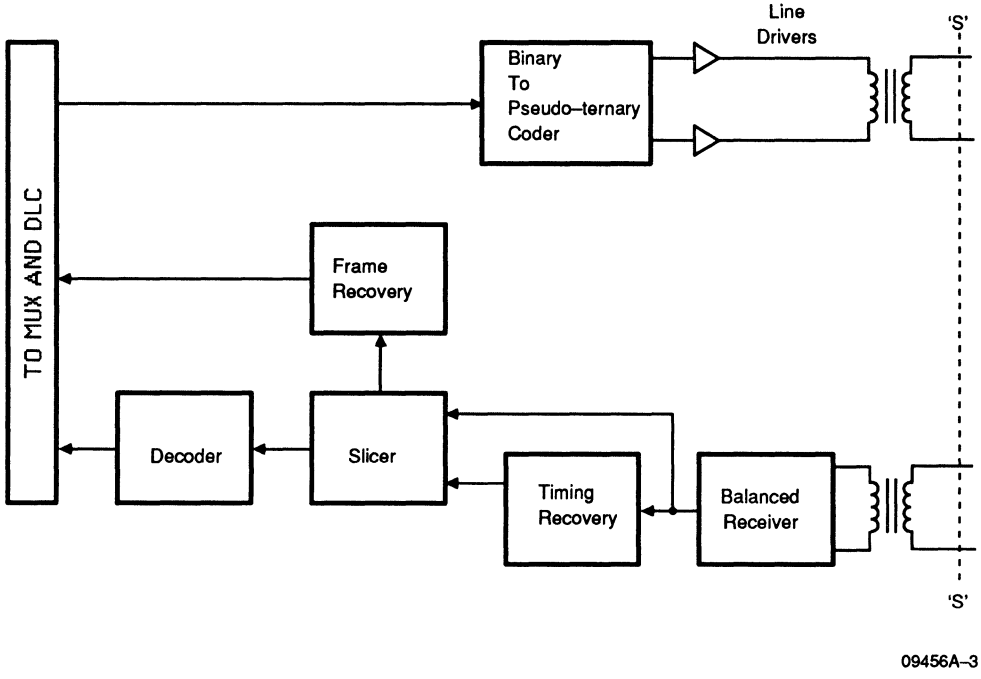


Figure 5. LIU Block Diagram

External Interface

The LIU can be connected to both point-to-point and point-to-multipoint configurations at the CCITT 'S' reference point. The point-to-point configuration consists of one TE connected to the NT or PABX line card. The point-to-multipoint configuration can have multiple TEs connected to one NT.

Line Code

Pseudo-ternary coding is used for both transmitting and receiving over the 'S' interface. In this type of coding, a binary '1' is represented by a space (zero voltage) and a binary '0' is represented by a high mark or a low mark. Two consecutive binary '0's are represented by alternate marks to reduce DC offset on the line. A mark followed, either immediately or separated by spaces, by a mark of the same polarity, is defined as a code violation. Code violations are used to identify the boundaries of the frame.

Frame Structures

In both transmit and receive directions, the bits are grouped into frames of 48 bits each. The frame structure is identical for both point-to-point and point-to-multipoint configurations. Each frame transmitted, at 4 kHz, consists of several groups of bits.

Multiframing

If multiframing is enabled, the Am79C30A/32A recognizes and establishes multiframe synchronization based on the monitoring of the FA- (Q-bit control) and M- (M-bit control) bits. The Am79C30A/32A also receives and compiles S-bits, and transmits Q-bits synchronized to the received frame.

Establishment of Multiframe Synchronization

When the 'enable multiframe synchronization' bit (bit 0 of the Multiframe Register) is set and the LIU is in state F7, the LIU monitors the FA- (Q-bit control) and M- (M-bit control) bits. When three consecutive multiframes with the M-bits and FA bits set as defined in Figure 2.4 are received, the 'multiframe synchronized' bit (bit 7 of the Multiframe Register) and 'multiframe change of state' bit (bit 7 of the Multiframe S-bit/Status Buffer) are set. Note that S-bit data is received, compiled and transferred to the user after attaining synchronization at the start of the next multiframe.

S-bit Reception

After multiframe synchronization has been requested and established, the microprocessor can read the Multiframe S-bit/Status Buffer (MFSB) once the 'S-data available' bit (bit 5 of MFSB) is set. The 'S-data available' bit is set to a logical '1' when the Am79C30A/32A has received five S-bits (one S-bit per 'S' interface frame) synchronized to the setting of the FA-bit to a logical '1' and transferred them into the MFSB. Once the 'S-bit available' bit is set, the MFSB must be accessed within 1.25 ms or succeeding S-data will be lost.

Transmission of Q-bits

The microprocessor can load the Multiframe Q-bit Buffer (MFQB) once the 'Q-bit buffer empty' bit (bit 6 of the Multiframe S-bit/Status Buffer) is set. The 'Q-bit buffer empty' bit is set to a logical '1' at reset or when data that has been written to the Multiframe Q-bit Buffer is transferred to the LIU. The 'Q-bit buffer empty' bit is cleared to a logical '0' when the Multiframe S-bit/Status Buffer is read. After multiframing has been requested and established, the Am79C30A/32A transfers the data written into the Q-bit Register to the LIU synchronized to the multiframe irrespective of the receipt of valid Q-control bits. If the microprocessor does not reload the Q-bit Register for retransmission, the Q-bit pattern is repeated in the next multiframe.

If multiframing is enabled but multiframe synchronization is not established, the LIU transmits the value loaded in MFQB bit 4 in all Q-bits. The default value of MFQB bit 4 is a logical '0' which satisfies the CCITT recommendations. When synchronization is achieved, the contents of MFQB bits 3 to 0 are transmitted according to Figure 6.

Loss of Multiframe Synchronization

The Am79C30A/32A continuously monitors the FA- (Q-bit control) and the M-bits to assure multiframe synchronization. Once multiframe synchronization is established, multiframe synchronization is lost if three consecutive invalid multiframes are received or the LIU exits state F7 or multiframing is disabled. When loss of multiframe synchronization occurs, bit 7 of the Multiframe Register is set to a logical '0' and bit 7 of the Multiframe S-bit/Status Buffer is set to a logical '1'. The Am79C30A/32A also terminates the reception of S-bits and transmission of Q-bits until multiframing synchronization is re-established.

LIU Registers

The LIU contains the following registers:

Registers	No.	Mnemonic
LIU Status Register	1	LSR
LIU Priority Register	1	LPR
LIU Mode Registers	2	LMR1, LMR2
Multiframe Register	1	MF
Multiframe S-Bit/Status Register	1	MFSB
Multiframe Q-Bit Buffer	1	MFQB

LIU Status Register (LSR), Read Only

The LSR has the following format:

Bit	Logical '1'	Generates Interrupt
2,1,0	Binary values 000 through 110 represent the LIU activation circuitry's current state (F2 through F8 respectively) Bit 2 is MSB.	No
3	Change of state to F3	If LMR2 bit 3 = 1
4	Change of state from/to F7	If LMR2 bit 6 = 1
5	Change of state from/to F8	If LMR2 bit 4 = 1
6	HSW state	No
7	HSW change of state	If LMR2 bit 5 = 1

When the microprocessor reads the LSR bits 3, 4, 5, and 7 are cleared. The other bits retain the current status of the LIU. Bits 0 to 2 are defined such that state F2 (see CCITT I.430 state matrix tables) is coded as 0, F3 as 1, F4 as 2, and so on, where bit 0 is the LSB. The LIU interrupts the microprocessor via bit 4 of the LSR when activation has been achieved (that is, when the LIU moves to state F7 upon receipt of INFO 4). During reset the LSR is zero. Subsequent to removing the reset, the Am79C30A/32A goes to state F2 and reflects the condition of the HSW as well as the condition of 'S' interface after the receiver has been enabled.

LIU D-channel Priority Register (LPR), Read/Write

The LPR contains the priority level for D-channel access. Its default value after reset is zero.

The D-channel access procedure of the Am79C30A/32A uses the priority level programmed in the LPR. The priority mechanism defined by the CCITT I-series recommendations is fully implemented if the LPR is programmed via the microprocessor to conform to the priority class of the layer 2 frame to be transmitted.

The LPR has 16 possible programmable priority levels. The priority levels are numbered from 0 to 15. Priority level 0 corresponds to counting 8 ones in the echo channel, priority level 1 corresponds to counting 10 ones in the echo channel, and priority level 2 corresponds to counting 12 ones, etc. The DSC automatically handles transitions between the programmed priority level "n" and the associated odd value n + 1. The priority is incremented following a successfully transmitted packet, and decremented when the higher count has been satisfied.

The LPR has the following format:

Bits	Description
3,2,1,0	D-Channel access priority level. Bit 0 is LSB.
7,6,5,4	Not used, read logical '0'

LIU Mode Register (LMR1), Read/Write

LMR1 is defined as follows:

Bit	Logical '1'	Logical '0' (default value)
0	Enable B1 transmit	Disable B1 transmit
1	Enable B2 transmit	Disable B2 transmit
2	Disable F transmit	Enable F transmit
3	Disable FA transmit	Enable FA transmit
4	Activation request	No activation request
5	Go from F8 to F3	No transition
6	Enable receiver/transmitter	Disable receiver/transmitter
7	Not used, reads logical '0 '	Not used, reads logical '0'

The F and FA bits in LMR1 (bits 2 and 3) should be enabled during the activation procedure so that the Am79C30A/32A can respond with INFO 3.

LMR1 bit 4 is used to transfer the signals 'PH-AR' and 'Expiry of Timer' from the microprocessor to the LIU (see CCITT I.430 state diagram — activation request). 'PH-AR' is defined as bit 4 being a logical '1' and 'Expiry of Time' is defined as the transition of bit 4 from a logical '1' to a logical '0'. This bit must not be set until the LIU, as reflected in the LSR, is in state F3, F6 or F7 and the receiver has been enabled for a minimum of 250 µs.

LMR1 bit 6 is primarily used to disable the receiver when the terminal does not require access to the 'S' interface signals. This bit is cleared by reset and must be written to logical '1' in order to receive activation from the 'S' interface, or to request activation.

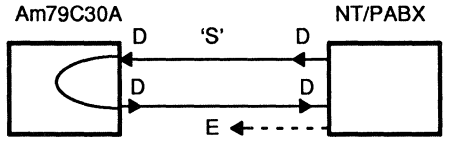
LIU Mode Register 2 (LMR2), Read/Write

LMR2 is used to select the following operations:

Bit	Logical '1'	Logical '0' (default value)
0	D-channel loopback at Am79C30A/32A enable	D-channel loopback at Am79C30A/32A disable
1	D-channel loopback at LIU enable	D-channel loopback at LIU disable
2	D-channel back-off disable	D-channel back-off enable
3	F3 change of state interrupt enable	F3 change of state interrupt disable
4	F8 change of state interrupt enable	F8 change of state interrupt disable
5	HSW interrupt enable	HSW interrupt disable
6	F7 change of state interrupt enable	F7 change of state interrupt disable
7	Not used, reads logical '0'	Not used, reads logical '0'

The three D-channel loopback controls defined in LMR2 bits 0,1,and 2 are explained below:

Bit 0, D-channel loopback at Am79C30A/32A enable:

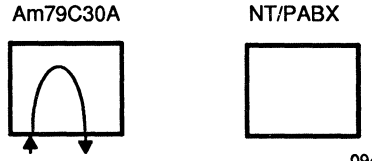


09456-4A

This remote loopback is provided for maintenance purposes from the NT's perspective. The NT transmits D-channel bits to the Am79C30A/32A where they are internally looped (with the Data Link Controller) and transmitted back to the NT. The incoming D-channel data can be accessed by the microprocessor; however, the microprocessor cannot send data on the outgoing D-channel.

Any difference between the transmitted D-channel bits and the received E-channel bits to/from the Am79C30A/32A (normally detected as an error which halts the transmission) is ignored thereby allowing the transmission to continue.

Bit 1, D-channel loopback at LIU enable:

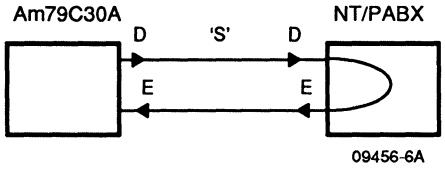


09456A-5

This local loopback is provided for local testing. Data on the incoming D-channel is ignored. The data from the microprocessor is processed by the DLC and then

looped back to the microprocessor in addition to being output to the 'S' interface. The looped back data stream is processed by the DLC.

Bit 2, D-channel back off disable:



This loopback is provided for maintenance purposes from the TE's perspective. The Am79C30A/32A transmits D-channel bits to the NT where they are looped and transmitted back to the Am79C30A/32A in the E-channel. The operation is normal except differences between the D- and E-channels do not halt the transmission.

Multiframe Register (MF), Read/Write

Bit	Logical '1'	Logical '0' (Default value)
0	Enable multiframe synchronization	Disable multiframe synchronization
1	Enable 'S-data available' interrupt	Disable interrupt
2	Enable 'Q-bit buffer empty' interrupt	Disable interrupt
3	Enable 'Multiframe change of state' interrupt	Disable interrupt
6,5,4	Not used, read logical '0'	Not used, read logical '0'
7	Multiframe synchronized (read only)	Multiframe not synchronized (read only)

Multiframe S-bit/Status Buffer (MFSB), Read Only

Bit	Description	Generates Interrupt
0	S1	No
1	S2	No
2	S3	No
3	S4	No
4	S5	No
5	S-data available	If MF bit 1 = 1
6	Q-bit buffer empty	If MF bit 2 = 1
7	Multiframe change of state	If MF bit 3 = 1

The MFSB is cleared when read or upon reset.

Multiframe Q-bit Buffer (MFQB), Write Only

Bit	Description
0	Q1 (default = 1)
1	Q2 (default = 1)
2	Q3 (default = 1)
3	Q4 (default = 1)
4	Q-bit value when multiframing enabled but synchronization not achieved (default = 0)
5,6,7	Not used

S-Bit Channel Structure

Frame Number	NT-to-TE Q Control Bit (F/A)	NT-to-TE M Bit (M)	NT-to-TE S Bit (S)
1	One	One	SC11
2	Zero	Zero	SC21
3	Zero	Zero	SC31
4	Zero	Zero	SC41
5	Zero	Zero	SC51
6	One	Zero	SC12
7	Zero	Zero	SC22
8	Zero	Zero	SC32
9	Zero	Zero	SC42
10	Zero	Zero	SC52
11	One	Zero	SC13
12	Zero	Zero	SC23
13	Zero	Zero	SC33
14	Zero	Zero	SC43
15	Zero	Zero	SC53
16	One	Zero	SC14
17	Zero	Zero	SC24
18	Zero	Zero	SC34
19	Zero	Zero	SC44
20	Zero	Zero	SC54
1	One	One	SC11
2	Zero	Zero	SC21
etc.			

Frame Number	NT-to-TE F_A Bit (Q Control Bit)	TE-to-NT F_A Bit (Q Bit)	NT-to-TE M Bit (M Bit)
1	One	Q1	One
2	Zero	Zero	Zero
3	Zero	Zero	Zero
4	Zero	Zero	Zero
5	Zero	Zero	Zero
6	One	Q2	Zero
7	Zero	Zero	Zero
8	Zero	Zero	Zero
9	Zero	Zero	Zero
10	Zero	Zero	Zero
11	One	Q3	Zero
12	Zero	Zero	Zero
13	Zero	Zero	Zero
14	Zero	Zero	Zero
15	Zero	Zero	Zero
16	One	Q3	Zero
17	Zero	Zero	Zero
18	Zero	Zero	Zero
19	Zero	Zero	Zero
20	Zero	Zero	Zero
1	One	Q1	One
2	Zero	Zero	Zero
etc.			

Figure 6. Multiframe Structures

Multiplexer (MUX)

The MUX connects to the Line Interface Unit (LIU), the Main Audio Processor (MAP), the Microprocessor Interface (MPI) and the Serial Port (SP).

The MUX selectively routes the 64 kbps full duplex B-channels of the LIU, MAP, SP, or MPI to the appropriate destinations depending on the control bits set in the MUX control registers. The D-channel data is routed to the DLC directly from the LIU and does not pass through the MUX.

Serial Port External Interface

The MUX has a serial port (SP) external interface providing three full duplex 64 kbps B-channels Bd, Be, and Bf. The timing for the SP is shown in Figure 7. The signal names for the SP pins are:

Pin	Description
SCLK	Serial Clock (192 kHz)
SFS	Serial Frame Sync.
SBOUT	Serial data out (the MSB is transmitted first)
SBIN	Serial data in (the MSB is expected to be received first)

The SCLK and SFS signals are synchronized to the received 'S' interface frame. If there is no 'S' interface frame synchronization, the SCLK and SFS signals are free running at 192 kHz and 8 kHz respectively.

MUX Internal Interface

The logical channels available at the MUX are shown in Figure 8. They are:

From/to the LIU channels B1 and B2
From/to the MAP channel Ba
From/to the MPI channels Bb and Bc
From/to the SP channels Bd, Be and Bf

For any specific application, the MUX can be programmed by the microprocessor to route any three B-channel ports to any other three B-channel ports. Programmable bidirectional bit reversal is provided for both of the MPI data channels Bb and Bc.

MUX Control Registers 1, 2, and 3 (MCR1, MCR2 & MCR3), Read/Write

The MUX can support three bidirectional paths. The contents of the MUX Control Registers MCR1, MCR2, and MCR3 direct the flow of data between the eight MUX logical B-channels (See Figure 8). These three MCRs are programmed to connect any two B-channel ports together by writing the appropriate channel code into an MCR. These MCRs have the same format, where bits 7 to 4 indicate port 1 and bits 3 to 0 indicate port 2. In each of these three MCR registers the following channel codes are used for both ports 1 and 2:

MCR Register Channel Codes

Code	Channel
0000	No connection (default value)
0001	B1 (LIU)
0010	B2 (LIU)
0011	Ba (MAP)
0100	Bb (MPI)
0101	Bc (MPI)
0110	Bd (SP channel 1)
0111	Be (SP channel 2)
1000	Bf (SP channel 3)

For example, to connect B1(LIU) with Bb (MPI) and B2 (LIU) with Ba (MAP), the contents of the MCRs would be:

Register	Port1 / Port2 7 6 5 4 3 2 1 0	Channel connection
MCR1	0 0 0 1 0 1 0 0	B1 (LIU)<—>Bb (MPI)
MCR2	0 0 1 0 0 0 1 1	B2 (LIU)<—>Ba (MAP)
MCR3	0 0 0 0 0 0 0 0	No connect<->No connect

Therefore, in this example, MCR1 provides a data link from the 'S' interface and MCR2 sets up a voice connection across the 'S' interface.

To loopback a channel, the same channel code is used for port 1 and port 2. For example, to loopback B1, B2 and Ba the MCRs would be:

Register	Port1 / Port2 7 6 5 4 3 2 1 0	Channel connection
MCR1	0 0 0 1 0 0 0 1	B1 (LIU) Loopback
MCR2	0 0 1 0 0 0 1 0	B2 (LIU) Loopback
MCR3	0 0 1 1 0 0 1 1	Ba (MAP) Loopback

MCR3 has higher priority than MCR2 which has higher priority than MCR1. If multiple connections are made to the same port, the data from the connecting ports in the highest priority MCR will overwrite the data from the connecting port in the lower priority MCR, for example:

Register	Port1 / Port2 7 6 5 4 3 2 1 0	Channel connection
MCR1	0 0 0 0 0 0 0 0	No connect
MCR2	0 0 0 1 0 1 0 0	B1 (LIU)<—>Bb (MPI)
MCR3	0 1 0 0 0 0 1 1	Bb (MPI)<—>Ba (MAP)

The final data transfers are:

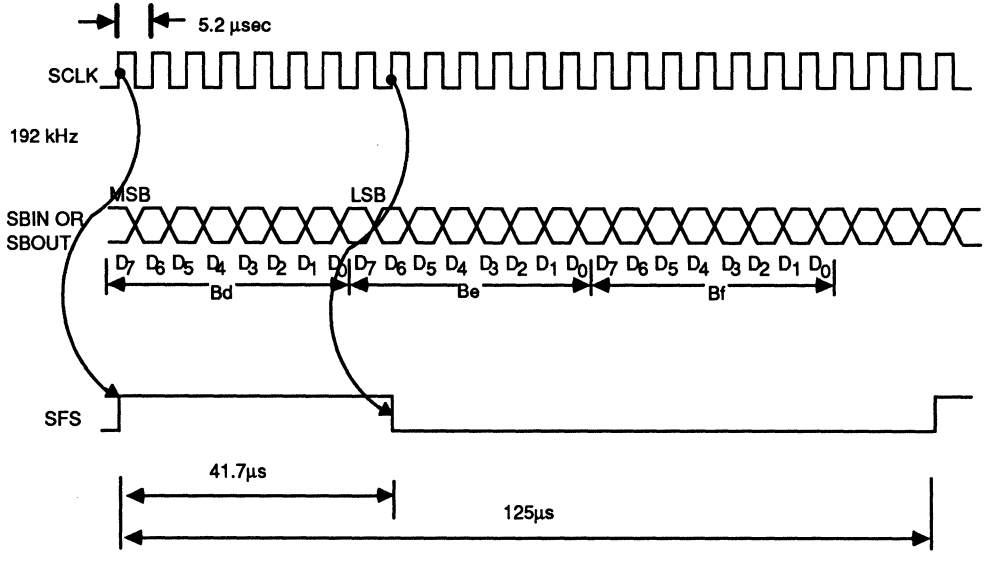
B1 (LIU) receives Bb (MPI),
Ba (MAP) receives Bb (MPI),
Bb (MPI) receives Ba (MAP).

Therefore, the data transfer from B1(LIU) to Bb(MPI) is lost in the arrangement proposed in MCR2.

MUX Control Register 4 (MCR4), Read/Write

The MUX Control Register 4 (MCR4) can prevent interrupt generation by masking the output of IR bit 4. MCR4 has the following format:

Bit	Logical '1'	Logical '0' (default value)
2,1,0	Reserved, must be set to logical '0'	Reserved, must be set to logical '0'
3	Enable 'Bb or Bc channel byte available' interrupt (IR bit 4)	Disable interrupt
4	Reverse bit order of Bb (LSB transmitted/received first)	No Bb bit reversal (MSB transmitted/received first)
5	Reverse bit order of Bc (LSB transmitted/received first)	No Bc bit reversal (MSB transmitted /received first)
6	Reserved, must be set to logical '0'	Reserved, must be set to logical '0'
7	Reserved, must be set to logical '0'	Reserved, must be set to logical '0'



09456–7B

Note: SBIN is sampled on the rising edge of SCLK, SBOUT is changed on the falling of SCLK.

Figure 7. Serial Port Timing

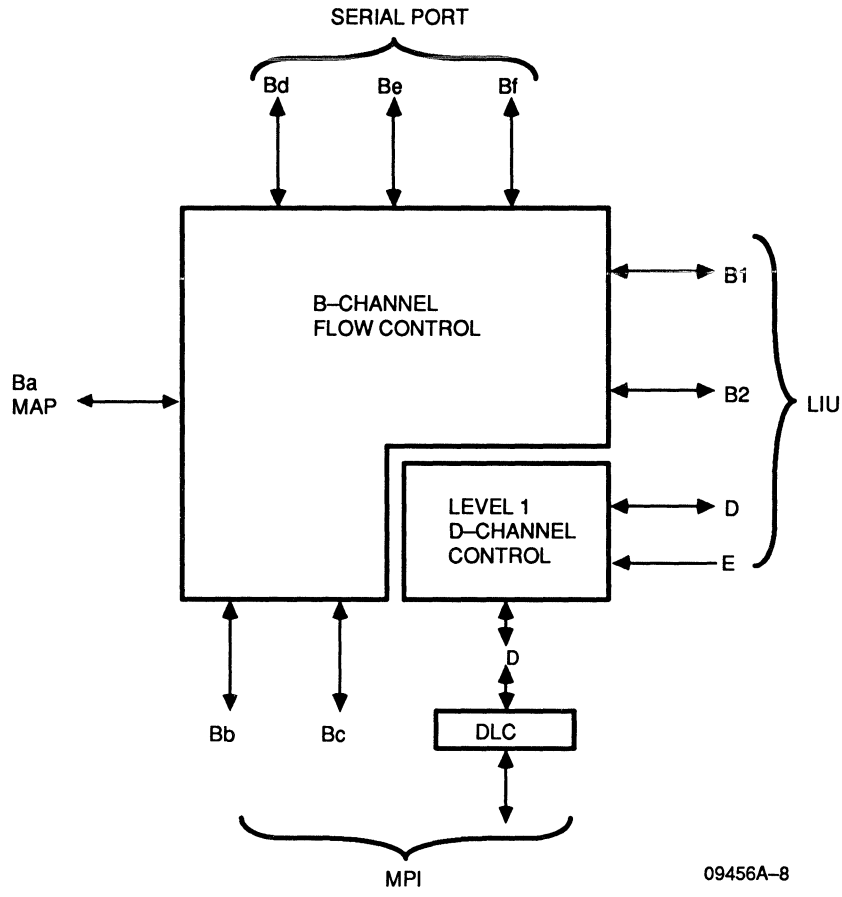


Figure 8. MUX Logical Channels and D-Channel Microprocessor Access

Main Audio Processor (MAP)
(Am79C30A Only)

This block performs the digital-to-analog (DAC) and analog-to-digital (ADC) conversions of the audio signals. The codec and filter functions are implemented using digital signal processing techniques to provide substantial flexibility and programmability. Analog interfaces are provided for a handset earpiece, a handset mouthpiece, a microphone and a loudspeaker. The MAP contains the following programmable, user accessible features:

Multi-Tone Generator

This generator can be used to generate a signal consisting of one or two tones where the frequency and amplitude of the tone is programmable. The tone(s) can be summed into the transmit (and sidetone) path for use as a DTMF tone generator, or single tones can be injected into the receive path. The tones can be used as ringing tones (to the loudspeaker output), dial tones, busy signals, ringback tones or other call progress tones.

Two Attenuation Distortion Correction Filters

There is one attenuation distortion correction filter in the transmit path and one in the receive path. These filters can be programmed to modify the frequency characteristics of the transmit or receive paths and to equalize for the characteristics of the microphones, earpiece speaker, or loudspeaker. They can also be used to add pre and/or post emphasis to make the signals match other characteristics.

Three Programmable Gain Stages

There is one gain adjustment in the transmit path and two in the receive path to provide a wide range of gain control.

Programmable Sidetone Gain

There is a built-in sidetone path which samples the transmit signal, attenuates it by a programmable amount, then sums it into the receive path.

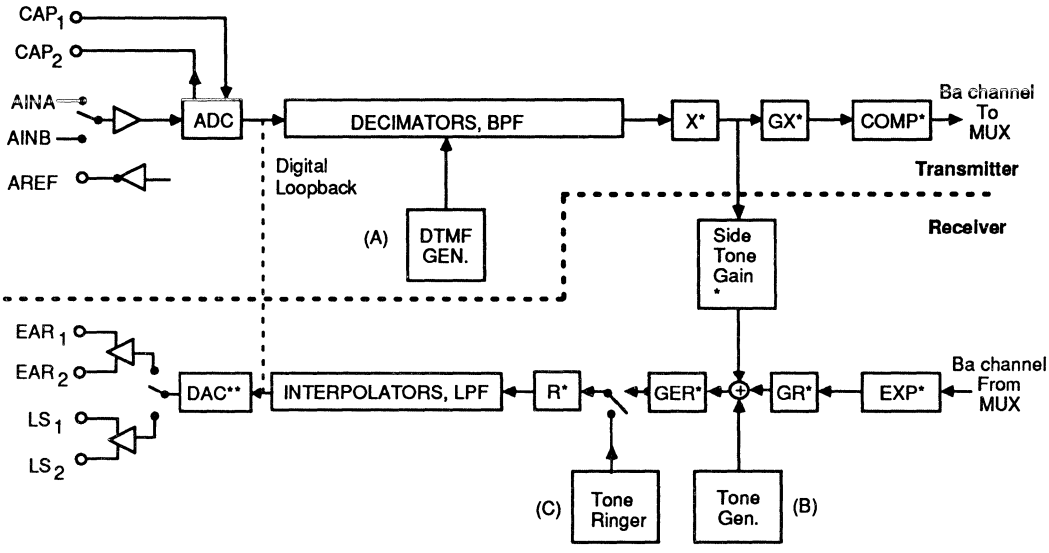
External Interface

Audio Input Port

The audio input port consists of two inputs (AINA and AINB) selectable, one at a time, by setting a bit in the MAP Mode Register 2, (MMR2 bit 0). Signals applied to these inputs must be referenced to AREF.

Earpiece and Loudspeaker Drivers

Each driver consists of push-pull amplifiers with a low impedance output. Either of these audio output ports can be selected, one at a time, via MMR2 bit 1.



	MINIMUM	DEFAULT	MAXIMUM	STEP
GX	0dB **	0dB	12dB	0.5dB
GER	−10dB **	0dB	18dB	0.5dB
GR	−12dB **	0dB	0dB	0.5db
STG	−18dB **	−18dB	0dB	0.5dB

*Programmable
**These registers can also be programmed for inifinite attenuation to break the signal path if desired.

09456–31B

Figure 9. Main Audio Processor Block Diagram

Signal Processing

Transmitter

The transmitter performs a series of operations as described below:

a) An ADC converts the incoming analog signal at a sampling rate of 512 kHz.

b) The Band Pass Filter and a series of decimators reject DC and 50 to 60 Hz line frequencies while reducing the sampling rate to 8 kHz.

c) The X filter is an 8-tap user programmable filter for tuning the microphone. The default is flat with unity gain.

d) The GX filter is a programmable gain filter that allows the user to program a gain of 0 to +12 dB in 0.5 dB steps. The default value is 0 dB.

e) The μ-law or A-law digital compression algorithm converts the linear output of the GX filter to μ- or A-law code. The default algorithm is μ-law code. The MSB (sign bit) is transferred first to (or from) the MUX.

Receiver

The receiver performs a series of operations as described below:

a) An expander converts the input A- or μ-law data to digital linear data. The most significant bit is transferred from the MUX first. The default value is μ-law.

b) The GR filter is a programmable gain filter that allows the user to program a gain of –12 to 0 dB in 0.5 dB steps. The default value of GR is 0 dB.

c) The GER and Sidetone Gain (STG) are programmable constant multipliers which allow the user to program a gain of –10 to +18 dB in 0.5 dB steps (default value 0 dB) and –18 to 0dB in 0.5 dB steps (default value –18 dB) respectively. The GER provides volume control (that is, for the hearing impaired) and should be programmed to 0 dB for normal operation. The sidetone gain path provides feedback from the transmitter.

d) The R filter is provided to correct for speaker attenuation distortion and is a user programmable filter similar to the X filter in the transmitter.

e) A series of interpolators increases the sampling frequency.

f) A DAC converts the digital signal to the analog audio output signal.

Tone Generators

The MAP contains three tone generators which can be enabled via MAP Mode Register 2 bits 2, 3 and 4. Only one of the three tone generator bits in the register can be set at a time. If more than one bit is set, all three bits are considered set to zero and tone generation is disabled. The tone generators are:

Tone Generator	Purpose of Tone Generator
DTMF Generator	Provides tone injection at a sampling rate of 32 kHz into the transmit and sidetone paths (Figure 9 Block A). The DTMF frequencies generated are guaranteed to ±1.2% deviation.
Tone Generation	Provides call progress tones to the receive path where it is added to the incoming speech (Figure 9 Block B).
Tone Ringer	Provides tone alert signals output through the receive path to the loudspeaker or earpiece (Figure 9 Block C).

To program the DTMF tone generators, two frequency values and two amplitude values must be written to the two 8-bit Frequency Tone Generator Registers (FTGR1, FTGR2) and the two 8-bit Amplitude Tone Generator Registers (ATGR1, ATGR2) respectively.

The Tone Generator and the Tone Ringer use the frequency programmed in FTGR1. The Tone Generator uses the amplitude programmed in ATGR1 while the Tone Ringer uses the amplitude programmed in ATGR2.

The FTGR codes to obtain DTMF dialing output frequencies are:

FTGR 2 or 1 → / HEX REG VALUE →		9B	AB	BF	D3
FREQ → / FTGR 1 OR 2 ↓		1209	1336	1477	1633
5A	697	1	2	3	A
63	770	4	5	6	B
6E	852	7	8	9	C
79	941	*	0	#	D

The output frequency of the DTMF tone generator approximately equals (1000 • i) / 128, where i is the value loaded in the FTGR register. This allows the DTMF generator to supply common dual tone call progress signals such as Busy or Dial tones.

FTGR codes to obtain Tone Ringer and Tone Generator output frequencies are:

Tone Ringer and Tone Generator Frequency Coefficients

Frequency (Hz)	Hex Code	Frequency (Hz)	Hex Code
2666	AB	533	23
2000	81	500	21
1600	67	471	1F
1333	56	444	1D
1142	4A	421	1B
1000	41	400	1A
889	39	381	19
800	34	364	18
727	2F	348	17
667	2B	333	16
615	28	320	15
571	25		

The ATGR registers allow the user to program a gain of −18 dB to 0 dB in 2 dB steps. Example ATGR codes to obtain amplitude gains are listed in the following table. 0 dB implies a level of +3 dBm0. The gain values are rounded off to the nearest 1 dB.

Amplitude Gain Coefficients

Gain (dB)	Hex Code	Gain (dB)	Hex Code
−18	37	−8	21
−6	32	−6	20
−14	31	−4	12
−12	27	−2	11
−10	22	0	10

Programmable Gain Coefficients

The GER, GR, GX and Sidetone gain coefficients are each 16 bits in length. Two consecutive register locations correspond to one gain coefficient. The LSB is transferred first to (or from) the microprocessor. Care should be taken to minimize the number of times GX, GR, X, R, STG, and GER are accessed during a phone call, to reduce interference with MAP calculations. Example coefficients for the GER filter are listed in the following table. The gain values are rounded off to the nearest 0.1dB.

GER Gain Coefficients

Gain (dB)	Hex Code MSB	Hex Code LSB	Gain (dB)	Hex Code MSB	Hex Code LSB
–10.0	AA	AA	4.0	31	DD
–9.5	9B	BB	4.5	44	1F
–9.0	79	AC	5.0	43	1F
–8.5	09	9A	5.5	33	1F
–8.0	41	99	6.0	40	DD
–7.5	31	99	6.5	11	DD
–7.0	9C	DE	7.0	44	0F
–6.5	9D	EF	7.5	41	1F
–6.0	74	9C	8.0	31	1F
–5.5	54	9D	8.5	55	20
–5.0	6A	AE	9.0	10	DD
–4.5	AB	CD	9.5	42	11
–4.0	AB	DF	10.0	41	0F
–3.5	74	29	10.5	11	1F
–3.0	64	AB	11.0	60	0B
–2.5	6A	FF	11.5	00	DD
–2.0	2A	BD	12.0	42	10
–1.5	BE	EF	12.5	40	0F
–1.0	5C	CE	13.0	11	0F
– .5	75	CD	13.4	22	10
0.0	00	99	14.0	72	00
.5	55	4C	14.5	42	00
1.0	43	DD	15.0	21	10
1.5	33	DD	15.5	10	0F
2.0	52	EF	15.9	22	00
2.5	77	1B	16.6	11	10
3.0	55	42	16.9	00	0B
3.5	41	DD	17.5	21	00
			18.0	00	0F

The coefficient 0008 provides an attenuation of infinity when GER gain is enabled.

Example coefficients for the GR, GX, and STG filter are listed in the following table. The gain values are rounded off to the nearest 0.1 dB

GX Gain Coefficients

Gain (dB)	Hex Code MSB	LSB
0.0	08	08
0.5	4C	B2
1.0	3D	AC
1.5	2A	E5
2.0	25	33
2.5	22	22
3.0	21	22
3.5	1F	D3
4.0	12	A2
4.5	12	1B
5.0	11	3B
5.5	0B	C3
6.0	10	F2
6.5	03	BA
7.0	02	CA
7.5	02	1D
8.0	01	5A
8.5	01	22
9.0	01	12
9.5	00	EC
10.0	00	32
10.5	00	21
11.0	00	13
11.5	00	11
12.0	00	0E

GR Gain Coefficients

Gain (dB)	Hex Code MSB	LSB
–12.0	91	F9
–11.5	91	C5
–11.0	91	B6
–10.5	92	12
–10.0	91	A4
–9.5	92	22
–9.0	92	32
–8.5	92	FB
–8.0	92	AA
–7.5	93	27
–7.0	93	B3
–6.5	94	B3
–6.0	9F	91
–5.5	9C	EA
–5.0	9B	F9
–4.5	9A	AC
–4.0	9A	4A
–3.5	A2	22
–3.0	A2	A2
–2.5	A6	8D
–2.0	AA	A3
–1.5	B2	42
–1.0	BB	52
–0.5	CB	B2
0.0	08	08

STG Gain Coefficients

Gain (dB)	Hex Code MSB	LSB
–18.0	8B	7C
–17.5	8B	44
–17.0	8B	35
–16.5	8B	2A
–16.0	8B	24
–15.5	8B	22
–15.0	91	23
–14.5	91	2E
–14.0	91	2A
–13.5	91	32
–13.0	91	3B
–12.5	91	4B
–12.0	91	F9
–11.5	91	C5
–11.0	91	B6
–10.5	92	12
–10.0	91	A4
–9.5	92	22
–9.0	92	32
–8.5	92	FB
–8.0	92	AA
–7.5	93	27
–7.0	93	B3
–6.5	94	B3
–6.0	9F	91
–5.5	9C	EA
–5.0	9B	F9
–4.5	9A	AC
–4.0	9A	4A
–3.5	A2	22
–3.0	A2	A2
–2.5	A6	8D
–2.0	AA	A3
–1.5	B2	42
–1.0	BB	52
–0.5	CB	B2
0.0	08	08

The coefficient 9008 provides an attenuation of infinity when GR, GX, and/or STG are enabled.

Programmable Filter Coefficients and Equations

The frequency domain transfer function equation for the X and R filters is :

$$h_f = h_0 + h_1z^{-1} + h_2z^{-2} + h_3z^{-3}\ h_4z^{-4} + h_5z^{-5} + h_6z^{-6} + h_7z^{-7}$$

where:

$$z = \cos(wT) + i \bullet \sin(wT)$$

$$i = (-1)^{1/2}$$

w = frequency of input signal in Hz • 2pi
T = sample period in seconds (0.125 ms)
hj (j = 0,1,...7) = user defined coefficients

Each hj coefficient is defined by the following equation:

$$hj = A3\ (1 + A2\ (1 + A1\ (1 + A0\ )\ )\ )$$

where each hj Coefficient Register pair has the following format:

Byte	7	6 5 4	3	2 1 0
LSB	S1	M1	S0	M0
MSB	S3	M3	S2	M2

and $Ai = -1^{Si}2^{-Mi}$, (i=0,1,2,3). The X and R filter coefficients are programmed using a 16 byte transfer with the following format:

Byte	Value
0	h0 LSB
1	h0 MSB
2	h1 LSB
3	h1 MSB
4	h2 LSB
5	h2 MSB
6	h3 LSB
7	h3 MSB
8	h4 LSB
9	h4 MSB
10	h5 LSB
11	h5 MSB
12	h6 LSB
13	h6 MSB
14	h7 LSB
15	h7 MSB

AmMAP, a software package which calculates X and R filter coefficients, is available from Advanced Micro Devices.

Overflow/Underflow Precautions when Using Programmable Gains

Care must be taken such that at any point in the signal processing path, the combination of gains and filters and/or tones does not result in a signal that is larger than full scale. Full scale is defined as the digital representation of the maximum analog signal which is allowed into the transmitter or out of the receiver with all filters and gain stages at their default (0 dB) settings (for example, in A-Law, the transmitter full scale is ±1.25 Vp and the receiver full scale is ±2.5 Vp). Likewise, it is desirable that the peak signal be kept as close to full scale as possible at any point in the signal processing path in order to minimize digital truncation effects in the A/D, D/A and MAP DSP.

Consider the following example: STG is programmed for infinite attenuation, GR is programmed to –6 dB while GER is programmed to +12 dB and the R filter is programmed such that it exhibits a net gain of –6 dB. Assume that the analog full scale out of the receiver is ±2.5 Vp and that a full scale PCM code is possible from the MUX. After GR, the equivalent analog signal would be 2.5/2 = ±1.25 Vp. However, after GER the signal would be 1.25 • 4 or ±5 Vp. Even though the R filter would have a net gain of –6 dB, the signal would have clipped after GER and the signal would be distorted for PCM codes between full scale and 6 dB below full scale because of the intermediate result at the output of GER.

Care should also be taken when programming the tone ringers/generators. For example if one of the DTMF tones is programmed to 0 dB, a tone is generated which is equivalent to a ± full scale signal in the transmit path. This means that there is no headroom left for the other DTMF tone. Therefore, the DTMF generator should never be programmed in such a way as to exceed full scale if signal quality is to be maintained. In the receive path, similar caution should be exercised in order to prevent the combination of Tone Generator, Side Tone, GR, and GER from clipping the signal.

Test Facilities

Two test/maintenance facilities, MAP Analog Loopback and MAP Digital Loopback, are provided to diagnose the MAP functions.

MAP Analog Loopback

This test enables the monitoring of the analog output (EAR1/EAR2 or LS1/LS2) and the analog input (AINA or AINB). Signals sent in on AINA or AINB are sent back on EAR1/EAR2 or LS1/LS2 via the MUX. The MUX must be set up for Ba to Ba loopback by writing 33 Hex to MCR1, 2 or 3. No other Ba connections should be programmed.

MAP Digital Loopback

The digital portion of the MAP can be tested from the 'S' interface or the microprocessor by setting bit 7 of MAP Mode Register 1 to a logical '1'. With this option enabled the data is taken out of the interpolator and fed as the input to the decimator, in place of the output of the ADC. It should be noted that in digital loopback, the D to D gain is approximately 2.5 dB.

MAP Registers

The MAP contains the following byte wide programmable registers:

MAP Registers	No.	Mnemonic
X Filter Coefficient Registers	16	X
R Filter Coefficient Registers	16	R
GX Gain Coefficient Registers	2	GX
GR Gain Coefficient Registers	2	GR
GER Gain Coefficient Registers	2	GER
Sidetone Gain Coefficient Registers	2	STGR
Frequency Tone Generator Registers	2	FTGR
Amplitude Tone Generator Registers	2	ATGR
MAP Mode Registers	2	MMR

Note: It is necessary to complete any transfers to the multibyte MAP registers; for instance, a total of 16 bytes must be transferred to update the X-filter.

Following power-up, all MAP registers, except the two FTGR registers and the two MAP mode registers, contain unknown values. Only the FTGR registers and the MMR registers reset to "0" HEX. All other registers in the MAP (X, R, GX, GR, GER, ATGR, and STG) are not affected by reset and must be programmed via the microprocessor before being enabled. When the registers are disabled, or after reset, the MAP will have the following responses:

Filter	Default Response
X Filter	Disabled (0 dB, Flat)
R Filter	Disabled (0 dB, Flat)
GX Filter	Disabled (0 dB, Gain)
GR Filter	Disabled (0 dB, Gain)
GER Filter	Disabled (0 dB, Gain)
Sidetone gain	Disabled (–18 dB, Gain)

The bits in the MAP Mode Register define the enable/disable options for the various MAP configurations as follows:

MAP Mode Register 1 (MMR1), Read/Write

Bit	Logical '1'	Logical '0' (default mode)
0	A-Law	μ-Law
1	GX coefficient loaded from register	GX bypassed; gain = 0 dB
2	GR coefficient loaded from register	GR bypassed; gain = 0 dB
3	GER coefficient loaded from register	GER bypassed; gain = 0 dB
4	X coefficient loaded from register	X bypassed; response = flat
5	R coefficient loaded from register	R bypassed; response = flat
6	Sidetone gain coefficient loaded from register	STG gain = –18 dB*
7	Digital loopback at MAP enabled	Digital loopback at MAP disabled

*To remove the sidetone path completely, it is necessary to enable the STG function by setting MMR1 bit 6 to '1', and program the STGR coefficient to 9008 (Hex)

MAP Mode Register 2 (MMR2), Read/Write

Bit	Logical '1'	Logical '0' (default mode)
0	AINB selected	AINA selected
1	LS1/LS2 selected	EAR1/EAR2 selected
2	DTMF enabled	DTMF disabled
3	Tone generator enabled	Tone generator disabled
4	Tone ringer enabled	Tone ringer disabled
5	High pass filter disabled	High pass filter enabled
6	ADC auto-zero function disabled	ADC auto-zero function enabled
7	Reserved, must be logical '0'	Reserved, must be logical '0'

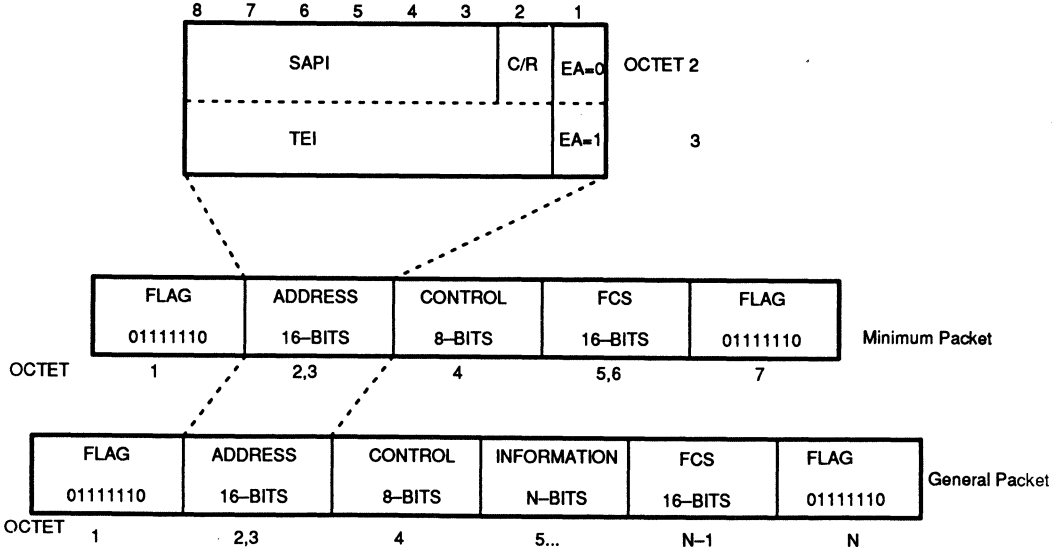
Note: For most applications, MMR2 bits 5 and 6 should always be written to logical '0'. This enables the 50-60 Hz rejection filter and the internal offset cancellation circuits to operate normally. They can both be disabled when system or test conditions require that DC or low frequency signals be transmitted.

Data Link Controller (DLC)

A 16 kbps D-channel is time multiplexed within the frame structure of the 'S' interface. The data carried by the D-channel is encoded using the Link Access Protocol D-channel (LAPD) format shown in Figure 10. The D-channel can be used to carry either end-to-end signaling or slow speed packet data. Further information concerning the LAPD protocol can be found in CCITT recommendations. The LIU controls the multiplexing and demultiplexing of the D-channel data between the 'S' interface and the DLC.

The DLC performs processing of level 1 and partial level 2 LAPD protocol including flag detection and generation, zero deletion and insertion, Frame Check Sequence (FCS) processing for error detection, and some addressing capability. High level protocol processing is done by the external microprocessor. The microprocessor may process the address field in the LAPD frame depending on the programmed state of the DLC. The status of the DLC is held in the status registers and relevant interrupts are generated under user program control. The DLC also contains a 16-bit pseudo-random number generator (RNG) which is used in the D-channel address allocation procedure as defined in the CCITT recommendations, as well as two 8-byte data FIFO's.

D-channel Processing



Notes: EA = Address Field Extension Bit
C/R = Command/Response Field Bit
SAPI = Service Access Point Identifier
TEI = Terminal Endpoint Identifier
FCS = Frame Check Sequence

09456–34B

Figure 10. Level Two Frame Structure Formats

Random Number Generator (RNG)

The RNG is accessible by the microprocessor and operates in the following manner:

On the low to high transition of the reset signal the RNG is cleared, then started. The RNG stops when the LSB or MSB of the 16-bit counter is read by the microprocessor or when the MSB is loaded by the microprocessor. Writing to the MSB of the counter loads this byte but does not start the RNG. The RNG starts when the LSB of the counter is loaded by the microprocessor.

Frame Abort

The DLC aborts an incoming D-channel frame upon the reception of seven contiguous logical '1's. When this occurs, an end of receive packet interrupt is issued to the processor. DER bit 0 is set to a logical '1' when the last byte of the aborted packet is read from the D-channel Receive Buffer. The 'Receive Abort' interrupt can be masked by setting DMR2 bit 0 to a logical '0'. With the exception of the 'Packet reception in progress' bit, no other bits associated with packet reception are updated after a receive packet abort. The receive frame can be aborted at any time by setting INIT bit 6 to logical '1'. Similarly, the transmit frame can be aborted by setting INIT bit 7 to a logical '1'. When the transmit frame is aborted, seven consecutive ones are transmitted on the 'S' interface followed by a logical '0' and DSR1 bit 7 is set to a logical '1'. Seven consecutive '1's followed by a '0' will continue to be transmitted as long as INIT bit 7 is set to a '1'. DSR1 bit 7 will be set after each sequence of seven consecutive 1's followed by a '0'.

Level Two Frame Structure

The D-channel level two frame structure conforms to one of the formats shown in Figure 10. All frames start and end with the flag sequence consisting of one '0' followed by six '1's followed by one '0'. A packet consists of a level two frame minus the flag bytes. The LSB is transmitted first for all bytes except the FCS.

The flag preceding a packet is defined as the opening flag. Therefore, the byte following an opening flag, by definition, cannot be an abort or another flag. A closing flag is defined as a flag that terminates a packet. This flag can be followed by another flag(s), interframe fill consisting of all ones or flags, or the address field of the next packet. In the latter case, the closing flag of one packet is the opening flag of the next packet. The DLC receiver is able to recognize interframe fill consisting of logical ones or flags. The DLC transmitter follows the closing flag with interframe fill consisting of all ones if mark idle is selected by setting DMR4 bit 4 to a logical '0' (thereby satisfying the D-channel access protocol defined in the CCITT I-series recommendations) or continuous flags if flag idle is selected by setting DMR4 bit 4 to a logical '1'.

Upon detection of collision (mismatch of a D and E bit), a complete frame must be retransmitted. Note that for transfer across the 'S' interface, the frame structure of the 'S' interface itself is impressed upon the frame structure (LAPD) of the D-channel.

Zero Insertion/Deletion

When transmitting, the DLC examines the frame content between the opening and closing flags. To ensure that a flag sequence is not repeated within the flag boundaries of the frame, a logical '0' bit is automatically inserted after each sequence of five contiguous logical ones. When receiving, the DLC examines the frame content between the opening and closing flags and automatically discards the first logical zero which directly follows five contiguous logical ones.

D-Channel Address Recognition

The address field, shown in Figure 10, allows for three types of addresses: 1) a one byte address signified by the LSB of the first address byte being set to a logical '1'; 2) a two byte address signified by the LSB of the first address byte being set to a logical '0' and the LSB of the second address byte being set to a logical '1'; 3) a more than two byte address signified by the LSB of both the first and second address bytes being set to a logical '0'. In the case of the LAPD operating environments, the address is a two byte address where the first byte is analogous to the Service Access Point Identifier (SAPI) and the second byte is analogous to the Terminal Endpoint Identifier (TEI) as defined by the CCITT recommendations.

The DLC is able to recognize D-channel addresses of all of the three types outlined above. Note that only the first two bytes of a more than two byte address can be checked by the DLC. There are four First Received Byte Address Registers (FRARs) which hold the values used to match against the first byte of the incoming address. Similarly, there are four Second Received Byte Address Registers (SRARs) which hold the values used to match against the second byte of the incoming address.

FRAR4 defaults to FE hex; SRAR4 defaults to FF hex. This default is analogous to the broadcast address defined by the CCITT recommendations. The type of address recognition which is enabled is determined as follows:

DMR4 bit 7	DMR4 bit 5	DMR1 bits 7	6	5	4		Type of address recognition
0	1	X	X	X	1	FRAR1	First received byte only address
		X	X	1	X	FRAR2	
		X	1	X	X	FRAR3	
		1	X	X	X	FRAR4	
1	1	X	X	X	1	SRAR1	Second received byte only address
		X	X	1	X	SRAR2	
		X	1	X	X	SRAR3	
		1	X	X	X	SRAR4	
X	0	X	X	X	1	FRAR1 : SRAR1	Two byte address
		X	X	1	X	FRAR2 : SRAR2	
		X	1	X	X	FRAR3 : SRAR3	
		1	X	X	X	FRAR4 : SRAR4	
X	X	0	0	0	0		Address recognition disabled

If DMR4 bit 6 is set to a logical '0', bit 1 of the FRARs is ignored when matching the first incoming address byte. If DMR4 bit 6 is set to a logical '1', all bits of the FRARs are used when matching the first incoming address byte. FRAR bit 1 is analogous to the C/R bit defined by the CCITT recommendations. The address recognition mechanism for the four FRAR/SRAR addresses can be individually enabled/disabled via DMR1 bits 4-7.

First Received Byte Only Address Recognition

If DMR4 bit 5 is set to a logical '1' and DMR4 bit 7 is set to a logical '0', only the first byte of the incoming address is compared to the values stored in the enabled FRARs. An interrupt is generated if there is an address match and the 'Valid Address' interrupt is enabled. If the address matches, the packet will be received.

Second Received Byte Only Address Recognition

If DMR4 bits 5 and 7 are set to a logical '1', the DLC compares only the value in the second byte of the incoming address to values stored in the enabled SRARs. An interrupt is generated if there is an address match and the 'Valid Address' interrupt is enabled. If the address matches, the packet will be received.

Two Byte Address Recognition

If DMR4 bit 5 is set to a logical '0', the first byte of the incoming address is compared to the values stored in the enabled FRARs and the second byte of the incoming address is compared to the value stored in the corresponding SRAR. An interrupt is generated if a match is found for both incoming address bytes with a FRAR/SRAR pair and the 'Valid Address' interrupt is enabled. If the address matches, the packet will be received.

Disabling Address Recognition

If DMR1 bits 4, 5, 6, and 7 are all set to logical '0', all address recognition is disabled and all addresses are recognized and received. In this case, the Am79C30A/32A receives the first two bytes following the opening flag (the incoming address), and then issues an 'End of Address' interrupt if the 'End of Address' interrupt is enabled.

DLC Operation

Receiving D-Channel Packets

The receiver controls the flow of D-channel data to the D-channel Receive Buffer and the termination of a receive packet. Up to two packets can be contained in the D-channel Receive Buffer.

After receiving an opening flag (a bit sequence of 01111110) and one byte of data which is not an abort or flag on the D-channel, the DLC sets the 'Packet reception in progress' status bit (bit 2) in D-channel Status Register 1 (DSR1). The DLC then receives the first two bytes (the two address bytes). If address recognition is enabled, the Am79C30A/32A issues a valid address interrupt if a match between the programmed values and the received address is detected. If no match is detected and address recognition is enabled, the DLC ignores the packet. If address recognition is disabled, the Am79C30A/32A receives the first two bytes, issues an 'End of Address' interrupt, and receives the packet. Both a 'Valid Address' and an 'End of Address' interrupt set Interrupt Register bit 2 to a logical '1' and bit 0 of the D-channel Status Register 1 (DSR1) to a logical '1'. The 'Valid Address/End of Address' interrupt can be disabled via DMR3 bit 0. There is an internal three byte delay which holds the first of the D-channel address bytes until the interrupt has been issued. Note that the incoming address bytes cannot be read however, until the 'D-channel receive byte available' or 'D-channel receive threshold' interrupt is set.

After the address is received, the DLC continues to receive D-channel bytes into the D-channel Receive Buffer FIFO. The DLC issues an interrupt when data is available in the D-channel Receive Buffer. This interrupt can be disabled by setting DMR3 bit 3 to a logical '0'. The DLC also issues an interrupt when the receive

threshold set in DMR4 is reached. This interrupt can be disabled by programming a logical '0' into DMR1 bit 1. By polling, the microprocessor can then read the D-channel bytes. The three byte delay incurred during address recognition is maintained. Therefore, the DLC receives the Frame Check Sequence (FCS) before issuing an interrupt to signal the last byte of the packet has been received and appropriate status bits have been updated. If DMR3 bit 7 is set, the two FCS bytes at the end of the packet are transferred into the D-channel Receive Buffer along with the data.

The DLC issues an interrupt when the last byte of the packet is read from the DCRB. This interrupt can be disabled by setting DMR3 bit 2 to a logical '0'.

After the FCS is received, the DLC receiver detects the closing flag (a bit sequence of 01111110) and then terminates the packet by issuing an 'end of receive packet' interrupt (bit 1 of DSR1) and returns to looking for opening flags. The DLC also terminates the packet when an abort or an overflow or overrun error condition is detected. The 'end of receive packet' interrupt can be disabled by setting DMR1 bit 3 to a logical '0'.

The D-channel Receive Byte Count Register (DRCR) is a 16-bit wide 2-word deep FIFO which is used to record the number of bytes in the incoming D-channel packets. Each count is terminated by an end of packet condition. Thus, the DRCR informs the microprocessor of the number of bytes, including the address bytes, which have been received. The counter is updated when the last byte of a packet is placed in the D-channel Receive Buffer. When the FCS bytes are included in the data transferred to the D-channel Receive Buffer, the FCS bytes are included in the byte count; if the FCS bytes are not included in the transfer, they are not included in the byte count. The opening flag and closing flag are not included in the byte count.

The D-channel Error and Address Status Registers are also double buffered. Reading the last byte of a packet causes the DER byte to propagate to the output of the FIFO and updates the D-channel Status and Interrupt Registers accordingly. Reading the MSB of the DRCR causes the next count and associated ASR byte to propagate to the output of the FIFOs and updates the D-channel Status and Interrupt Registers accordingly. For this reason it is important to read ASR, DER and DSR1 prior to reading the DRCR.

When a receive error occurs, an end of packet interrupt is generated and the packet is terminated. When the last byte of the associated packet is read from the D-channel Receive Buffer the appropriate DER bits are set and an error interrupt is generated. All error interrupts can be individually masked by setting the corresponding bits in DMR2 to a logical '0'.

There is one 16-bit D-channel Receive Byte Limit Register (DRLR). The received byte count is compared to the DRLR. When the byte count of the currently received D-channel packet exceeds the limit value, a receiver overflow is detected, the packet is terminated and an end of packet interrupt is issued. D-channel Error Register (DER) bit 4 is set to a logical '1' and an overflow interrupt issued when the last byte of the associated packet is read from the D-channel Receive Buffer. The overflow error interrupt can be masked by setting DMR2 bit 4 to a logical '0'.

The minimum packet length is five bytes for a two byte address packet (not including flags). If the packet length is less than the above, an interrupt is issued and DER bit 5 is set to a logical '1' when the last byte of the associated packet to read from the D-channel Receive Buffer. The error interrupt can be masked by setting DMR2 bit 5 to a logical '0'.

If packet reception is in progress and the D-channel Receive Buffer is full, the microprocessor has a maximum of 425 µsec to respond to the D-channel receive data available interrupt. If the microprocessor fails to do so, then an overrun error occurs when the data byte is overwritten. When this happens, the packet is terminated. DER bit 6 is set to a logical '1' when the last byte of the associated packet is read from the D-channel Receive Buffer. The overrun error interrupt can be masked by setting DMR2 bit 6 to logical '0'.

Error indication is given if two packets have been received and not serviced by the user and a third packet is received via DSR2 bit 2. When this error occurs, the third packet is terminated (that is, not received).

Error indication is given for a receiver abort (the reception of seven contiguous 1s) by DER bit 0.

If the number of bits received between two flags is not an integral multiple of 8 (that is, if the received packet does not contain an integral number of bytes), DER bit 1 is set and an interrupt is generated when the last byte of the associated packet is read from the D-channel Receive Buffer.

The incoming bit stream (including FCS) is run through the FCS generation and compare block. Upon receipt of the closing flag, the result is checked and must be (MSB first) 0001110100001111. Any other pattern indicates an FCS error and DER bit 3 is set to a logical '1' when the last byte of the associated packet is read from the D-channel Receive Buffer.

The DLC receiver does not assume the packet to be byte-aligned. The architecture supports shared flags between packets, interframe fill consisting of logical ones (mark idle), and interframe fill consisting of flags (flag idle). Mark idle is defined as at least 15 or more contiguous ones. Flag idle is defined as more than two consecutive flag characters, not including a closing flag. DSR2 bit 5 is set to a logical '1' while mark idle is being detected. DSR2 bit 6 is set to a logical '1' while flag idle is being detected. The receiver D-channel packet can be aborted at any time during reception by setting INIT bit 6.

Transmitting D-Channel Packets

The DLC Transmitter is activated as soon as the MSB (the second byte) of the 16 bit D-channel transmit byte count register (DTCR) has been loaded by the microprocessor.

Next the LIU starts counting the number of consecutive '1's on the E-channel until the number of '1's defined by the LIU priority mechanism is detected. After the sequence of '1's the DLC transmitter will begin packet transmission.

Address bytes for a transmit packet can be handled in two ways: they can be loaded into the transmit buffer or loaded into the transmit address register (TAR).

There is one 16-bit Transit Address Register (TAR) which can be loaded by the microprocessor. The bytes loaded into the TAR are transmitted LSB first followed by MSB. For LAPD operation the LSB contains the SAPI and the MSB contains TEI. This 16-bit address (loaded LSB first) is transmitted within the address field of the D-channel packet if enabled by setting DMR1 bit 2 to a logical '1'. If the TAR is enabled, the DTCR should be loaded with the number of bytes to be transmitted excluding the address, flags and FCS. If the TAR is disabled, the DTCR should be loaded with the number of bytes to be transmitted excluding the flags and FCS and the microprocessor must load the address to be transmitted as the first two bytes of the D-channel packet data.

The DLC issues an interrupt when a position is available in the D-channel Transmit Buffer. This interrupt can be disabled by setting DMR3 bit 5 to a logical '0'. The DLC also issues an interrupt to the microprocessor to request D-channel data bytes when the D-channel Transmit Buffer empties to the threshold specified in the D-channel FIFO Mode Register. This interrupt can be disabled by setting DMR1 bit 0 to a logical '0'.

If the D-channel Transmit Buffer is empty, the microprocessor has up to 375 µs to respond to the D-channel transmit buffer interrupt. If the microprocessor fails to load the data bytes in this time frame, an underrun interrupt is generated in DER bit 7 and packet transmission is terminated with a transmitted abort. The underrun interrupt can be masked by setting DMR2 bit 7 respectively to a logical '0'. Transmission is also terminated when a collision is detected or LIU loss of synchronization occurs.

The D-channel Transmit Byte Count Register is decremented each time a byte of data is transferred from the D-channel Transmit Buffer to the DLC. The count represents the number of bytes left to be transferred excluding the FCS and flags. If the transmit abort bit (INIT bit 7) is set, the transmit byte count is frozen and indicates the number of bytes left to transfer and not the number of bytes transmitted. The last byte of the packet is determined by the D-channel Transmit Byte Count decrementing to zero. When this occurs, DSR2 bit 3 is set to a logical '1'.

After the last byte of the packet is transmitted, the DLC adds the FCS and closing flag. Then the DLC issues an interrupt (bit 6 of DSR1) to signify the end of the packet transmission. This interrupt can be masked by setting DMR3 bit 1 to a logical '0' and is reset either by reading DSR1 or when the D-channel Transmit Byte Count Register is loaded for the next packet.

Once the D-channel Transmit Byte Count has decremented to zero, a second packet may be loaded into the D-channel Transmit FIFO. If the MSB of the D-channel Transmit Byte Count Register is loaded prior to the end of transmit packet interrupt, the second packet is transmitted back-to-back with the previous packet. The end of transmit packet interrupt is not set between the two packets. If the MSB of the D-channel Transmit Byte Count Register is loaded after the end of packet interrupt, the second packet is transmitted once the LIU priority mechanism has been re-satisfied.

Collision Detection

The Network Terminator echoes the transmitted D-channel data back to the DLC in the E-channel bits of the 'S' interface frame. If there is a difference between the data transmitted and the data echoed back, a collision has occurred. The DLC alerts the microprocessor to this event by asserting the interrupt line ($\overline{INT}$) and setting DER bit 2. If a collision occurs during the transmission of an abort sequence, the interrupt is still issued. The collision detect interrupt can be masked by setting DMR2 bit 2 to a logical '0'.

D-Channel Receive and Transmit Errors

Non-Integral Number of Bytes

A non-integral number of bytes occurs when the number of D-channel bits received between opening and closing flags is not divisible by 8. If a received packet consists of a non-integral number of bytes, the DLC sets bit 1 in the D-channel Error Register (DER) to a logical '1' when the last byte of the associated packet is read from the D-channel Receive Buffer.

Frame Check Sequence Error

If a received packet including its 16-bit Frame Check Sequence is not received perfectly, the DLC sets DER bit 3 to a logical '1' when the last byte of the associated packet is read from the Receive Buffer.

Receive Packet Abort

If seven contiguous 1's are received while receiving a packet, the packet will be terminated and DER bit 0 will be set to a logical '1' when the last byte of the associated packet is read from the D-channel receive buffer.

Overflow

Overflow occurs when the total number of D-channel bytes within a packet (including, only when enabled, the Frame Check Sequence bytes) exceeds the limit contained in the D-channel Receive Byte Limit Register. (See Receiving D-channel Packets section.) When this occurs, the DLC terminates the packet, and sets DER bit 4 to a logical '1' when the last byte of the associated packet is read from the D-channel Receive Buffer.

Underflow

If a received D-channel (including FCS) packet is less than five bytes for a two byte address packet, an underflow error condition occurs and the DLC sets DER bit 5 to a logical '1' when the last byte of the associated packet is read from the D-channel Receive Buffer.

Overrun

A D-channel overrun error occurs when the receiver buffer is full and another byte is received. This can happen if the D-channel receive buffer fills and is not read within 425 µs. When this error occurs, the DLC sets DER bit 6 to a logical '1' and terminates the packet.

Underrun

A D-channel underrun error occurs when an empty D-channel buffer is transmitted. This can happen if the D-channel transmit buffer is not loaded within 375 µs of the 'D-channel transmit buffer empty' interrupt being asserted (IR bit 0). When this error occurs, the DLC sets DER bit 7 to a logical '1' and terminates the packet.

Receive Packet Lost

Receive packet lost occurs when two outstanding packets have been received and not serviced (that is, the microprocessor has not read the DRCB register) and a third packet is received. When this error occurs, DSR2 bit 2 is set to a logical '1' and the incoming packet is terminated (not received).

DLC Registers

The DLC contains the following registers:

Registers	No.	Mnemonic
First Received Byte Address Registers	4	FRAR
Second Received Byte Address Registers	4	SRAR
Transmit Address Register (16 bit)	1	TAR
D-channel Receive Byte Limit Register (16 bit)	1	DRLR
D-channel Receive Byte Count Register (16 bit) (2 word FIFO)	1	DRCR
D-channel Transmit Byte Count Register (16 bit)	1	DTCR
Random Number Generator Registers	2	RNGR
D-channel Mode Registers	4	DMR
Address Status Register (2 byte FIFO)	1	ASR

There are three other accessible registers associated with the DLC. They are microprocessor read only registers. These registers are:

D-channel Status Registers	(DSR1 and DSR2)
D-channel Error Register	(DER) (2 byte FIFO)

D-Channel Mode Register 1 (DMR1), Read/Write

DMR1 controls the enable/disable options for the DLC. It is under sole control of the microprocessor and does not generate any interrupts. DMR1 is defined below:

Bit	Logical '1'	Logical '0'
0	Enable 'D-channel transmit threshold' interrupt (see IR bit 0)	Disable interrupt (default value)
1	Enable 'D-channel receive threshold' interrupt (see IR bit1)	Disable interrupt default value)
2	Enable Transmit Address Register	Disable Transmit Address Register (default value)
3	Enable 'end of receive packet' interrupt (see DSR1 bit 1)	Disable interrupt (default value)
4	Enable FRAR1/SRAR1	Disable FRAR1/SRAR1 (default value)
5	Enable FRAR2/SRAR2	Disable FRAR2/SRAR2 (default value)
6	Enable FRAR3/SRAR3	Disable FRAR3/SRAR3 (default value)
7	Enable FRAR4/SRAR4 (default value)	Disable FRAR4/SRAR4

D-Channel Mode Register 2 (DMR2), Read/Write

DMR2 is used to enable/disable the interrupts generated in the DER (See DER definition below). DMR2 is controlled by the microprocessor and does not generate interrupts. DMR2 is defined below:

Bit	Logical '1'	Logical '0' (default value)
0	Enable 'receive abort' interrupt (see DER bit 0)	Disable interrupt
1	Enable 'non-integral number of bytes received' interrupt (see DER bit 1)	Disable interrupt
2	Enable 'collision abort detected' interrupt (see DER bit 2)	Disable interrupt
3	Enable 'FCS error' interrupt (see DER bit 3)	Disable interrupt
4	Enable 'overflow error' interrupt (see DER bit 4)	Disable interrupt
5	Enable 'underflow error' interrupt (see DER bit 5)	Disable interrupt
6	Enable 'overrun error' interrupt (see DER bit 6)	Disable interrupt
7	Enable 'underrun error' interrupt (see DER bit 7)	Disable interrupt

D-Channel Mode Register 3 (DMR3), Read/Write

Bit	Logical '1'	Logical '0'
0	Enable 'Valid Address/End of Address' interrupt (default value) (see DSR1 bit 0)	Disable interrupt
1	Enable 'End of valid transmit packet' interrupt (default value) (see DSR1 bit 6)	Disable interrupt
2	Enable 'Last byte of received packet' interrupt (see DSR2 bit 0)	Disable interrupt (default value)
3	Enable 'Receive byte available' interrupt (see DSR2 bit 1)	Disable interrupt (default value)
4	Enable 'Last byte transmitted' interrupt (see DSR2 bit 3)	Disable interrupt (default value)
5	Enable 'Transmit buffer available' interrupt (see DSR2 bit 4)	Disable interrupt (default value)
6	Enable 'Received packet lost' interrupt (see DSR2 bit 2)	Disable interrupt (default value)
7	Enable FCS transfer to FIFO	Disable FCS transfer to FIFO (default value)

D-Channel Mode Register 4 (DMR4), Read/Write

Bit 7 6 5 4 3 2 1 0	Control	Function
X X X X X X 0 0	Receiver Threshold	1 Byte (default value)
X X X X X X 0 1		2 bytes
X X X X X X 1 0		4 bytes
X X X X X X 1 1		8 bytes
X X X X 0 0 X X	Transmitter Threshold	1 Empty Byte
X X X X 0 1 X X		2 Empty Bytes
X X X X 1 0 X X		4 Empty Bytes
X X X X 1 1 X X		8 Empty Bytes (default value)
X X X 0 X X X X	Interframe Fill	Mark Idle (default value)
X X X 1 X X X X		Flag Idle
X X 0 X X X X X	Address Recognition	Two byte (default value)
0 X 1 X X X X X		First Received Byte only
1 X 1 X X X X X		Second Received Byte only
X 0 X X X X X X	C/R Bit Compare	Disable FRAR bit 1 compare (default value)
X 1 X X X X X X		Enable FRAR bit 1 compare

Note that the receiver and transmitter thresholds can only be changed when the Am79C30A/32A is in idle mode.

Address Status Register (ASR), Read Only

Bit	Logical '1'	Logical '0' (default value)
0	FRAR1/SRAR1 address recognized	No FRAR1/SRAR1 address match
1	FRAR2/SRAR2 address recognized	No FRAR2/SRAR2 address match
2	FRAR3/SRAR3 address recognized	No FRAR3/SRAR3 address match
3	FRAR4/SRAR4 address recognized	No FRAR4/SRAR4 address match
4–7	Not used, reads logical '0'	Not used, read logical '0'

D-Channel Status Register 1 (DSR1), Read Only

DSR1 has the following format:

Bit	Logical '1'	Logical '0' (default value)
0	Valid Address (VA) if the address decode logic is enabled or End of Address (EOA) if the address decode logic is disabled	No valid address
1	End of receive packet	Not end of packet
2	Packet reception in progress	Packet not being received
3	Loopback in operation at Am79C30A/32A	No loopback in operation at Am79C30A/32A
4	Loopback in operation at LIU	No loopback in operation at LIU
5	D-channel back-off not in operation	D-channel back-off in operation
6	End of valid transmit packet	Not end of transmit packet or no transmission
7	Current transmit packet has been aborted	No transmit packet abort

The DSR1 bits generate interrupts and are set/reset under the following conditions (in addition to a hardware reset or idle mode).

Bit	Generate Interrupt	Bit Set	Bit Reset
0	Yes if DMR3 bit 0=1	Two bytes after an opening flag if a VA is decoded or address recognition is disabled	When the microprocessor reads DSR1 or associated DRCR
	Yes if DMR1 bit 3=1	When a closing flag is received	When the microprocessor reads DSR1 or associated DRCR
2	No	One byte after the opening flag of any packet, valid or not	When a flag or an abort is received
3	No	When the operation is in progress	When the operation is not in progress
4	No	When the operation is in progress	When the operation is not in progress
5	No	When the operation is in progress	When the operation is not in progress
6	Yes if DMR3 bit 1=1	When the closing flag is transmitted	When the microprocessor reads DSR1 or when DTCR is loaded
7	No	When seven ones and a zero have been transmitted	When the microprocessor reads DSR1 or when DTCR is loaded

D-channel Status Register 2 (DSR2), Read Only

DSR2 has the following format:

Bit	Logical '1'	Logical '0' (default value)
0	Last byte of received packet	Not last byte of received packet
1	Receive byte available	Receive byte not available
2	Receive packet lost	Receive packet not lost
3	Last byte transmitted	Last byte not transmitted
4	Transmit buffer available	Transmit buffer not available
5	Mark idle detected (15 or more contiguous ones)	Mark idle not detected
6	Flag idle detected (more than 2 contiguous flags)	Flag idle not detected
7	Not used, reads logical '0'	Not used, reads logical '0'

The DSR2 bits generate interrupts and are set/reset under the following conditions (in addition to a hardware reset or idle mode):

Bit	Generate Interrupt	Bit Set	Bit Reset
0	Yes, if DMR3 bit 2 = 1	When last byte of a received packet is read from the DCRB	When the microprocessor reads the DSR2
1	Yes, if DMR3 bit 3 = 1	When DCRB contains one or more bytes of data	When DCRB is empty
2	Yes, if DMR3 bit 6 = 1	When two outstanding packets are received and not serviced and a third packet is received	When the microprocessor reads DSR2
3	Yes, if DMR3 bit 4 = 1	When the last byte of a transmit packet is transferred from the DCTB	When the microprocessor reads DSR2
4	Yes, if DMR3 bit 5 = 1	When the DCTB is available to be loaded with a data byte	When the DCTB is full
5	No	When 15 contiguous one bits have been detected in the incoming D-channel	When the first zero bit is detected on the incoming D-channel
6	No	When more than two contiguous flags are detected on the incoming D-channels not including a closing flag	When a non-flag character is detected on the incoming D-channel
7	Not used		

D-channel Error Register (DER), Read Only

The DER has the following format:

Bit	Logical '1'	Logical '0' (default value)
0	Received packet abort	No abort received
1	Non-integral number of bytes have been received	Integral number of bytes received
2	Collision detected	No error
3	FCS error	No error
4	Overflow error	No error
5	Underflow error	No error
6	Overrun error	No error
7	Underrun error	No error

The DER bits generate interrupts and are set/reset under the following conditions (in addition to a hardware reset):

Bit	Generates Interrupt	Bit Set	Bit Reset
0	Yes, if DMR2 bit 0 = 1	When seven consecutive ones are received within a packet (DSR1 bit 2 = 1)	When the µP reads the DER or associated DRCR
1	Yes, if DMR2 bit 1 = 1	Upon error condition after closing flag has been received or associated DRCR	When the µP reads the DER
2	Yes, if DMR2 bit 2 = 1	See section on collision detection	When the µP reads the DER or when DTCR is loaded
3	Yes, if DMR2 bit 3 = 1	If error occurs	When the µP reads the DER or associated DRCR
4	Yes, if DMR2 bit 4 = 1	If error occurs	When the µP reads the DER or associated DRCR
5	Yes, if DMR2 bit 5 = 1	If error occurs	When the µP reads the DER or associated DRCR
6	Yes, if DMR2 bit 6 = 1	If error occurs	When the µP reads the DER or associated DRCR
7	Yes, if DMR2 bit 7 = 1	If error occurs	When the µP reads the DER or when DTCR is loaded

DER bits 0, 1, 3, 4, 5, and 6 are set when the last byte of the associated packet is read from the D-channel Receive Buffer.

OSCILLATOR (OSC)

This interface sources the primary timing for the Am79C30A/32A and contains facilities for the connection of a 12.288 MHz clock, or a parallel resonant crystal and load capacitors.

APPLICATIONS

Figure 11— ISDN Feature Phone

This basic feature phone is the ISDN equivalent to the common analog phone. The Keypad can be a simple four by four single pole switch matrix or up to a six by six matrix to provide full key system features. The display option, illustrated, can be included in any of the applications shown in this section.

Figure 12 — ISDN Feature Phone with Parallel and Serial Data Ports plus other Peripherals

Access to the CCITT 'R' reference interface is illustrated via both the serial and parallel ports in the diagram. In addition, an example of several other peripherals interfaced to the microprocessor bus is shown. This application has the capability of supporting various voice options. Hence, the keypad is included in the figure even though it is not required for data only applications.

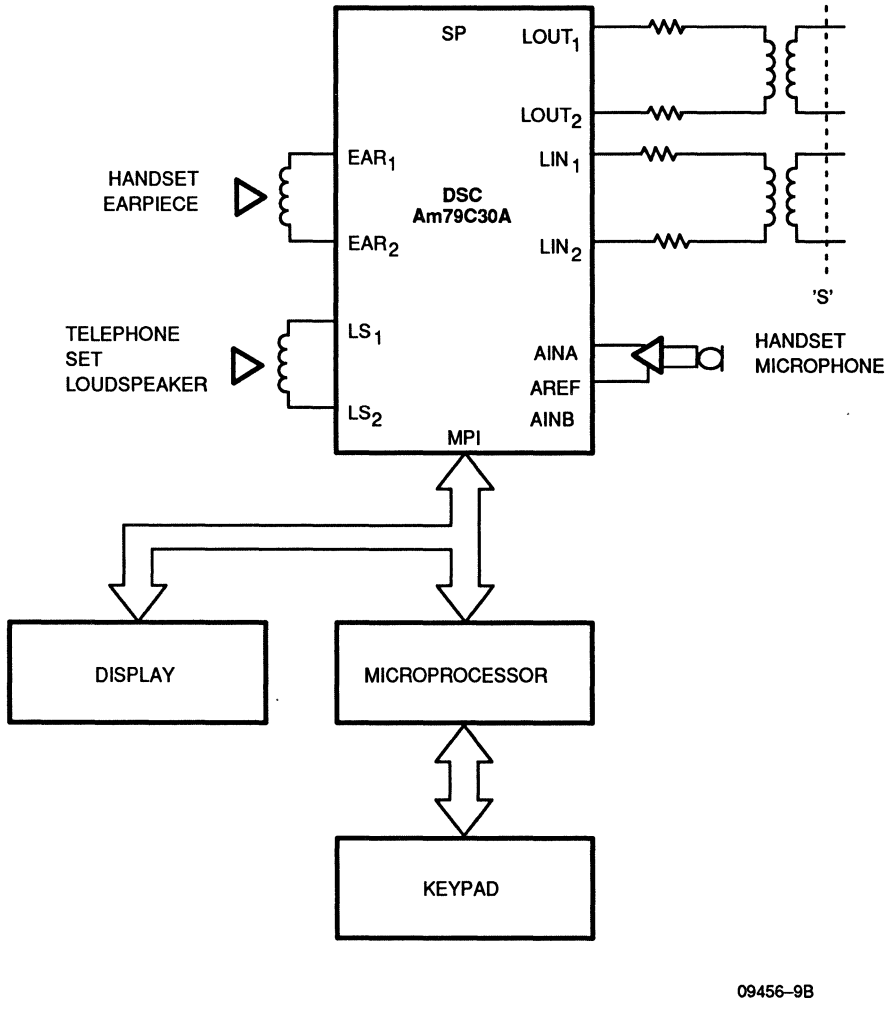


Figure 11. Feature Phone with Display

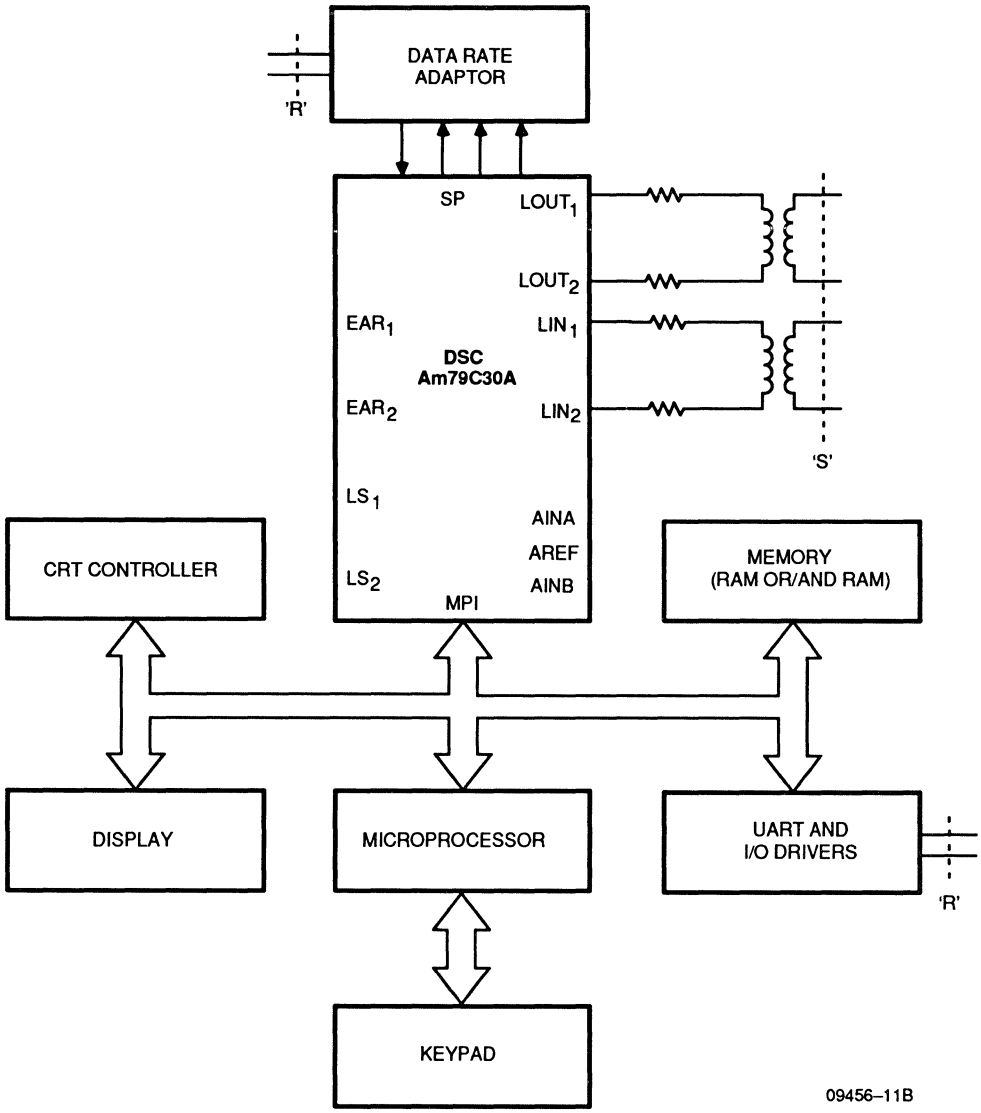


Figure 12. Feature Phone with Parallel and Serial Data Ports Plus Other Peripherals

ELECTRICAL AND SWITCHING CHARACTERISTICS

Absolute Maximum Ratings

Storage Temperature	–65° C to +150° C
Ambient Temperature with Power Applied	–55° C to +125° C
Supply Voltage to Ground Potential Continuous	0 V to +7.0 V
Lead Temperature (soldering, 10 sec)	300° C
Maximum Power Dissipation	1.5 W
Voltage from any pin to V_{SS}	V_{SS} –0.5 V to V_{CC} to +0.5 V
DC input/output current (except LS1, LS2)	10 mA
DC output current, LS1, LS2 only	100 mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

Operating Ranges

Commercial (C) Devices	
Operating V_{CC} range with respect to V_{SS}	4.75 V to 5.25 V
Ambient Temperature (T_A)	0° to +70° C

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC Characteristics (over operating ranges unless otherwise specified)

Parameter Symbol	Parameter Descriptions	Test Conditions	Min.	Max.	Unit
V_{IH}	Input high level except XTAL2		2.0	V_{CC} + 0.25	V
V_{IH2}	Input high level XTAL2		0.8 V_{CC}	V_{CC} + 0.25	V
V_{IL}	Input low level		V_{SS} –0.25	0.8	V
V_{OL}	Output low level	I_{OL} = 2 mA		0.4	V
V_{OH}	Output HIGH level	I_{OH} = –400 µA = –10 µA	2.4 0.9 V_{CC}		V
I_{OL}	Output leakage current	0 < V_{OUT} < V_{CC} Output in high Z state		±10	µA
I_{IL}	Input Leakage Current Digital Inputs LIN1/LIN2 XTAL2	0 < V_{IN} < V_{CC}		 ±10 ±200 TBD	 µA µA µA
C_I	Input capacitance digital input	temp = 25° C freq. = 1 MHz		10 (typical)	pF
C_O	Output capacitance, digital input/output	temp = 25° C freq. = 1 MHz		15 (typical)	pF

Parameter Symbol	Parameter Descriptions	Test Conditions	Min.	Typ.	Max.	Unit
I_{CC1}	V_{CC} supply current (Idle)	V_{CC} = 5.25 V; V_{IH} = V_{CC}; V_{IL} = V_{SS}; mode = IDLE; f_{MCLK} = 3.072 MHz; LIU receiver enabled; 'S' interface silent (INFO 0)		15	33	mA
I_{CC2}	V_{CC} supply current (Active; call setup)	V_{CC} = 5.25 V; V_{IH} = V_{CC}; V_{IL} = V_{SS}; mode = Active, data only; f_{MCLK} = 3.072 MHz; LIU receiver and transmitter enabled; 'S' interface activated with data on D-channel only; 'S' interface load = 50 ohms		20	35	mA
I_{CC3}	V_{CC} supply current (Active; voice mode)	V_{CC} = 5.25 V; V_{IH} = V_{CC}; V_{IL} = V_{SS}; mode = Active voice + data; f_{MCLK} = 3.072 MHz; LIU receiver/ transmitter enabled; 'S 'interface activated with data on D-ch & one B-ch; 'S' interface load = 50 ohms; AINA = –15 dBm0, 1 kHz sine wave; EAR1/EAR2 = –15 dBm0, 1 kHz tone driving 600 ohms		40	50	mA

AC Characteristics

V_{CC} = 5 V ±5%, V_{SS} = 0 V; TA = 0° C —>70° C, MCLK = 3.072 MHz

MAP Analog Characteristics (Am79C30A only)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Units
Z_{IN}	Analog input impedance AINA or AINB to AREF	–1.25 V < V_{IN} < +1.25 V f_{IN} < 4 kHz	200		Kohm
V_{IOS}	Allowable offset voltage at AINA or AINB	with respect to AREF pin	–5	+5	mv
V_{IR}	Analog input full scale reference level AINA or AINB	with respect to AREF pin	±1.25 (nominal)		Vpeak
V_{PR}	Analog output full scale reference level (PCM code = +3 dBm0)	EAR1 to EAR2 with RLOAD > 540 ohm and CLOAD < 100 pF, or LS1 to LS2 with RLOAD > 40 ohm and CLOAD < 100 pF	±2.5 (nominal)		Vpeak
Z_{LS}	Allowable Load LS1 to LS2		RLOAD > 40 ohms and CLOAD < 100 pF		
Z_{EAR}	Allowable Load EAR1 to EAR2		RLOAD > 540 ohms and CLOAD < 100 pF		

MAP Transmission Characteristics (Am79C30A only)

Parameter Description	Test Conditions	Min	Max	Units
Attenuation vs frequency relative	*50–60 Hz (Xmit only)	24.0		dB
to –10 dBm0 at 1020 Hz	<300Hz	–0.25		dB
(See Figures 13, 17)	0.3–3 kHz	–0.25	0.25	dB
	3–3.4 kHz	–0.25	0.9	dB
	3.4–3.6 kHz	–0.25		dB
	3.6–3.9 kHz	0.0		dB
	3.9 kHz–4 kHz	9.0		dB
Group delay variation with frequency	500–600 Hz		750	µs
at 0 dBm0 (See Figures 14, 18)	600–1000 Hz		380	µs
	1 K–2.6 kHz		130	µs
	2.6–2.8 kHz		750	µs
Gain tracking CCITT Method 2	+3 to –40 dBm0	–0.3	0.3	dB
(Tone at 1020 Hz)	–40 to –50 dBm0	–0.6	0.6	dB
(See Figures 15, 19)	–50 to –55 dBm0	–1.6	1.6	dB
Signal to total distortion CCITT	0 to –30 dBm0	35		dB
Method 2 (Tone at 1020 Hz)	–40 dBm0	29		dB
(See Figures 16, 20)	–45 dBm0	24		dB
Idle channel noise	Receiver		–75	dBm0
	Transmitter		–66	dBm0
Absolute gain	Relative to 0 dBm0 at 1020 Hz, RL > 540 ohms (See notes below.)	–0.3	+0.3	dB

The half channel parameters are specified above from AINA or AINB input pins to a B-channel for the transmit path and from a B-channel to EAR1/EAR2 pins measured differentially for the receive path. The parameters are applicable for both A- or U-law conversion. (A-law assumes psophometric filtering and U-law assumes c-message weighting.) All parameters are specified with the GR, X, R, GX, and GER filters disabled; STG filter enabled but programmed for infinite attenuation.

A 0 dBm0 signal is equivalent to 0.625 Vrms at AINA or AINB and to 1.25 Vrms measured differentially at EAR1/EAR2 or LS1/LS2, when programmed for A-law operation. In µ-law mode, 0 dBm0 is equivalent to 0.620 Vrms at AINA or AINB and to 1.24 Vrms at EAR1/EAR2 or LS1/LS2.

An external 1 Kohm resistor (±5%) and 2000 pf capacitor (±20%) are connected in series between the CAP1 and CAP2 pins.

*Valid only when high pass filter and auto zero bits are enabled in MAP Mode Register 2.

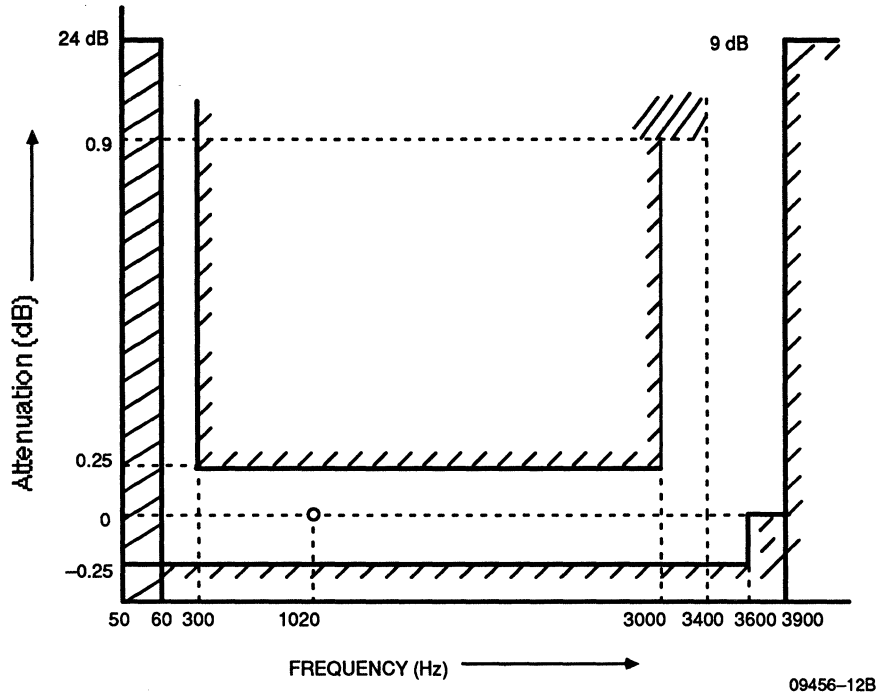


Figure 13. Attenuation/Frequency Distortion (Transmit)

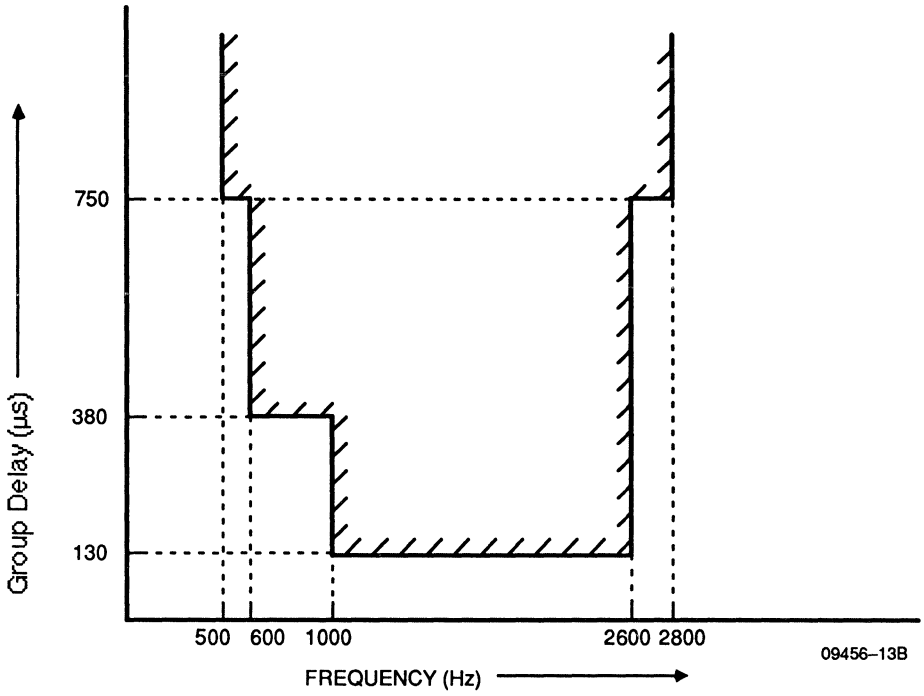


Figure 14. Group Delay Variation with Frequency (Transmit)

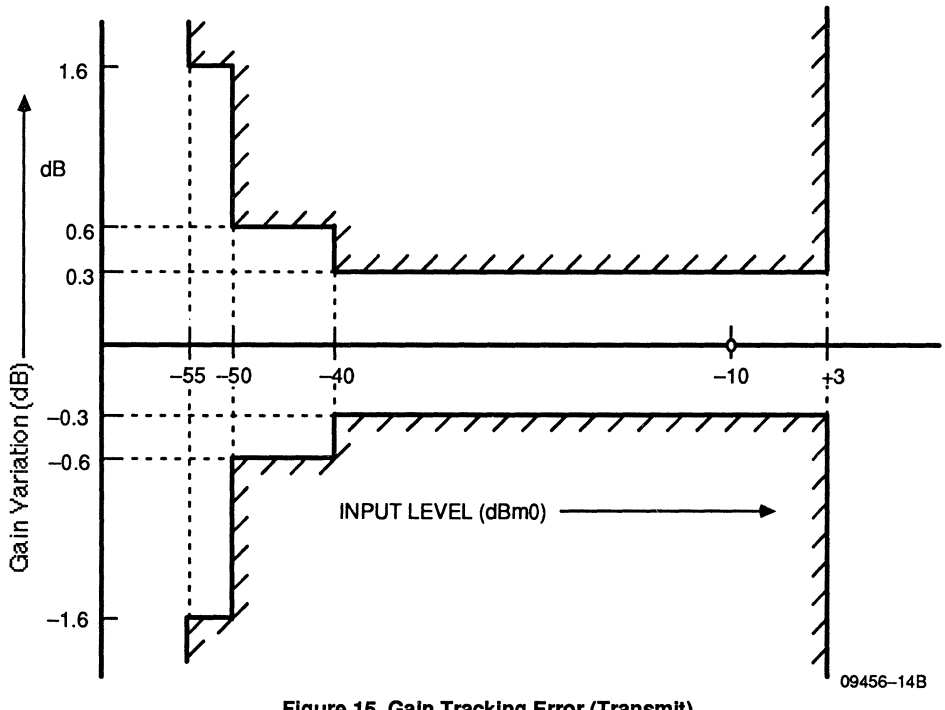


Figure 15. Gain Tracking Error (Transmit)
(CCITT Method 2 at 1020 Hz)

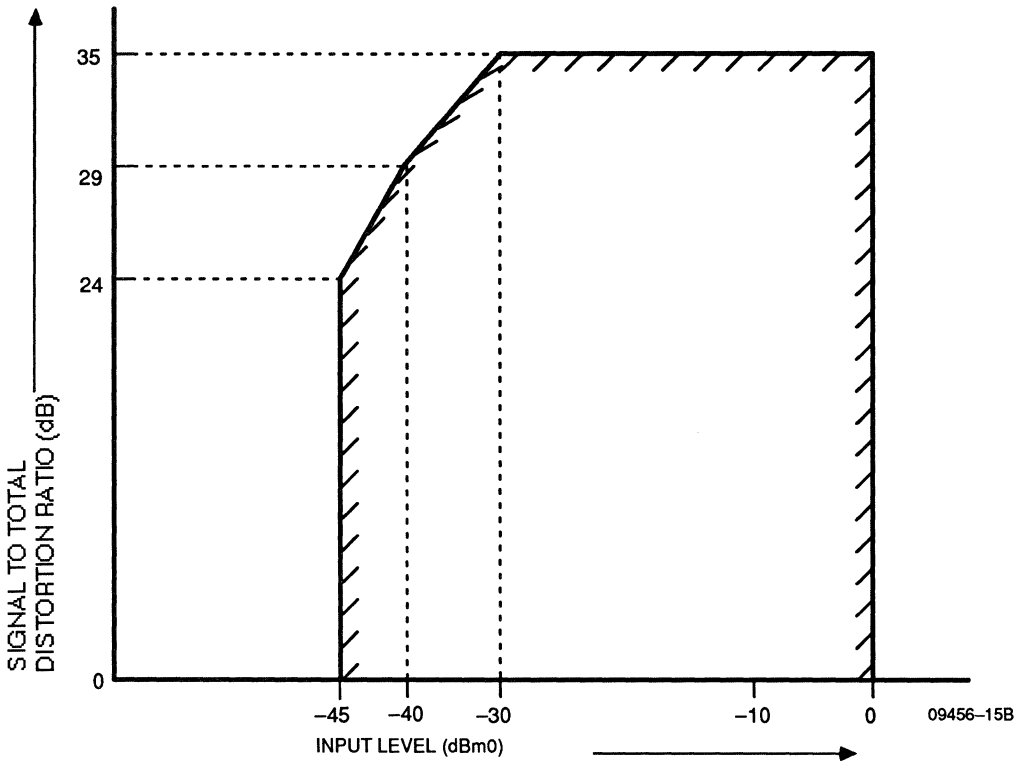


Figure 16. Signal to Total Distortion Ratio (Transmit) (CCITT Method 2 at 1020 Hz)

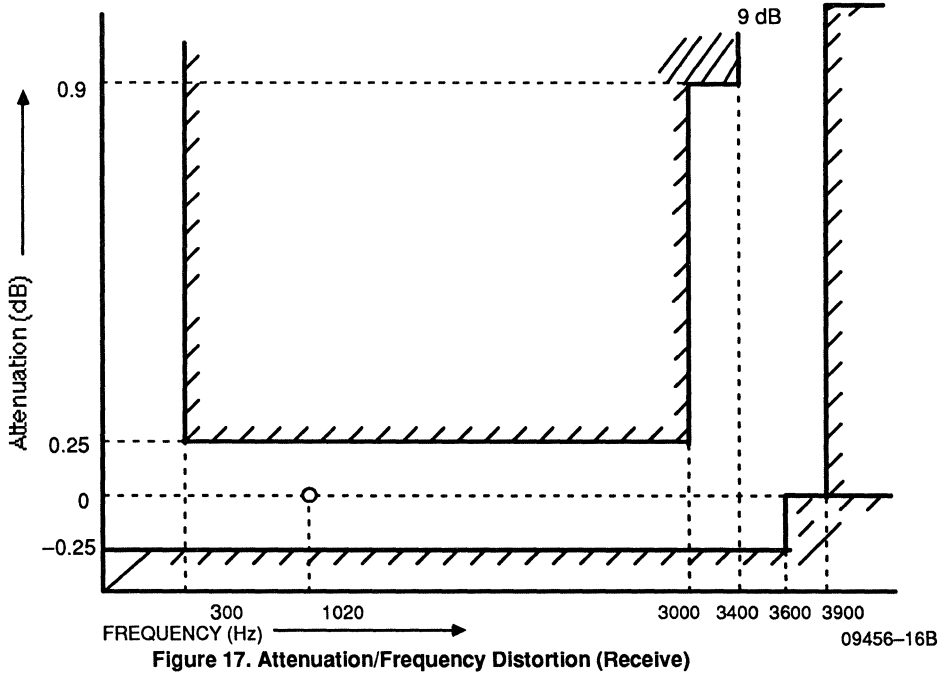


Figure 17. Attenuation/Frequency Distortion (Receive)

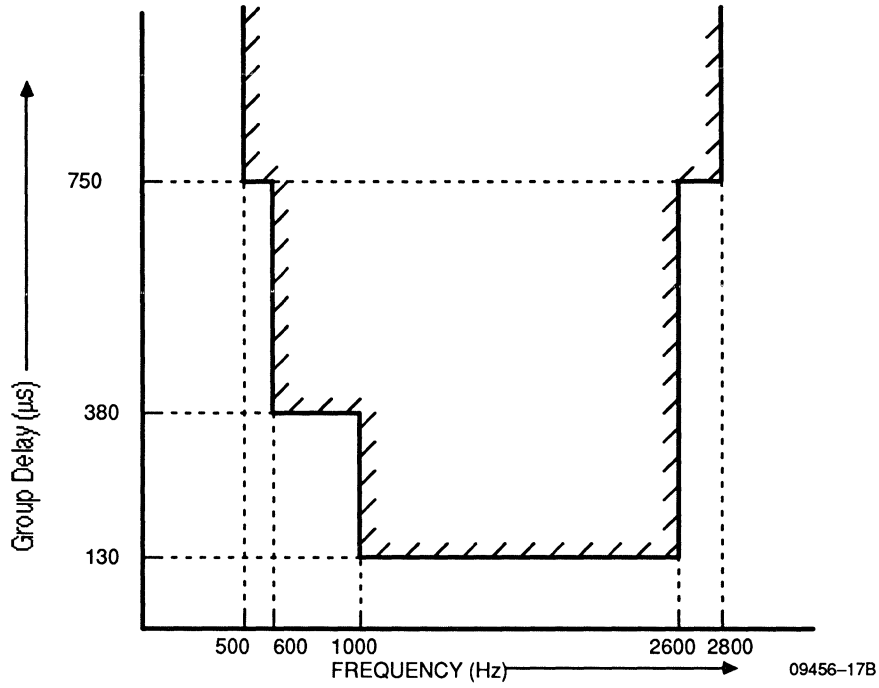


Figure 18. Group Delay Variation with Frequency (Receive)

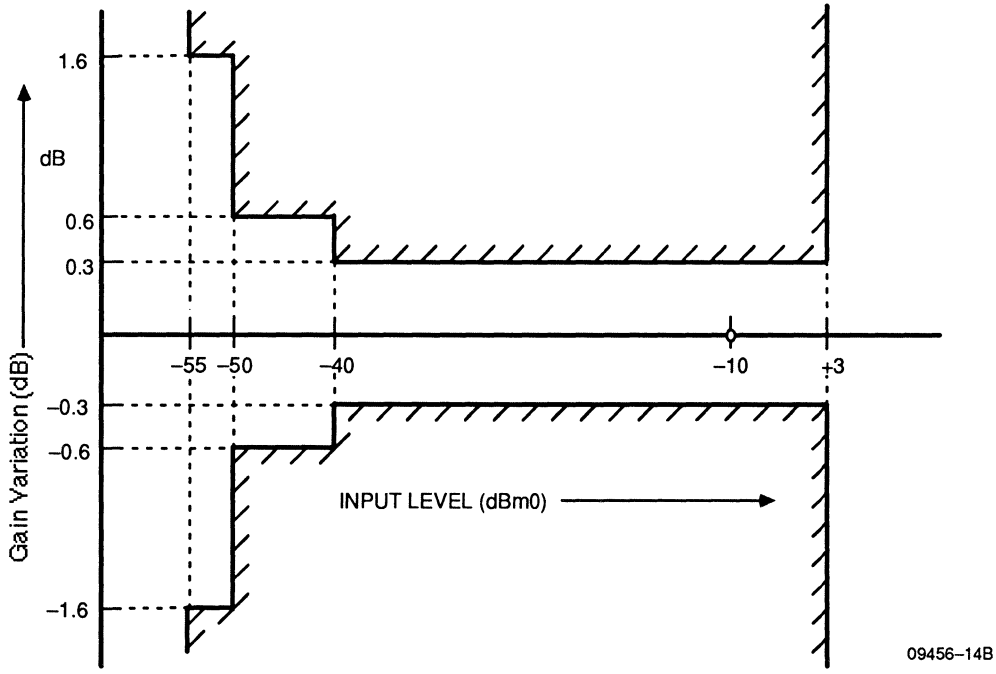


Figure 19. Gain Tracking Error (Receive)
(CCITT Method 2 at 1020 Hz)

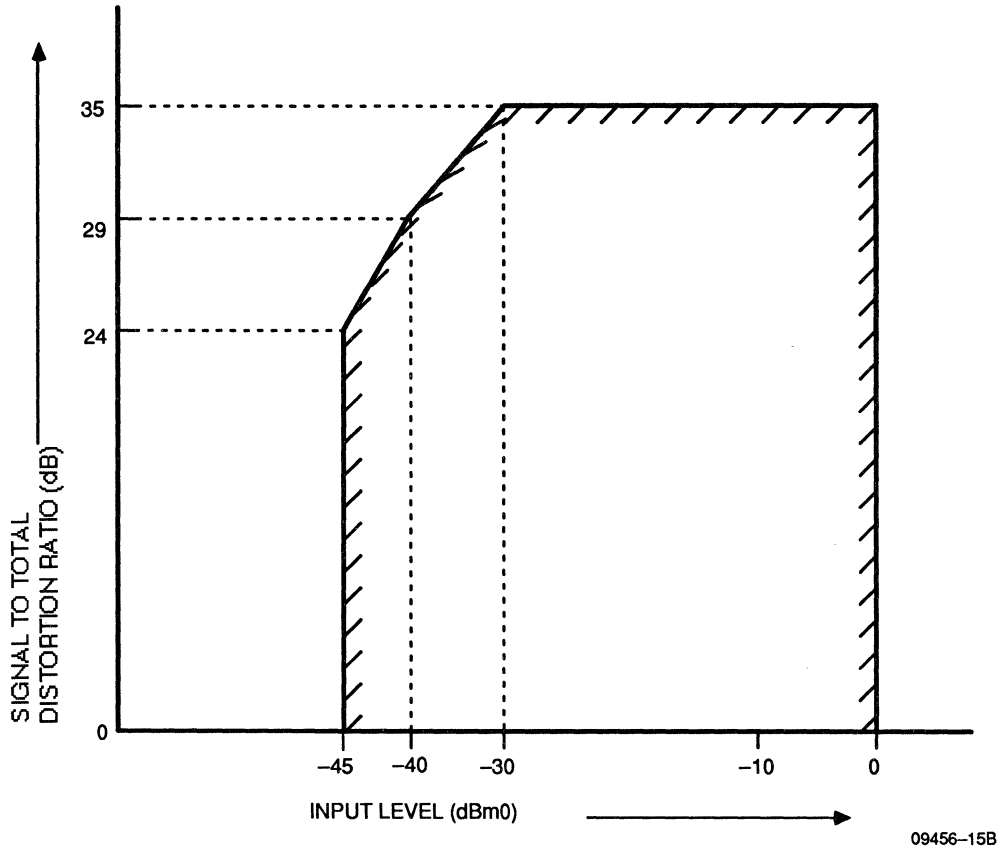


Figure 20. Signal to Total Distortion Ratio (Receive)
(CCITT Method 2 at 1020 Hz)

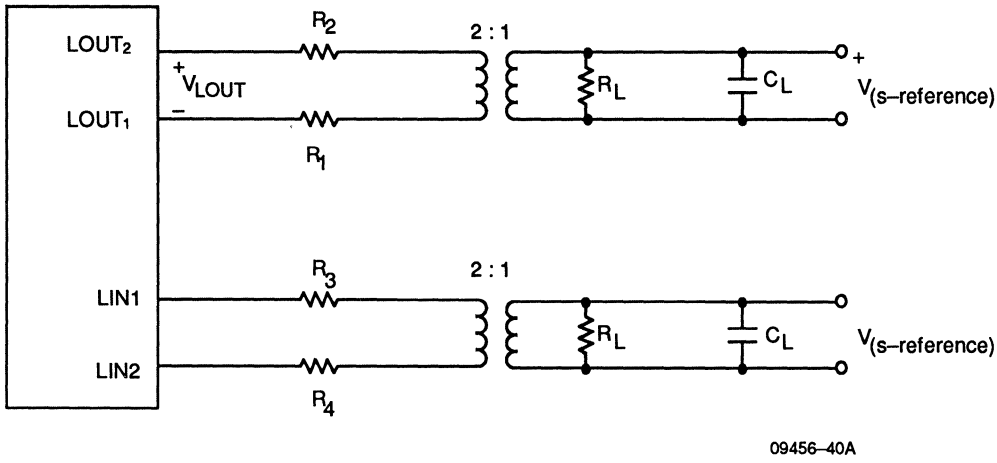
LIU Characteristics

All of the parameters below are measured at the chip terminals and are consistent with 2:1 transformers.

Parameter Symbol	Parameter Description	Min	Typ.	Max	Units
V_{LOUT}	Output mark amplitude measured between LOUT2 and LOUT1 (Note 1)	2.210	2.326	2.442	V
V_{LIN}	Receivable input level measured between LIN2 and LIN1 with noise added as specified by CCITT I.430 section 8.6.2.1. (Note 2)	530		1800	mV
Z_{OUT}	Output impedance measured between LOUT2 and LOUT1 Spacing condition	20			Kohm
Z_{IN}	Input Impedance measured between LIN2 & LIN1	20			Kohm
J	Timing Extraction Jitter on LOUT		–7	+7	%
PD	Total Phase Deviation (LOUT with respect to LIN)		–7	+15	%
PU	Pulse Unbalance measured between LOUT2 & LOUT1 (Note 1)	–5		+5	%
PW	Output Pulse Width measured between LOUT2 & LOUT1 (Note 1)	4.7	5.2	5.7	µs

Note 1: See the equivalent test load circuit and pulse template in Figures 22 and 23.

Note 2: The 530 mV receive input level is equivalent to 9.0 dB of attenuation from a nominal transmit level when measured at the LIN pins. Allowing 0.5 dB loss in the isolation transformer and 1.0 dB loss in the input isolation resistors, this level will guarantee compliance to the CCITT receiver sensitivity spec of 7.5 dB when measured at the 'S' reference point.



V (S-Interface): Transmitter output at the 'S' interface reference point.

RL is the termination impedance at the 'S' interface.

CL is the effective capacitance at the 'S' interface (see I.430 B.5.1.2).

R1 and R2 are the transmitter output series resistors; their value depends upon the characteristics of the pulse transformer. (See equations in Figure 24.) R3 and R4 are required for multi-point operation to prevent loading of the line when power is removed from the terminal.

Figure 21. System Interface to LIU

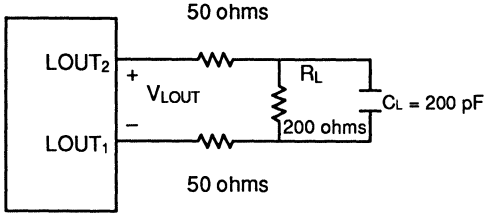


Figure 22. Equivalent Test Load Conditions

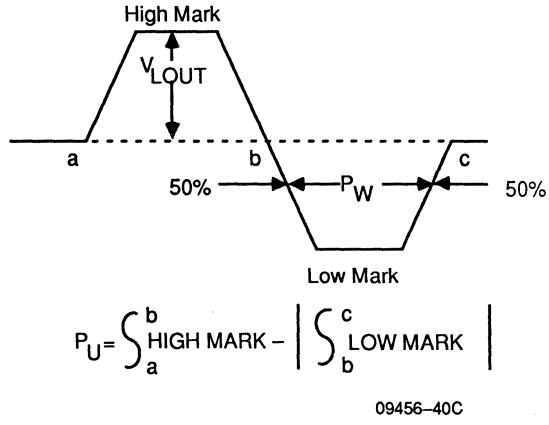
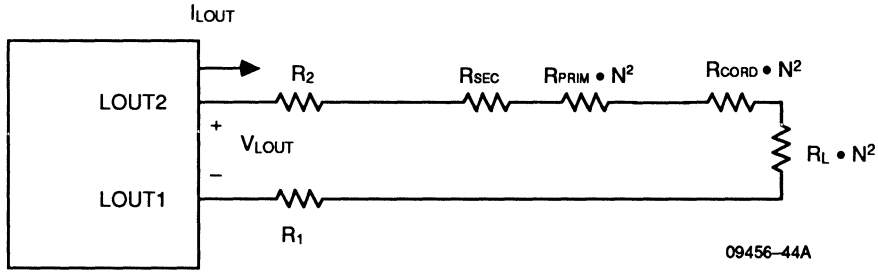


Figure 23. Differential Output Signals Between LOUT2 and LOUT1 Using the Test Circuit In Figure 22

Series Resistor Calculations



Rsec is the DC impedance of the transformer secondary. (IC side of transformer)
Rprim is the DC impedance of the transformer primary. (Line side of transformer)
R_{CORD} is the DC impedance of the TE connecting cord typically 4–6 ohms
N is the transformer turns ratio. (N = 2 for Am79C30A/32A)
R_L is the 'S' interface line impedance. (50 ohms)
I_{LOUT} is the desired load current for the CCITT transmission templates (7.5 mA for 50 ohm line).
V_{LOUT} is the nominal output voltage from the DSC/IDC line driver.

$$I_{LOUT} = \frac{V_{LOUT}}{R1 + R2 + Rsec + (Rprim \bullet N^2) + (RL \bullet N^2) + (R_{CORD} \bullet N^2)}$$

$$R1 + R2 = \frac{V_{LOUT}}{I_{LOUT}} - Rsec - (Rprim \bullet N^2) - (RL \bullet N^2) - (R_{CORD} \bullet N^2)$$

Let R1 = R2

$$R1 = R2 = 1/2 \left\{ \frac{V_{LOUT}}{I_{LOUT}} - Rsec - (Rprim \bullet N^2) - (RL \bullet N^2) - (R_{CORD} \bullet N^2) \right\}$$

N = 2
RL = 50 ohms
V_{LOUT} = 2.326 V
I_{LOUT} = 7.5 mA

$$\mathbf{R1 = R2 = 55.067 - 1/2 \{ Rsec + (4 \bullet Rprim) + (4 \bullet R_{CORD}) \}}$$

This equation should be used to determine the value of R_1 and R_2 for the particular transformer used by each customer.

Figure 24. Equivalent DC Circuit at LOUT Pins for calculation of R_1 & R_2

Microprocessor Read/Write Timing

Microprocessor Read Timing

Parameter Symbol	Parameter Description	Min	Max	Units
t_{RLRH}	$\overline{RD}$ pulse width	200		ns
t_{RHRL}	Read recovery time (Note 1,2)	200		ns
t_{CLRL}	$\overline{CS}$ low to $\overline{RD}$ low	0		ns
t_{RHCH}	$\overline{RD}$ high to $\overline{CS}$ high	0		ns
t_{ADRL}	Address to $\overline{RD}$ low	20		ns
t_{RHAD}	Address hold after $\overline{RD}$ high	10		ns
t_{RLDA}	$\overline{RD}$ low to data available		80	ns
t_{RHDZ}	$\overline{RD}$ high to data high Z		50	ns

No timing relationship is required between ADDR and $\overline{CS}$ pulses.

The $\overline{CS}$ can be tied low.

Read and write indirect register operations cannot be mixed without at least one write command register operation between them.

Note 1: The read/write recovery time of 200 ns holds in all cases except when a write command register operation is followed by a read data register operation when accessing the MAP coefficient RAM. This operation requires a minimum recovery time of 450 ns.

Note 2: Successive reads of the D-channel Receive Buffer require a minimum cycle time (t_{RLRH} + t_{RHRL}) of 480 ns.

Microprocessor Write Timing

Parameter Symbol	Parameter Description	Min	Max	Units
t_{WLWH}	$\overline{WR}$ pulse width	200		ns
t_{WHWL}	Write recovery time (Note 1)	200		ns
t_{CLWL}	$\overline{CS}$ low to $\overline{WR}$ low	0		ns
t_{WHCH}	$\overline{WR}$ high to $\overline{CS}$ high	0		ns
t_{ADWL}	Address to $\overline{WR}$ low	20		ns
t_{WHAD}	Address hold after $\overline{WR}$ high	10		ns
t_{DAWH}	Data available to $\overline{WR}$ high	100		ns
t_{WHDZ}	$\overline{WR}$ high to data high Z	10		ns

No timing relationship is required between ADDR and $\overline{CS}$ pulses.

The $\overline{CS}$ can be tied low.

Read and write indirect register operations cannot be mixed without at least one write command register operation between them.

Note 1: The read/write recovery time of 200 ns holds in all cases except when a write command register operation is followed by a read data register operation when accessing the MAP coefficient RAM. This operation requires a minimum recovery time of 450 ns.

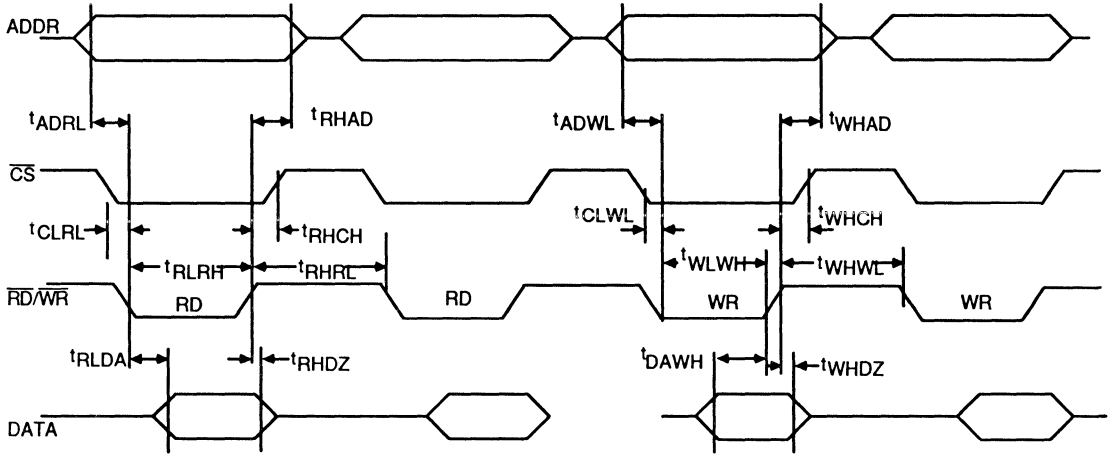


Figure 25. Microprocessor Read/Write Timing

$\overline{INT}$ Timing

Interrupt Timing

Parameter Symbol	Parameter Description	Min	Max	Units
t_{INTC}	$\overline{INT}$ cycle time	125		µs
t_{REC}	$\overline{INT}$ recovery time	500		ns

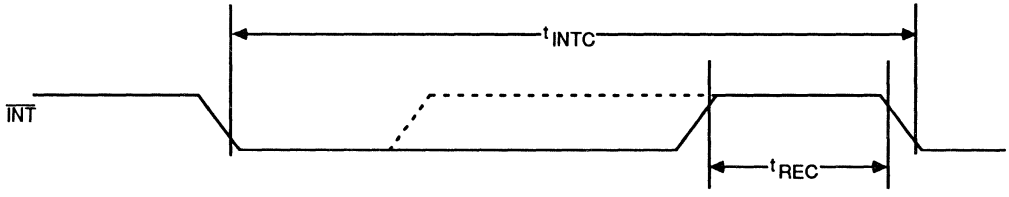


Figure 26. $\overline{INT}$ Timing

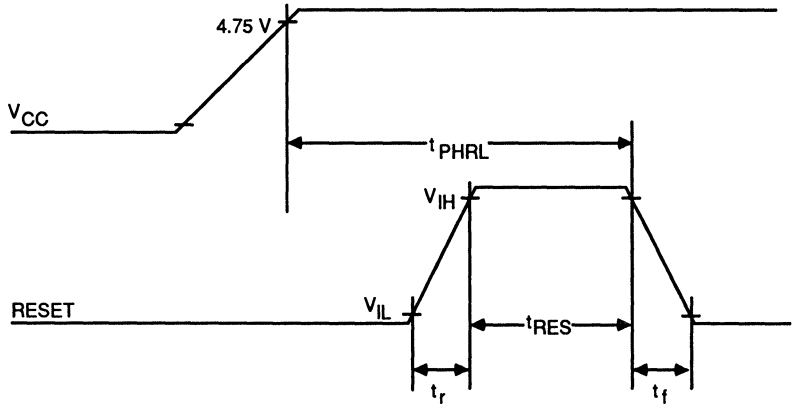
Reset and Hookswitch Timing

Reset Timing

Parameter Symbol	Parameter Description	Min	Max	Units
t_{RES}	Reset pulse width	1		µs
t_{PHRL}	Power stable to reset low	1		µs
t_f	Reset transition fall time		1	ms
t_r	Reset transition rise time		20	µs

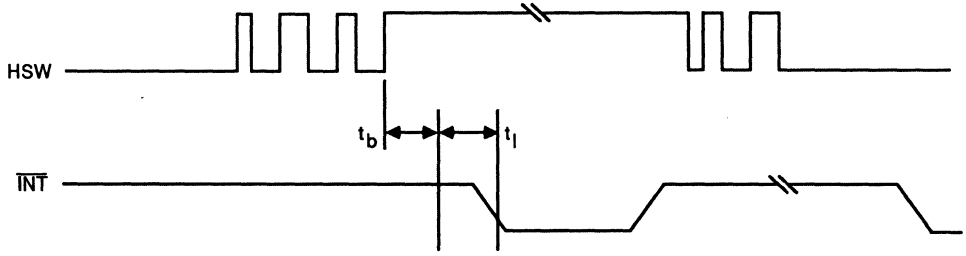
Hookswitch Timing

Parameter Symbol	Parameter Description	Min	Max	Units
t_b	Debounce time	16	16.25	ms
t_1	HSW detected to $\overline{INT}$ Delay	0	370	µs



09456–22B

Figure 27. Reset Timing



09456A–23

Figure 28. Hookswitch Debounce Timing

OSC (XTAL2) Timing

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Units
t_{CLCL}	Oscillator period		81.374	81.387	ns
t_{CH}	High Time		33		ns
t_{CL}	Low Time		33		ns
t_{CLCH}	Rise Time			10	ns
t_{CHCL}	Fall Time			10	ns

Frequency = 12.288 MHz ± 80 ppm.

MCLK Timing

Parameter Symbol	Parameter Description		Test Conditions	Min	Max	Units
t_D	XTAL2 VCC/2 to MCLK VCC/2		MCLK Load < 80pF		60	ns
t_{RISE}	Rise Time		MCLK Load < 80pF		15	ns
t_{FALL}	Fall Time		MCLK Load < 80pF		15	ns
t_{PWH}	High Pulse Width	12.288 MHz	MCLK Load < 80pF	33		ns
		6.144 MHz		73		ns
		4.069 MHZ		114		ns
		3.072 MHz		155		ns
t_{PWL}	Low Pulse Width	12.288 MHz	MCLK Load < 80pF	33		ns
		6.144 MHz		73		ns
		4.096 MHz		114		ns
		3.072 MHz		155		ns

The MCLK output can only change modes every 12 OSC cycles (that is, when all four of the MCLKs are synchronized on their low-to-high transitions).

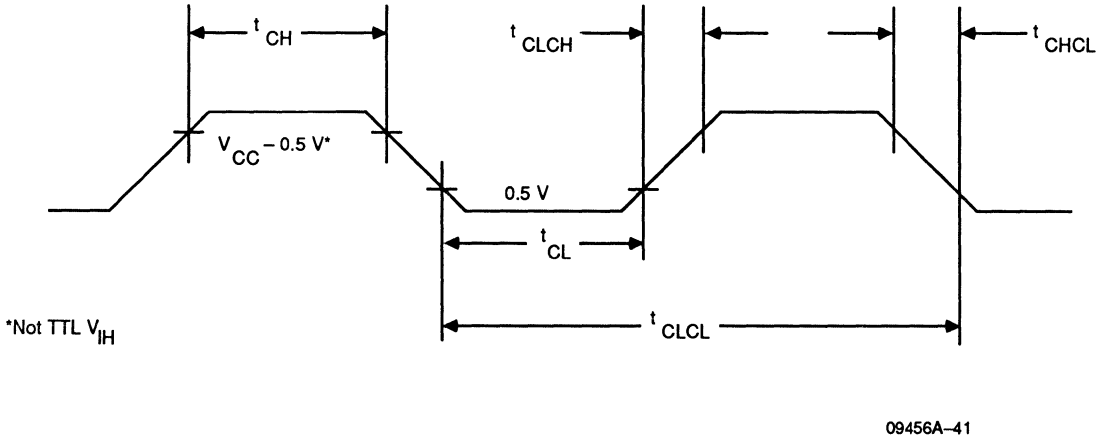


Figure 29. External Clock Driver (XTAL2) Timing

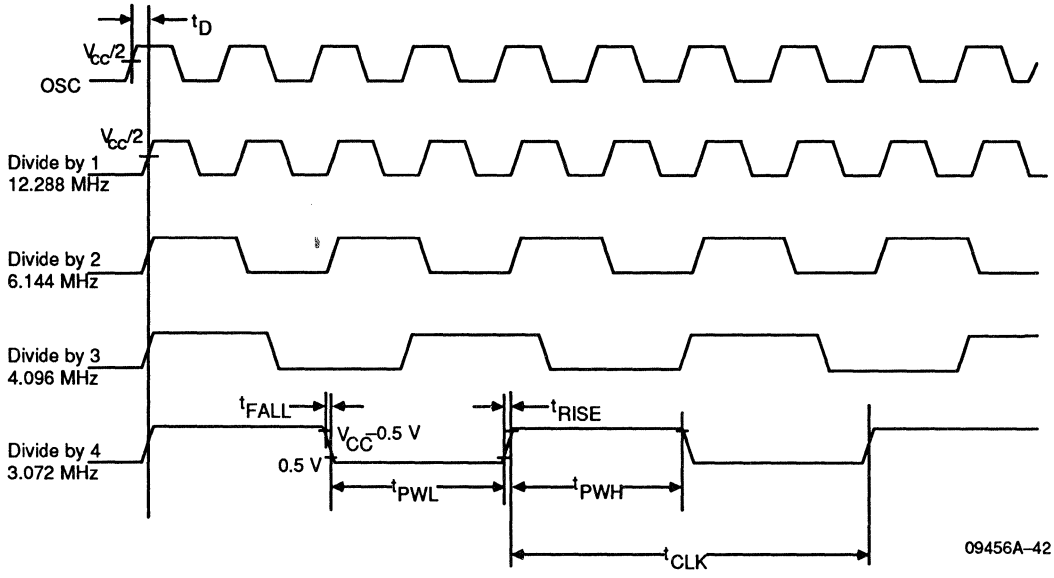
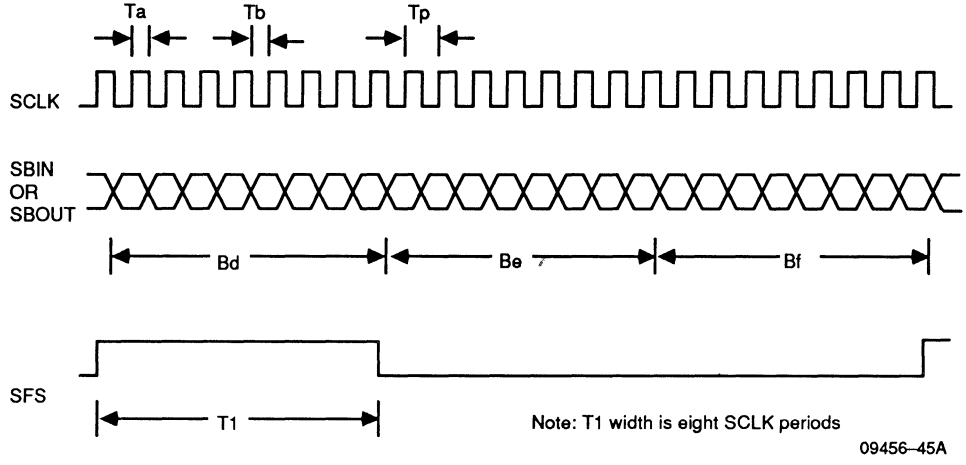


Figure 30. OSC/MLK Timing

Serial Port Timing

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Units
Tp*	SCLK		5.025	5.392	us
Ta	High Time		2.594	2.615	us
Tb*	Low Time		2.431	2.777	us
trise	SCLK rise time	SCLK Load < 80pF		20	ns
tfall	SCLK fall time	SCLK Load < 80pF		20	ns
t_{MCSC}	MCLK to SCLK @6.144 MHz	MCLK Load < 80pF SCLK Load < 80pF		60	ns
t_{CHFS}	SCLK high to Frame Sync		50	250	ns
t_{CLDO}	SBOUT Data Available	SBOUT/SFS Load = 80 pF	50	250	ns
t_{DICH}	SBIN Set up time		200		ns
t_{CHDZ}	SBIN Hold Time		0		ns

*The frequency of SCLK is always f_{XTAL2} / 64. Tp and Tb are based on this SCLK frequency, but include a ± 163 nsec allowance for internal phase lock loop correction.



Note: SBIN data is sampled on the rising edge of SCLK; SBOUT data is changed on the falling edge of SCLK

Figure 31. Serial Port Input/Output Timing

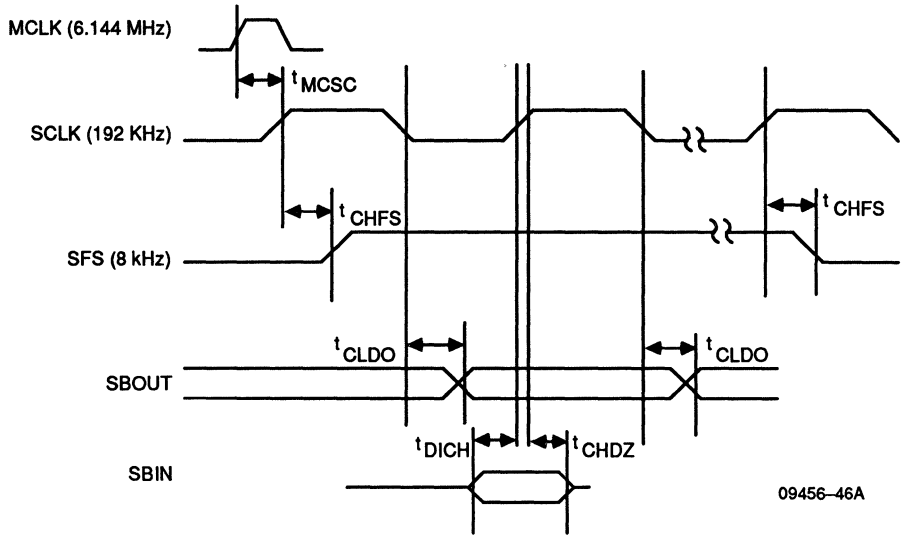
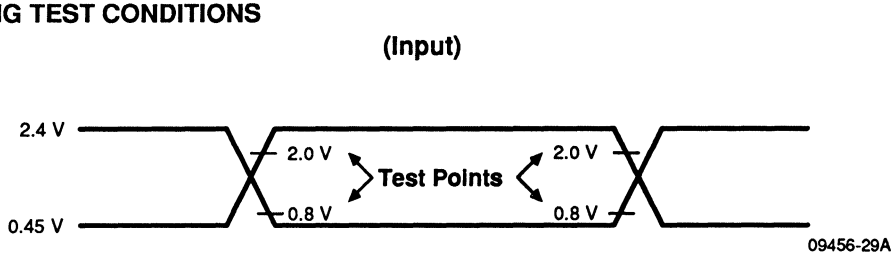


Figure 32. Serial Port MCLK/SCLK/SFS Timing

SWITCHING TEST CONDITIONS



AC testing inputs are driven at 2.4 V for a logical one and 0.45 V for a logical zero. Timing measurements are made at 2.0V and 0.8V for a logical one and a logical zero respectively.

Figure 33. Switching Test Input/Output Waveform

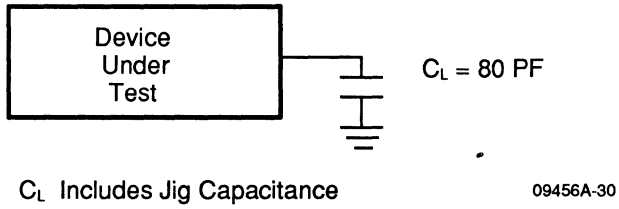


Figure 34. Switching Test Load Circuit

Key Design Hints For The DSC/IDC

Due to the high level of integration of the Am79C30A/32A DSC/IDC, it is easy to overlook important design information when reading the datasheet. The following list of key design hints have been compiled to streamline the design process. A comprehensive series of ISDN application notes and tutorials is available from Advanced Micro Devices; please contact an AMD sales office or factory for current information.

1) The AREF pin MUST be used to bias the AINA and AINB inputs. There is a datasheet parameter Vios which states that the analog inputs must be biased to within 5 mV of AREF. AREF is NOMINALLY 2.4 volts; normal device-to-device variation will exceed the 5 mV Vios specification. If a voltage other than AREF is used, transmission performance at very low signal levels will be degraded.

2) The recommended method of biasing the AINA and AINB inputs is to use a 15k-100k ohm resistor between the input and AREF. The signal source should be AC-coupled to the analog input. Care should be taken such that the RC formed by the biasing resistor and blocking capacitor does not distort the input signal. A 3-dB point below 10 Hz is recommended.

3) The AREF output must not be loaded with a capacitor, since it may cause the internal buffer amplifier to become unstable. For some applications involving significant gain external to the DSC, the AREF output may require a simple RC noise filter. In this case, the AREF output should be isolated from the capacitor by a resistance of greater than 1k ohms to ensure stability.

4) The DSC/IDC should be provided with decoupling capacitors, situated as closely as possible to the package power leads. In general 0.1 microfarad ceramic capacitors are sufficient, but bulk decoupling capacitors will be required if the LS1 and LS2 loudspeaker outputs are driving a heavy load.

5) The DSC/IDC is constructed on a single substrate, and therefore the device power pins must not be from separate supplies. If there is a DC offset between the analog and digital power supply pins, excessive current may flow through the device substrate.

6) The LS1, LS2 and EAR1, 2 outputs are intended to be used differentially. Although it is possible to use only a single output, the rejection of power supply noise and internal digital noise is improved if the outputs are used differentially.

7) It is necessary to observe the maximum loading specification for the LS and EAR outputs. When used differentially, the EAR outputs must see a minimum of 540 ohms between them. Similarly, the LS outputs must see a minimum of 40 ohms. The maximum capacitive loading in either case is 100 pF.

8) The LS and EAR outputs do not need to be "matched" to the load. The LS and EAR outputs are voltage drivers, and do not assume the presence of any particular load impedance. As long as the maximum loading specification is met, the LS and EAR outputs will function satisfactorily. In some cases an external resistor may be used to centre the desired output volume, for instance while driving a 150 ohm earpiece with the EAR outputs.

9) It is not recommended to unnecessarily access the ATGR, FTGR, GX, GR, GER, STG, X, R, MMR1, and MMR2 registers while audio is active. When the microprocessor accesses these registers it may interfere with the MAP calculations, resulting in degraded transmission performance or audible noise.

10) When using programmable gains and filters in the MAP, it is necessary to consider dynamic range effects such as truncation error and clipping. In case of question in any particular application, please contact AMD applications staff for assistance.

11) All MAP tone generators are referenced with respect to the +3 dBm0 overload voltage, i.e., a 0 dB tone yields a +3 dBm0 output. Care must be taken to avoid clipping when adding tones to signals, as for example when generating DTMF waveforms.

12) The RC connected to CAP1/CAP2 must be situated as close as possible to the DSC package, to reduce the amount of noise coupled in from other signal traces.

13) It is necessary to observe the XTAL2 frequency accuracy requirement of 12.288 MHz ± 80 ppm. Since crystals from different manufacturers will vary, it is necessary to measure the DSC oscillator output frequency at the MCLK pin and if necessary, adjust the value of the crystal load capacitors as part of the initial design procedure. An application note of oscillator considerations is available from AMD.

14) If driving the XTAL2 pin with the external oscillator, it is necessary to observe the datasheet input voltage and rise/fall time requirements. Note that the XTAL2 levels are not TTL-compatible.

15) The DSC, as any sensitive analog device, requires care in board layout. An application note of DSC board layout hints is available from AMD.

16) The sidetone path defaults to –18 dB attenuation. If it is desired to disable the sidetone path, it is necessary to enable the sidetone block and program it for infinite attenuation.

17) The LIU transformers, series resistors, and IC LIU output drivers must be considered as a functional

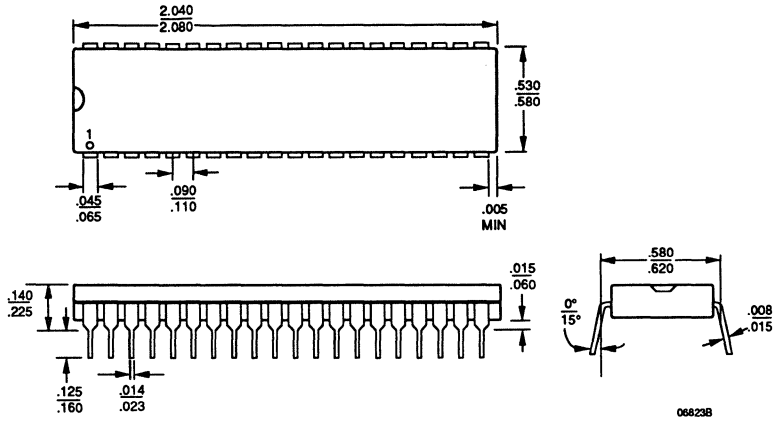
unit. Transformers that meet CCITT I.430 requirements with other transceivers are not necessarily appropriate for use with the DSC, and vice versa. An application note of DSC transformer and series resistor considerations is available from AMD.

18) Interrupts should be masked when reading or writing any indirect or multi-byte DSC registers, to prevent the possibility of an interrupt occurring and destroying the contents of the Command Register.

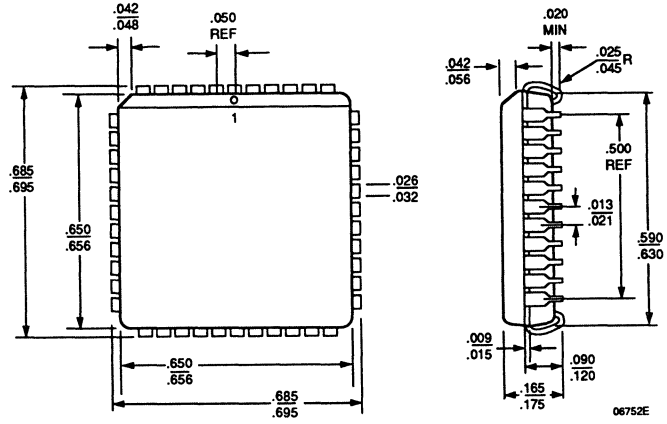
19. If the MAP is disabled, the EAR, AREF, and LS outputs are high impedance. If the MAP is enabled, the unselected audio output is high impedance.

PHYSICAL DIMENSIONS

PD 040



PL 044



Preliminary

Am79C401

Integrated Data Protocol Controller (IDPC)

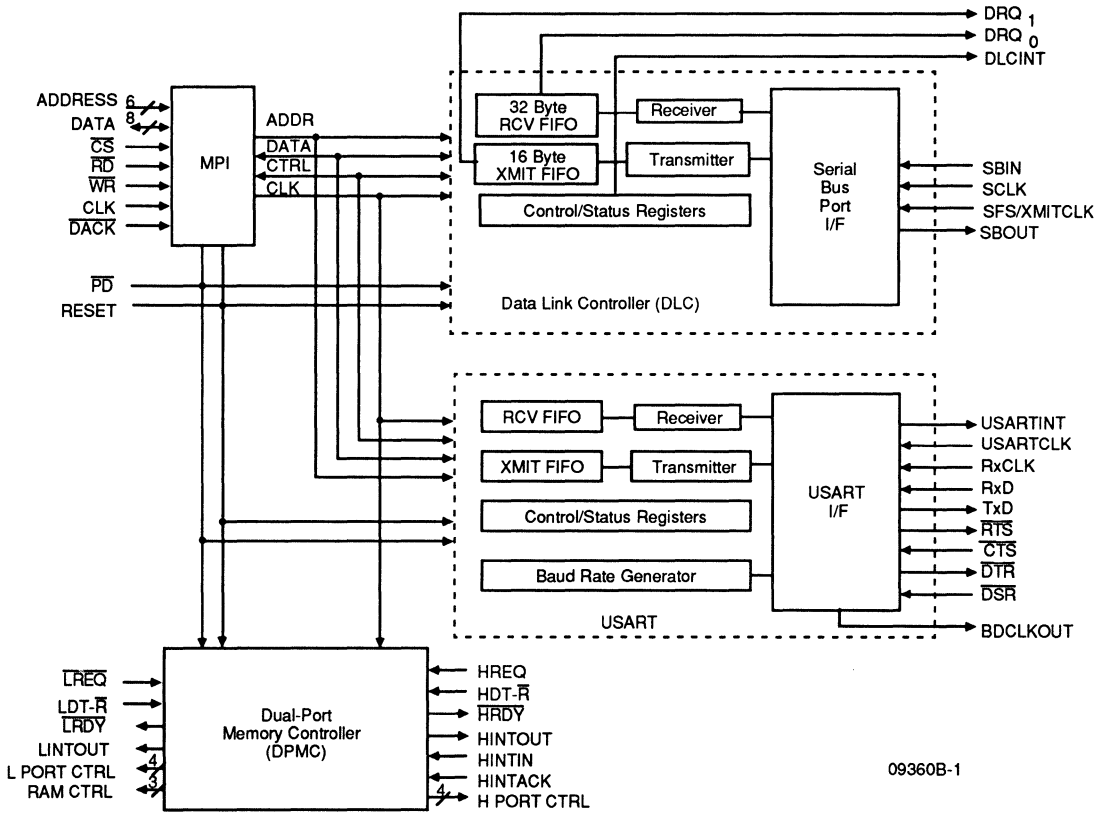
Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

Data Link Controller

- Full-featured bit-oriented communication controller supporting HDLC, SDLC, LAPB, LAPD, and DMI
- Data transfer rate: 2.048 Mb/s
- 32-byte receive FIFO and 16-byte transmit FIFO with programmable thresholds and DMA handshakes
- Multiple (four plus broadcast) address recognition modes
- Multiplexed serial interface with up to 31 8-bit channels or non-multiplexed serial interface
- Local and Remote Loopback Modes
- Transparent (Protocol-Free) Mode
- 56 kb/s Mode

BLOCK DIAGRAM



Publication # 09360 Rev. C Amendment /0
Issue Date: June 1989

DISTINCTIVE CHARACTERISTICS (continued)

USART

- Superset of Industry-Standard 8250 UART features
- Four-byte transmit/receive FIFOs
- Special character recognition (up to 128 programmable)
- Synchronous mode provides a transparent serial data path
- Local Loopback mode

Dual-Port Memory Controller

- Memory bus arbitrator provides dual-port access to standard low-cost static RAM
- Programmable inter-processor interrupts support RAM-based inter-processor mailboxes

Microprocessor Interface

- 8-bit non-multiplexed data bus
- Operates with 12.5 MHz 80188 processor with zero wait states

General Features

- Compatible with Am79C30A DSC and Am79C32A IDC
- CMOS technology, single +5 V supply
- Power-down mode
- 68-pin PLCC

GENERAL DESCRIPTION

The Am79C401 Integrated Data Protocol Controller (IDPC) and companion software product, the Am79LLD401 IDPC Low-Level Driver (LLD), provide the essential building blocks for construction of a variety of communications systems. When combined with ROM, RAM, a microprocessor, and the physical layer transceiver, a complete ISDN, X.25, SNA, or similar system can be constructed. The IDPC is design-compatible with existing AMD communication components such as the Am79C30A Digital Subscriber Controller (DSC) and Am79C32A ISDN Data Controller (IDC) and off-the-shelf microprocessors such as the 80188.

The IDPC contains hardware and software support features for use in a single-processor environment (such as a terminal adapter for an ISDN network) or a multiprocessor application (such as a communication interface for a PC or integrated voice/data work station application). For multi-processor applications, the IDPC controls access to an external "shared" static RAM by arbitrating simultaneous shared memory requests and supports an interprocessor interrupt scheme.

A companion software product, the Am79LLD401 IDPC LLD, provides isolation of the various hardware functions from the higher levels of packet protocol software. The LLD can be used with any bit-oriented protocol, including AmLink™, which is AMD's LAPD/LAPB implementation. Additionally, AMD's AmLink3™ package offers a complete implementation of the X.25 Packet Layer (Layer 3) protocol.

For ISDN D-channel applications, a similar function is provided by the Am79C30A DSC and Am79LLD30A LLD software.

The interfaces presented by the Am79LLD30A DSC LLD and the Am79LLD401 IDPC LLD use the same primitives so that both the D-channel and the B-channel can use the same layer-2 software. Both LLDs provide a hardware-independent interface to upper-layer protocols such as AmLink LAPD.

Functionally, the IDPC consists of four sections: a Data Link Controller (DLC), a Universal Synchronous/Asynchronous Receiver/Transmitter (USART), a Dual-Port Memory Controller (DPMC), and a Microprocessor Interface (MPI).

Data Link Controller (DLC)

The DLC shown in the Block Diagram is a high-speed, bit-oriented protocol processor that supports either multiplexed or non-multiplexed data rates up to 2.048 Mb/s.

The DLC provides full-duplex (simultaneous transmit and receive) data transfer between the chip's Serial Bus Port (SBP) and internal parallel bus. Through the use of a 32-byte receive FIFO, a 16-byte transmit FIFO, and two external DMA channels, the DLC provides efficient movement of data to and from external memory and the SBP (network interface).

The DLC supports data transfers via DMA, interrupts, or polled I/O. The use of the FIFO buffers minimizes interrupt latency and frequency of interrupts.

The DLC has several programmable modes of operation which include:

- Non-multiplexed mode
- Multiplexed mode
- Local/Remote Test modes
- Transparent (Protocol-Free) Mode
- 56 kb/s Mode

Non-multiplexed Mode

In non-multiplexed mode, the DLC functions as a conventional serial communications controller capable of supporting full-duplex data transfers at rates up to 2.048 Mb/s. This mode is useful in non-ISDN applications such as Local Area Networks, personal computer networks, Host-to-Host or terminal-to-Host applications.

Multiplexed Mode

In multiplexed mode, the DLC's SBP interfaces cleanly with the SBP on the Am79C30A DSC or the Am79C32A IDC, and provides access to 31 64 kb/s time slots.

Local/Remote Loopback Test Modes

The DLC can be placed in either local or remote loopback mode under software control. In local loopback, the transmitter output is tied back to the receiver input. In remote loopback, the receiver input from the network is transmitted back to the network for system test purposes.

Transparent Mode

In Transparent Mode, the DLC receives and transmits data without performing any HDLC protocol processing, creating a clear path between the SBP and the transmit and receive FIFOs. This mode can be used in either multiplexed or non-multiplexed modes.

56 kb/s Mode

In 56 kb/s Mode, the DLC transmits and receives data at a 56 kb/s data rate. In this mode, the DLC sends data within an 8-bit time slot period, and always pulls the SBOUT pin High during the transmit time period of the eighth bit. Similarly, during data reception, the DLC shifts in 7 bits during the programmed time slot, but always waits until eight bits have been accumulated before transferring the data to the receive FIFO.

Universal Synchronous/Asynchronous Receiver/Transmitter (USART)

The IDPC contains a USART for exchanging data between RS232 type terminals and the ISDN network in applications where there is no host processor. The USART provides a superset of 8250 UART features and supports both synchronous and asynchronous serial communications. The USART is capable of full-duplex operation at speeds up to 64 kb/s.

The USART supports the following functions:

- Programmable synchronous/asynchronous modes
- Software reset
- Line break recognition and generation
- Special character recognition
- Selectable stop bits (1-, 1.5-, or 2-stop bits)
- Modem control handshake lines ($\overline{RTS}$, $\overline{CTS}$, $\overline{DSR}$, and $\overline{DTR}$)
- Local loopback and "stick parity" test features

The USART receiver can detect up to 128 user-identified special characters. As each character is received, it is tested. If it is identified as a special character, a maskable interrupt can be generated.

The USART includes an internal Baud Rate Generator (BRG) that provides a clock for the transmitter and receiver sections (and to the external pin BDCLKOUT). The Baud Rate Generator's data rate is programmed by loading two "divisor latches" under software control.

Dual-Port Memory Controller (DPMC)

The DPMC provides access control and an inter-processor interrupt mechanism that permits two processors to share static RAM memory without the expense of dual-port RAM. These features are necessary in developing network interface applications for PCs and Integrated Voice/Data Workstations (IVDWs).

In a typical multi-processor application, a local processor (such as the 80188) exchanges data with the host processor in the PC or IVDW using shared memory. The DPMC performs RAM access arbitration between the local and host processors, allowing the static RAM to appear as a dual-port memory to each processor.

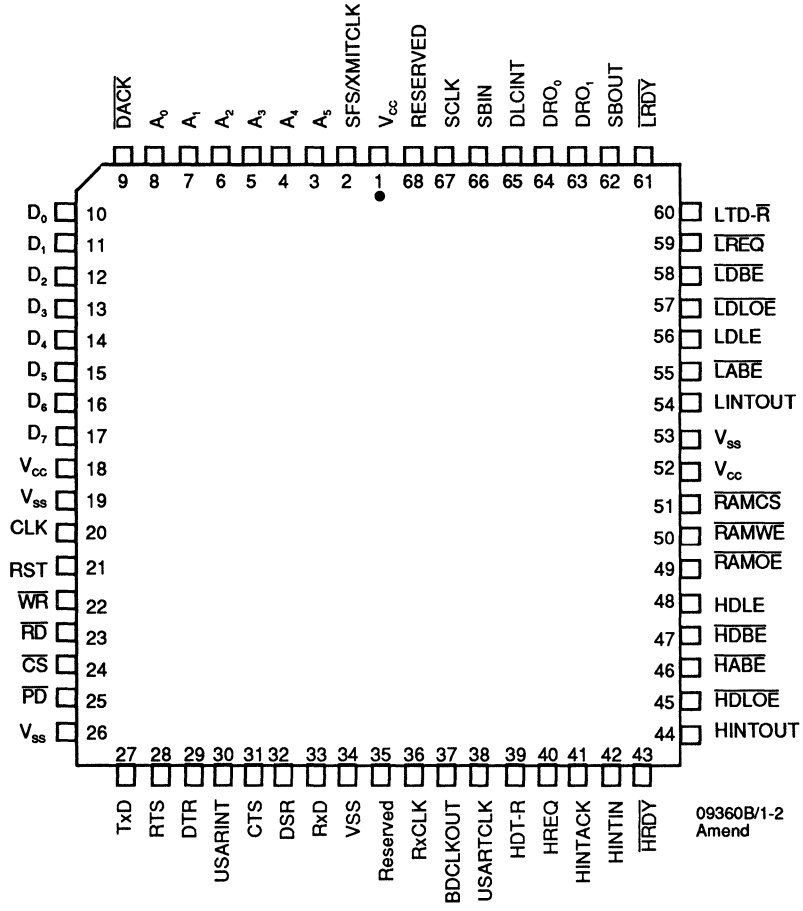
The local processor can access any device on the IDPC local bus. The host processor can only access the RAM on the IDPC local bus. Any contention between the local processor and the host processor is arbitrated by the DPMC on the IDPC. Both processors communicate via memory-resident data buffers and mailboxes. An interprocessor interrupt scheme notifies the other processor when one of the processors has written data to a buffer or a command to a mailbox.

Microprocessor Interface (MPI)

The MPI consists of an 8-bit non-multiplexed data bus that allows the IDPC to function with a 12.5-MHz 80188 processor (or other similar microprocessor) with zero wait-states.

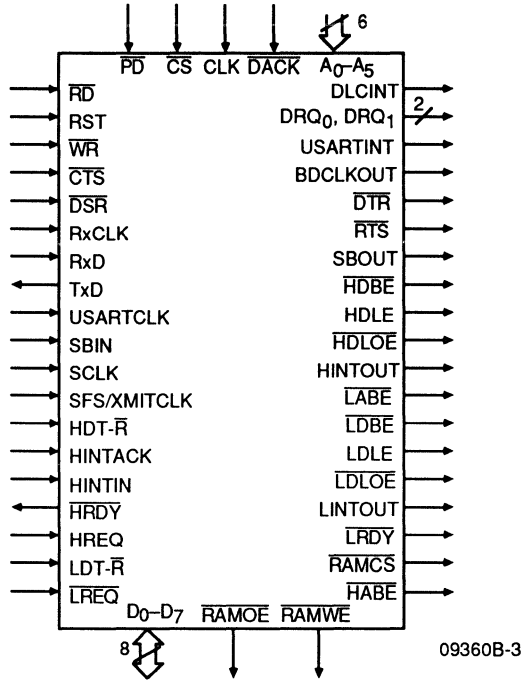
CONNECTION DIAGRAM
Top View

PLCC



Note: Pin 1 is marked for orientation.

LOGIC SYMBOL

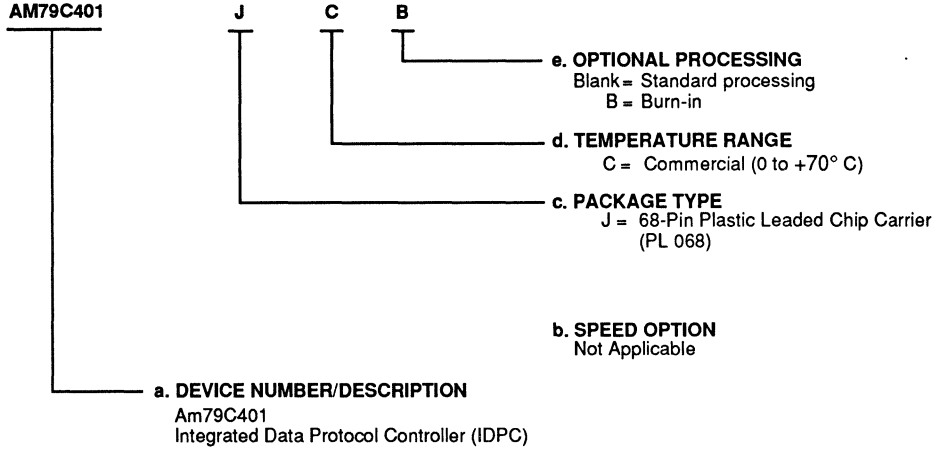


ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM79C401	JC, JCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

The interface pins of the 68-pin IDPC chip can be classified into six major groups:

Processor Bus Interface	(25 pins)
USART Interface	(9 pins)
Serial Bus Port Interface	(4 pins)
Bus Arbitration Control	(21 pins)
Power/Ground	(7 pins)
No Connects (Reserved)	(2 pins)

Unless otherwise specified, all input pins are TTL compatible.

Processor Bus Interface

A_5–A_0 Address Lines (Input)

These six address lines are generated by the external processor to select internal registers of the IDPC, and are valid only when $\overline{CS}$ is active Low.

$\overline{DACK}$ DMA Acknowledge (Input; Active Low)

The $\overline{DACK}$ signal is an indication that the DMA Controller is executing a DMA cycle to the DLC transmit FIFO. This indication occurs early in the DMA cycle, allowing the transmit FIFO to deactivate the DRQ_1 signal when the last data transfer takes place (before an unwanted DMA cycle is initiated). An equivalent signal is not required for the DLC receive FIFO operation.

CLK Master Clock (Input)

The Master Clock is an input that provides synchronization and timing for internal IDPC logic functions. CLK is normally the same clock used by the CPU.

$\overline{CS}$ Chip Select (Input; Active Low)

$\overline{CS}$ is an input that indicates the IDPC has been selected for a read or write cycle.

D7–D0 Data Lines (Input/Output; Three State)

D7–D0 are bidirectional data lines used to transfer data between the local processor and the IDPC. The direction of the data transfer is controlled by the $\overline{RD}$ and $\overline{WR}$ control lines. When $\overline{CS}$ is inactive High, the data lines remain in a high-impedance state.

DLCINT DLC Interrupt (Output; Active High)

DLCINT goes active High when the IDPC's DLC sets a status bit and the associated interrupt enable bit is set, and remains active until all pending DLC interrupts are cleared.

DRQ_0 Receive DMA Request (Output; Active High)

DRQ_0 is an output signal used by the DLC's receive FIFO to begin a DMA cycle for received data.

DRQ_0 goes active High under the following conditions:

1) when number of bytes transferred into the receive FIFO equals the number specified in the DLC FIFO Threshold Register, or
2) when an "end of packet" byte is loaded into the receive FIFO.

DRQ_0 is deactivated Low under the following conditions:

1) Reset
2) when the receive FIFO becomes empty, or
3) when the last byte of a packet is transferred from the receive FIFO to external memory.

DRQ_1 Transmit DMA Request (Output; Active High)

DRQ_1 is an output signal used by the DLC's transmit FIFO to request the start of a DMA cycle for the transmit data.

DRQ_1 goes active High when ALL of the following conditions are met:

1) the transmit byte count is not equal to zero,
2) last byte of the packet has not been loaded into the transmit FIFO, and
3) number of bytes in the FIFO is equal to or less than the value programmed in the transmit FIFO Threshold.

DRQ_1 is deactivated Low under the following conditions:

1) Reset,
2) when the transmit FIFO is full, or
3) when the last byte of the packet is loaded into the transmit FIFO.

$\overline{PD}$ Power Down (Input; Active Low)

This signal disables all internal clocks and places all three-state signals in a high-impedance state. $\overline{HRDY}$ and $\overline{LRDY}$ are disabled and all interrupt outputs are deactivated. Status and data will be lost but programming is retained. $\overline{PD}$ should be held active for a period of 8 MCLK cycles.

$\overline{RD}$ Read (Input; Active Low)

This input is used by the IDPC to indicate when data from the IDPC is being requested.

$\overline{RD}$ is qualified internally with an active Low $\overline{CS}$.

RST Reset (Input; Active High)

This input forces all functions to terminate and places the IDPC in a default state. $\overline{HRDY}$ and $\overline{LRDY}$ are disabled and all three-state outputs are placed in a high-impedance state. RST should be held active for a period of 8 MCLK cycles.

USARTINT USART Interrupt (Output; Active High)

USARTINT goes active High when the IDPC's USART sets a status bit and the associated interrupt enable bit is

active, and remains active until all pending USART interrupts are cleared.

$\overline{WR}$ Write (Input; Active Low)

$\overline{WR}$ is used by the IDPC to latch incoming data (D_7–D_0) during a write cycle.

$\overline{WR}$ is qualified internally with an active Low $\overline{CS}$.

USART Interface

BDCLKOUT Baud Rate Generator Clock Out (Output)

This signal is the output of the USART's Baud Rate Generator, and can be used as a common clocking source for a modem or other similar device.

$\overline{CTS}$ Clear To Send (Input; Active Low)

Activity on $\overline{CTS}$ generates a maskable interrupt, but does not directly control the USART's transmitter.

$\overline{DSR}$ Data Set Ready (Input; Active Low)

Activity on $\overline{DSR}$ generates a maskable interrupt, but does not directly control the USART's transmitter or receiver.

$\overline{DTR}$ Data Terminal Ready (Output; Active Low)

$\overline{DTR}$ is user-controlled and does not directly control the USART's transmitter or receiver.

$\overline{RTS}$ Request To Send (Output; Active Low)

$\overline{RTS}$ is user-controlled and does not directly control the USART's transmitter.

RxCLK Receive Clock (Input)

RxCLK is an input to the USART and is used in synchronous and asynchronous operation. In asynchronous mode, the RxCLK should be 16 times the data rate. In synchronous mode, RxCLK is synchronized to the incoming data, and the rising edge is used to latch data on the RxD pin. The maximum data rate supported is 64 kb/s.

RxD Receive Data (Input; Active High)

RxD is an input to the USART. Data on this pin is clocked into the IDPC on the rising edge of the selected clock source.

TxD Transmit Data (Output; Active High)

TxD is an output of the USART. Data is clocked out of the IDPC on the falling edge of the selected clock source.

USARTCLK USART Clock (Input)

This pin is the clock input for the USART's Baud Rate Generator. The frequency of this clock source must be an integer multiple of the desired baud rate (output of the Baud Rate Generator is the same as the data rate for synchronous operation and 16 times the data rate for asynchronous operation). If the Baud Rate Generator is programmed to divide by one, USARTCLK operates as a direct input to the USART. When the IDPC is used in conjunction with the Am79C30 (DSC), the 12.288-MHz clock output can be used as the USART clock source. The maximum data rate supported is 64 kb/s.

Serial Bus Port Interface

SBIN Serial Data In (Input)

SBIN is the serial data input to the IDPC's DLC. Data is clocked into the DLC, LSB (bit 0) first, on the rising edge of SCLK. The maximum data rate supported is $^1/_5$ of clock supplied via the CLK pin. This data rate, however, should not exceed 2.048 Mb/s.

In applications where an Am79C30A (DSC) is used, SBIN can be tied to the SBOUT pin of the DSC directly.

SBOUT Serial Data Out (Output; Open Drain)

SBOUT is the serial data output of the IDPC's DLC. Data is clocked out, LSB (bit 0) first, on the falling edge of either SCLK or SFS/XMITCLK. SBOUT data rate may range from 0 to 2.048 Mb/s. In applications where an Am79C30A (DSC) is used, SBOUT of the IPDC is tied to SBIN of the DSC with a pullup resistor.

SCLK Serial Clock In (Input)

SCLK is used as the clocking source for the DLC.

In multiplexed mode, SCLK supplies both the transmit and receive clocks synchronized to SFS/XMITCLK. In non-multiplexed mode, SCLK is used as the receive clock and is not synchronized to SFS/XMITCLK. The rising edge of SCLK is used to latch data on SBIN and the falling edge is used to shift data out on SBOUT in multiplexed mode. The maximum data rate supported is 1/5 of clock supplied via the CLK pin. This data rate, however, should not exceed 2.048 Mb/s.

SFS/XMITCLK Serial Frame Sync/Transmit Clock (Input)

This input clock signal has two different functions depending on the mode of operation selected by bits 4–0 in the DLC SBP Control Register. In multiplexed mode, this input pin functions as SFS, the synchronization pulse used to indicate the first of up to 31 independent 8-bit time slots on SBIN and SBOUT.

In non-multiplexed mode, SFS/XMITCLK is used by the DLC as the input for an independent transmit clock. SFS/XMITCLK is used by the DLC to shift data out onto SBOUT, LSB (bit 0) first, on the falling edge. This clock operates from 0 to 2.048 MHz.

Bus Arbitration Control

$\overline{HDBE}$ Host Data Bus Enable (Output; Active Low)

$\overline{HDBE}$ is an active Low output used to enable the data bus lines from the host processor to the shared RAM data bus. $\overline{HDBE}$ is driven active as a result of HDT-$\overline{R}$

being sampled High (write cycle) and remains High until the end of the memory cycle.

$\overline{\text{HABE}}$ Host Address Bus Enable (Output; Active Low)

$\overline{\text{HABE}}$ is driven active Low by the IDPC as a result of receiving an HREQ from the host processor and is used to enable the address lines from the host processor and remains active until the end of the memory cycle.

HDLE Host Data Latch Enable (Output; Active High)

This active High output is used to latch data from shared RAM to the host processor. HDLE is driven High as a result of HDT-$\overline{\text{R}}$ being sampled Low. HDLE returns Low at the end of the memory cycle.

$\overline{\text{HDLOE}}$ Host Data Latch Output Enable (Output; Active Low)

This active Low output is used by the host processor to enable the output of the data bus latches to the host processor. $\overline{\text{HDLOE}}$ is driven Low when HDT-$\overline{\text{R}}$ is sampled Low. It is deactivated when HREQ goes inactive Low.

HDT-$\overline{\text{R}}$ Host Data Transmit-Receive (Input)

HDT-$\overline{\text{R}}$ indicates whether a read or write cycle takes place to shared memory from the host processor. When HDT-$\overline{\text{R}}$ is sampled High, it indicates that a shared RAM write cycle is in progress. As a result, $\overline{\text{RAMWE}}$ and $\overline{\text{HDBE}}$ are driven active Low .

When HDT-$\overline{\text{R}}$ is sampled Low, a shared RAM read cycle occurs and $\overline{\text{RAMOE}}$ and $\overline{\text{HDLOE}}$ are driven active Low, and HDLE is driven active High .

HINTACK Host Interrupt Acknowledge (Input; Active High)

HINTACK is generated by the host processor in response to a Host Interrupt Out signal (HINTOUT) from the IDPC. HINTACK is used to clear bit 0 of the Semaphore Register, and deactivate the HINTOUT signal Low.

HINTIN Host Interrupt In (Input; Active High)

This signal is used by the host processor to generate an interrupt to the local processor via the LINTOUT pin. When the host processor pulses this pin High, bit 1 of the Semaphore Register is set to '1' and LINTOUT is driven active High.

HINTOUT Host Interrupt Out (Output; Active High)

HINTOUT is used to generate an interrupt to the host processor. This signal goes active High when the local processor sets bit 0 of the Semaphore Register to '1'. HINTOUT is deactivated Low by a pulse on the HINTACK pin or by Reset.

$\overline{\text{HRDY}}$ Host Ready (Output; Active Low; Open Drain)

$\overline{\text{HRDY}}$ is an output from the IDPC used by the host processor to complete a shared RAM memory cycle. $\overline{\text{HRDY}}$ is normally High. It is pulled Low when a request for shared RAM is received from the host processor (HREQ) and is returned High at the end of the memory cycle, or by Reset.

HREQ Host Processor Bus Request (Input; Active High)

The HREQ is a active High input to the IDPC from the host processor requesting access to the shared RAM. HREQ is sampled on the falling edge of every IDPC Master clock cycle. When sampled active, HREQ drives $\overline{\text{RAMCS}}$ and $\overline{\text{HABE}}$ active Low, and $\overline{\text{HRDY}}$ active Low. HREQ is an asynchronous input with respect to the IDPC's Master Clock and is synchronized internally.

$\overline{\text{LABE}}$ Local Address Bus Enable (Output; Active Low)

This signal is driven Low by the IDPC to enable the address lines from the local processor bus onto the shared memory bus when a Local Processor Bus Request ($\overline{\text{LREQ}}$) is received from the local processor. $\overline{\text{LABE}}$ remains active Low until the end of the memory cycle.

$\overline{\text{LDBE}}$ Local Data Bus Enable (Output; Active Low)

This signal is used to place the data from the local processor bus onto the shared RAM data bus. $\overline{\text{LDBE}}$ is driven active Low as a result of LDT-$\overline{\text{R}}$ being sampled High. $\overline{\text{LDBE}}$ remains Low until the end of a memory cycle.

LDLE Local Data Latch Enable (Output; Active High)

This signal is driven High to latch data from shared RAM onto the local processor data bus. LDLE is driven High as a result of LDT-$\overline{\text{R}}$ being sampled Low, and is deactivated Low at the end of a memory cycle.

$\overline{\text{LDLOE}}$ Local Data Latch Output Enable (Output; Active Low)

This signal is an active Low output from the IDPC that enables the output of the data bus latch onto the local processor. $\overline{\text{LDLOE}}$ is driven active Low when LDT-$\overline{\text{R}}$ is sampled Low, and is cleared when $\overline{\text{LREQ}}$ goes inactive High.

LDT-$\overline{\text{R}}$ Local Data Transmit-Receive (Input)

LDT-$\overline{\text{R}}$ indicates whether a read or write cycle takes place to shared memory from the local processor. When this signal is sampled High, a shared RAM write cycle occurs. As a result, $\overline{\text{RAMWE}}$ and $\overline{\text{LDBE}}$ are driven active Low.

When LDT-$\overline{R}$ is sampled Low, a shared RAM read cycle occurs, $\overline{RAMOE}$ and $\overline{LDLOE}$ are driven active Low, and LDLE is driven active High .

LINTOUT Local Interrupt Out (Output; Active High)

LINTOUT is driven active High when the HINTIN pin is pulsed high by the host processor, and goes Low when bit 1 in the Semaphore Register is cleared to "0" by software, or after a Reset.

$\overline{LRDY}$ Local Ready (Output; Active Low; Open Drain)

$\overline{LRDY}$ is an active Low output from the IDPC used by the local processor to complete a shared RAM memory cycle. $\overline{LRDY}$ is normally High, and is driven Low when a request for shared RAM is received from the local processor ($\overline{LREQ}$) and the host processor is currently accessing shared RAM.

$\overline{LREQ}$ Local Processor Bus Request (Input; Active Low)

This active Low signal is an input to the IDPC from the local processor when it requests access to the shared RAM. $\overline{LREQ}$ is sampled on the falling edge of every IDPC Master Clock cycle.

$\overline{LREQ}$ must be synchronous to CLK.

$\overline{RAMCS}$ RAM Chip Select (Output; Active Low)

This signal is an active Low output from the IDPC used by the shared RAM as its chip select. $\overline{RAMCS}$ is a driven Low when either $\overline{LREQ}$ or HREQ is sampled active. $\overline{RAMCS}$ remains active until the end of a memory cycle.

$\overline{RAMOE}$ RAM Output Enable (Output; Active Low)

This signal is an active Low output signal from the IDPC used by the shared RAM to enable its output drivers. $\overline{RAMOE}$ is driven active Low when either LDT-$\overline{R}$ or HDT-$\overline{R}$ is sampled Low and is deactivated High at the end of the memory cycle.

$\overline{RAMWE}$ RAM Write Enable (Output; Active Low)

This signal is an active Low output from the IDPC used by the shared RAM as a write strobe. $\overline{RAMWE}$ is driven Low when either LDT-$\overline{R}$ or HDT-$\overline{R}$ is sampled High and remains active until the end of a memory cycle.

Power/Ground

V_{CC} +5 V Power Supply

V_{SS} Ground

FUNCTIONAL DESCRIPTION

IDPC Block Diagram Description

The IDPC contains three major functional modules which include:

- Data Link Controller (DLC)
- Universal Synchronous/Asynchronous Receiver/ Transmitter (USART)
- Dual-Port Memory Controller (DPMC)

Data Link Controller (DLC)

The DLC consists of a transmitter, receiver, and of Control/Status registers (Figure 1).

DLC Transmitter

The transmitter resides between the IDPC's parallel bus and the serial communication network. The local processor builds a data block in memory containing the address, control, and information fields of an HDLC frame. This block of data is then moved into the transmit FIFO under either DMA or programmed I/O control.

The transmitter functions include: opening flag transmission, data transparency (via zero insertion), generation and transmission of the Frame Check Sequence (FCS) characters (if enabled), and transmission of the closing flag.

The transmitter may also be programmed to perform such functions as inverting the polarity of the data stream, transmitting an abort sequence, and transmitting an all "1s" pattern (Mark Idle) or back-to-back flags (Flag Idle) between packets.

A block diagram of the DLC transmitter is shown in Figure 2.

Transmit FIFO—The transmit FIFO consists of a 16-byte FIFO buffer, transmit byte count register, transmit byte counter, and DMA request logic. Data can be loaded into the transmit FIFO under programmed I/O or Direct Memory Access (DMA) control.

Data Register—The "user-addressable" portion of the FIFO is termed the data register. The transmit FIFO sets bit 3 (XMIT BUFFER AVAIL) in the FIFO Status Register to "1" to indicate when the Data Register is available. This bit is set any time the Data Register is empty, and is cleared when the Data Register is written to and the FIFO becomes full or the last byte of a packet is loaded into the FIFO.

Parallel-to-Serial Shift Register—The output of the transmit FIFO is loaded one byte at a time into the parallel-to-serial shift register then shifted out to the zero bit insertion logic before appearing at the SBOUT pin.

Flag and abort characters are loaded into the parallel-to-serial shift register for transmission by the DLC when a flag or abort sequence is to be sent. During the transmission of a flag or abort sequence the zero bit insertion logic is disabled.

Bit Residue—The transmitter also has provisions for handling bit-residue. Bit-residue refers to the number of bits left over after the I (Information) field of a frame (excluding inserted zeros) is divided into 8-bit bytes. Protocols such as LAPD and SDLC specify that data be sent in 8-bit quantities. In HDLC, however, the information field can be any number of bits and not necessarily an integral number of 8-bit characters. On the IDPC a programmable 3-bit field is provided that specifies the number of valid data bits received or transmitted during the last byte of a frame. Transmitting packets with bit-residue I-fields requires that the user load a 3-bit register field (Residue Bit Status Control Register) with the number of residue

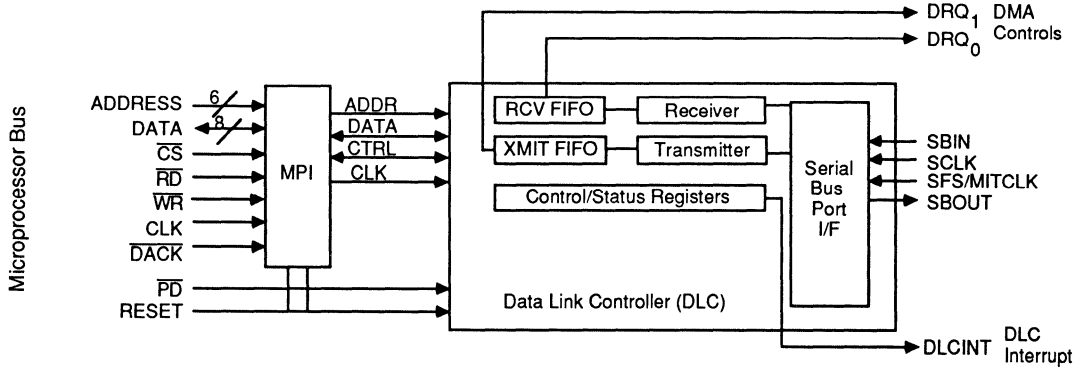


Figure 1. DLC Block Diagram

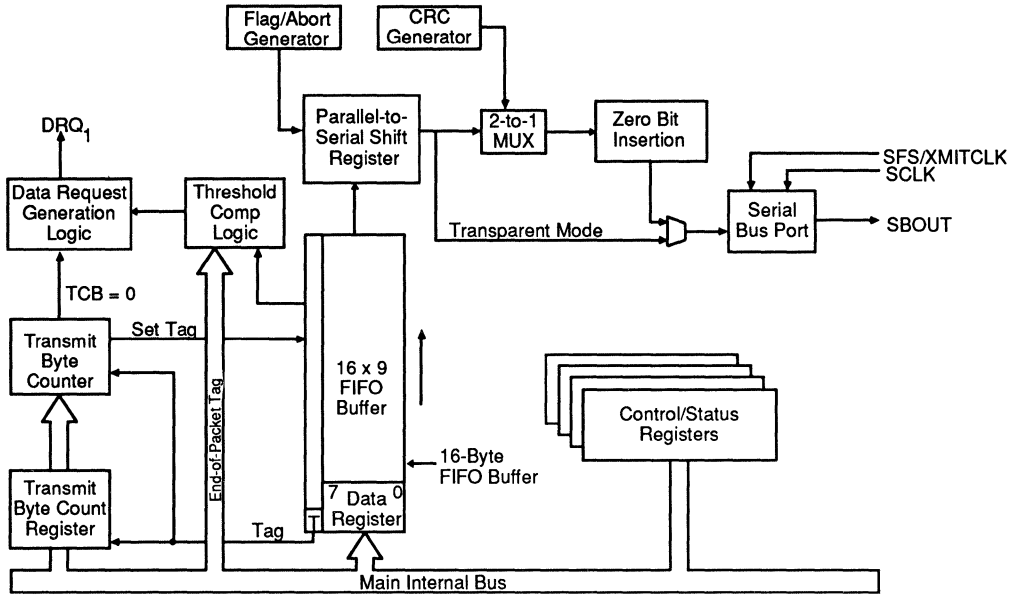


Figure 2. DLC Transmitter

09360B–5

bits in the packet (in the last byte of the I-field just prior to transmitting the FCS). When the last byte of the I-field is to be transmitted, the contents of the Residue Bit Status Control Register are used to determine when the FCS is actually sent.

CRC Generator—The Frame Check Sequence (FCS) is 16 bits long and contains the generated CRC code for the frame being transmitted. All data transmitted between the opening and closing flags (excluding inserted zeros) is included in the CRC calculation. The standard CCITT generator polynomial, $X^{16}+X^{12}+X^5+1$, is used.

Zero Bit Insertion—The zero bit insertion logic provides data transparency by ensuring that flag or abort characters embedded in the data is not recognized by the remote receiver as actual data link control characters. The zero bit insertion logic monitors the data stream between the opening and closing flags of a frame and inserts a "0" after detecting five contiguous "1"s.

Flag/Mark Idle Generation —The DLC transmitter may be programmed to either flag or mark idle when the transmitter is enabled and is not actively sending a data packet or an abort sequence.

Abort Generation —An abort sequence will be sent by the DLC transmitter whenever the Send Abort bit (bit 0 in the DLC Command/Control Register) is set. The transmitter will continue sending an abort sequence as long as this bit is set; however, if the Send Abort bit is set and cleared on two successive writes to the DLC Command/Control Register at least one abort character will be sent.

Serial Bus Port—The Serial Bus Port (SBP) sits at the output of the transmitter and performs a number of functions related to time slot assignment, clock selection, data inversion, transmitter enable/disable, and loopback testing.

Time Slot Multiplexer: The time slot multiplexer operates in two modes: multiplexed and non-multiplexed. In the multiplexed mode, one of 31 time slots may be programmed to transmit data; in the non-multiplexed mode, the data is transmitted directly without multiplexing. In addition, time slots 0 and 1 can be concatenated in multiplexed mode into one 16-bit wide slot by holding the frame sync signal, SFS/XMIT , active through the first bit time of time slot 1.

Transmit Enable/Disable Logic: The transmit enable/disable logic monitors bit 1 of the DLC Command/Control Register. When this bit is reset to "0," the SBOUT pin is set to a physical mark idle state (i.e., the SBOUT pin is pulled High). Note that if the transmitter is disabled or en-

abled during the programmed time slot (in multiplex mode), the disabling or enabling of the transmitter will occur at the end of the time slot.

Local Loopback Logic: The DLC can be placed in a local Loopback mode for test purposes by setting bit 6 in the SBP Control Register to "1." Local loopback disconnects the SBIN pin and connects the transmitter output to the receiver input. The selected transmit clock is used as the receive clock.

Remote Loopback Logic: The DLC can be placed in the remote loopback mode for diagnostic purposes by setting bit 7 of the SBP Control Register to "1." In remote loopback mode, the transmitter is disabled and data received at the SBIN pin is echoed out the SBOUT pin. The receiver operates normally in this mode.

Transparent Mode: In Transparent Mode, all HDLC protocol functions are bypassed. Data is transmitted exactly as it is loaded in the transmit FIFO. Transparent Mode is enabled when bit 7 of the DLC Residual Bit Status Control Register is set to "1." Two modes of operation are available in Transparent Mode using bit 6 of the DLC Residual Bit Status Control Register.

Transparent Mode 1—If bit 6 and bit 7 of the DLC Residual Bit Status Control Register are set to "1," the number of bytes to be sent, after the XMIT Enable bit in the DLC Command/Control Register is set to "1," is determined by the count programmed in the DLC Transmit Byte Count Register. When the number of bytes loaded into the transmit FIFO equals the programmed count further DMA requests are inhibited and, when the last bit of the last byte is transmitted, the XMIT Enable bit will be cleared to "0" and the Valid Packet Sent Status bit in the DLC Interrupt Source Register will be set to "1" to indicate that transmission has completed.

Transparent Mode 2—If bit 6 is set to "0" and bit 7 is set to "1" in the DLC Residual Bit Status Control Register, the number of bytes transmitted will be controlled by the XMIT Enable bit in the DLC Command/Control Register and the DLC Transmit Byte Count Register will be ignored. In this mode of operation the DLC will start generating DMA requests and sends data for as long as the XMIT Enable bit is set to "1." When this bit is cleared to "0" further DMA requests are inhibited and, when the last bit of the last byte is transmitted, the Valid Packet Sent Status bit in the DLC Interrupt Source Register will be set to "1" to indicate that transmission has completed.

56 kb/s Mode: If bit 7 of the DLC Residual Bit Status Control Register is set to "1" and the DLC is programmed for multiplex operation, the DLC will transmit data at a 56 kb/s data rate instead of the normal 64 kb/s data rate. In this mode, the DLC will send data within an 8-bit time slot period, but will always disable the SBOUT pin High during the eighth bit time period.

DLC Receiver

The receiver (shown in Figure 3) processes serial data packets from the SBP and transfers the data to a 32-byte receive FIFO, where it is transferred to "off-chip" RAM memory under DMA or processor control. Data rates from 0 to 2.048 Mb/s are allowed. The hardware can receive an entire packet and move the information to off-chip RAM without processor intervention if DMA is used in the design. Packet status information is then reported on a packet-by-packet basis at the time that the last byte of the packet has been moved to memory.

Dedicated hardware modules perform bit-level operations on each frame of data including mark-Idle and flag-Idle detection, data inversion, flag/abort recognition, zero bit deletion, CRC checking, and address recognition.

Functionally, the receiver consists of the following major circuits:

Serial Bus Port
Zero bit Deletion Unit
Short Frame Byte Counter
CRC Checker
Serial-to-Parallel Shift Register
Address Detection Unit
Receive FIFO
Receive Byte Counter

Serial Bus Port—The Serial Bus Port (SBP) receives serial data from the SBIN pin and sends it to the flag/abort detection unit and the zero bit deletion unit.

The SBP performs three operations on the incoming data:

1) Mark Idle Detection
2) Programmable data inversion
3) Time slot de-multiplexing

Zero Bit Deletion Unit—The zero bit deletion unit monitors the data stream between the opening and closing flags and removes a zero that appears after a string of five consecutive "1"s ("0"s are added during transmission to prevent a data pattern from resembling an abort, opening, or closing Flag).

Short Frame Byte Counter—The Short Frame Byte Counter is a 4-bit counter that counts the number of characters that have reached the serial-to-parallel shift register. If a frame ends in a flag, and the number of bytes received is less than the value programmed into the Minimum Packet Size Register, and data has been placed in the FIFO (Receive Byte Counter > 0), the SHORT FRAME ERROR status bit in the Receive Frame Status Register will be set to "1" when the last byte of the packet is read from the receive FIFO. Note that packet status is reported to the user and interrupts are generated, when the last byte of the received packet is read by

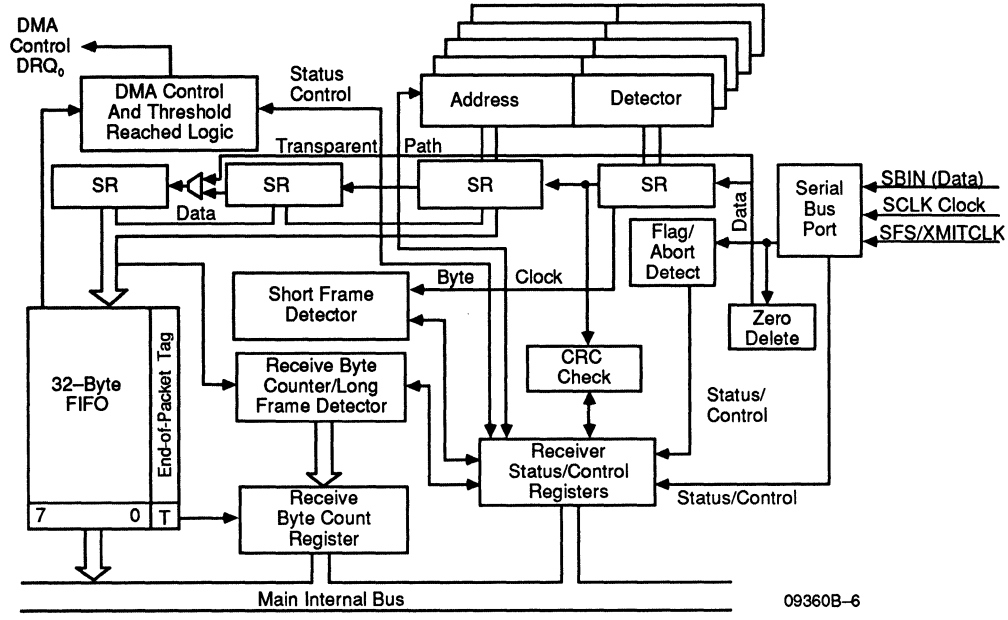


Figure 3. DLC Receiver

the processor or placed in off-chip RAM by the DMA controller.

CRC Checker—The output of the flag/abort detection unit is shifted into the CRC Checker. When the closing flag is detected, the 16-bit Frame Check Sequence has just been shifted into the checker. At this point, the contents of the CRC Checker is examined. If an error is detected, the CRC ERROR status bit is set in the Receive Frame Status Register (bit 2) when the last byte of the packet is read from the receive FIFO. The FCS can optionally be placed in the receive FIFO by setting the ENABLE FCS PASS-THRU bit (bit 7) to "1" in the DLC Command/Control Register. If this bit is set to "0" the FCS is discarded by the DLC and is not available for use.

Serial-to-Parallel Shift Register—The output of the zero bit deletion unit is fed into a 32-bit shift register which converts the serial stream into bytes.The parallel output of the shift register is fed to the receive FIFO, one byte at a time.

Address Detection Unit—The address detection unit identifies packets that are addressed to the receiver. Depending on programming, the first, second, or both of the first two bytes of each received packet (after receipt of an opening flag) are compared with the contents of five address registers (four user-programmable registers and one broadcast). If address recognition is enabled and the incoming packet's address field matches one of the address registers, the packet is received; otherwise, the packet is discarded and the receiver goes into a wait pattern looking for the next flag. As a programmable option, the Command/Response (C/R) bit (bit 1 of the first byte) can be ignored. If address recognition is disabled (i.e., bits 4–0 of the DLC Address Control Register are reset to "0"), all packets are accepted by the DLC.

Receive FIFO—The receive FIFO is a 32-byte buffer located between the serial-to-parallel shift register and the Microprocessor Interface (MPI). The FIFO is loaded by the shift register and unloaded by either the local processor or DMA controller. When the user-visible Data Register contains data, the RECV DATA AVAIL status bit in the DLC FIFO Status Register is set to "1" and a maskable interrupt is generated. This status bit is cleared to "0" when the byte is read from the Data Register and the receive FIFO becomes empty.

Receive FIFO Threshold Logic: This logic generates a receive FIFO Threshold Reached signal. This signal is High whenever the number of bytes in the receive FIFO are equal to or greater than the threshold level programmed into the FIFO Threshold Register (bits 7–4). This signal is used to set the RECV TRSHLD REACHD status bit to "1" in the DLC FIFO Status Register and generate a maskable interrupt. The Threshold Reached signal is also used to generate DRQ_0.

Data Movement Control: Data is moved from the receive FIFO either by DMA or Programmed I/O. The DRQ_0 pin is the receive FIFO's Data Request output to

the off-chip DMA controller. DRQ_0 is activated when the level in the FIFO reaches the programmed threshold, or the last byte of a packet is placed in the FIFO. Once activated, DRQ_0 remains active until the FIFO becomes empty or the last byte of a packet is removed from the FIFO. DRQ_0 will not be re-activated until the status of the previous packet has been read (by reading the least significant byte of the Receive Byte Count Register).

When programmed I/O is used, a maskable interrupt is generated when the programmed threshold is reached. The user removes data by reading the receive FIFO Data Register. If the last byte of a packet is placed in the FIFO, the end-of-packet Interrupt (maskable) indicates that the FIFO requires servicing. In this case, the user reads a data byte and then polls the RECV DATA AVAIL bit in the DLC FIFO Status Register. The user continues to remove data in this read-a-byte, poll RECV DATA AVAIL bit, read-a-byte fashion until the RECV DATA AVAIL bit is no longer set to "1." Since this procedure is in response to an end-of-packet Interrupt, the lack of a valid RECV DATA AVAIL bit indicates that the previously read data byte was the last in the packet and packet status should be read. The RECV DATA AVAIL bit de-activated by the receive is FIFO in response to the packet's last byte being removed. This happens even if the FIFO is not empty (for example, data from a new packet has been received). The RECV DATA AVAIL status bit will not be re-activated until the least significant byte of the Receive Byte Count Register is read (and additional data is in the FIFO).

Receive Byte Counter—The Receive Byte Counter keeps track of the number of current packet bytes transferred into the receive FIFO. When the last byte is removed from the FIFO, the contents of the counter are loaded into the Receive Byte Count Register.

The receive byte count is used to identify long frames and frames that have terminated prior to any data being placed in the buffer. Software uses the receive byte count to determine the length of a received frame.

Long Frame Error—A long frame error occurs when the closing flag of a frame is not detected before the number of received bytes equals the value programmed in the Maximum Packet Size Register. When this occurs, the byte that caused the long frame error is tagged as the last byte and the LONG FRAME ERROR status bit is set to "1" in the DLC Receive Frame Status Register.

Bit Residue—As data is shifted into the serial-to-parallel shift register, the bits in each byte are counted. When the counter reaches eight, a byte of data is transferred into the receive FIFO. If a closing flag is detected and this count does not equal eight the NON-INT # BYTES RECV status bit in the DLC Receive Frame Status Register will be set to "1" to indicate that a non-integer number of bytes has been received. When this occurs the 3-bit field (bits 2–0) in the DLC Residual Bit Status Control Register reports the number of data bits received at the time the closing flag was detected. This register is a delayed-reporting type register like the Receive Byte Count Register. The residue bits are right-justified before being placed in the receive FIFO.

Transparent Mode—In Transparent Mode, all receive HDLC functions (i.e., flag/abort detection, CRC checking, and zero bit deletion) are bypassed. Two modes of operation are provided through the use of bit 6 and bit 7 of the DLC Residual Bit Status Control Register. When the RECVER ENABLE bit in the DLC Command/Control Register is set to "1" data is loaded into the receive FIFO exactly as it is received. The FIFO operates normally. In addition, if bit 6 of the DLC Residual Bit Status Control Register is set to "1," the DLC Maximum Packet Receive Register is used to determine when the IDPC disables data reception.

Transparent Mode 1—If bit 6 and bit 7 of the DLC Residual Bit Status Control Register are set to "1," the number of bytes to be received, after the RECVER ENABLE bit in the DLC Command/Control Register is set to "1," is determined by the count programmed in the DLC Maximum Packet Receive Register. When the number of bytes loaded into the receive FIFO equals the programmed count further DMA requests are inhibited and, when the last byte is read from the FIFO, the VALID PACKET RECVD status bit in the DLC Interrupt Source Register is set to "1" to indicate that data reception is complete.

Transparent Mode 2—If bit 6 is reset to "0" and bit 7 is set to "1" in the DLC Residual Bit Status Control Register, the number of bytes received is controlled by the RECVER ENABLE bit in the DLC Command/Control Register. and the DLC Maximum Packet Receive Register is ignored. In this mode of operation, the DLC accumulates 8-bit characters and operates as in normal mode of operation as long as the RECVER ENABLE bit is set to "1." When this bit is reset to "0" further DMA requests are inhibited and, when the last byte is read from the FIFO, the VALID PACKET RECVD status bit in the DLC Interrupt Source Register will be set to "1" to indicate that data reception is complete.

56 kb/s Mode: If bit 7 of the DLC Residual Bit Status Control Register is set to "1" and the DLC is programmed for multiplexed operation, the DLC receives data at a 56 kb/s data rate instead of the normal 64 kb/s data rate. In this mode, the DLC receives 7 bits of data within an 8-bit time slot period, but always accumulates eight bits in the receive shift register before transferring it to the receive FIFO.

Universal Synchronous/Asynchronous Receiver/Transmitter (USART)

The IDPC's USART provides a superset of the 8250 UART features and supports both synchronous and asynchronous serial communications. The USART is capable of full-duplex operation at speeds up to 64 kb/s us-

ing either the on-chip programmable Baud Rate Generator or external clock sources.

The USART consists of the following major circuits:

- USART Receiver
- USART Transmitter
- Modem Control Unit
- Interrupt Controller
- Baud Rate Generator
- Clock Selector Unit

A simplified block diagram of the USART is shown in Figure 4.

USART Receiver

The receiver performs serial-to-parallel conversion, verifies framing, buffers the data in a FIFO, and detects break conditions, parity errors, and special characters. A maskable interrupt is generated if a parity error, framing error, or a break condition is detected. Data can be read out of the receive FIFO into external RAM memory under control of the external processor or DMA.

Parity Checking—If the ENABLE PARITY bit (bit 3) in the USART Line Control Register is set to "1," the parity generation/checking logic is enabled and parity is checked on all characters loaded into the receive FIFO. When the parity check bit generated by the receiver does not match the parity bit appended by the transmitter, the PARITY ERROR IN FIFO status bit in the USART Line Status Register is set to "1" and, if the RECV LINE STATUS bit in the USART Interrupt Enable Register is set to "1," an interrupt will be generated when that character is transfered to the receive FIFO. The CHAR w/ PARITY ERROR AVAIL status bit in the USART Status Register is then set to "1" when the character with the parity error reaches the output of the receive FIFO. No interrupt is generated by this bit. The selection of odd or even parity is made via bit 4 of the USART Line Control Register. Parity checking is available only in asynchronous mode.

Stick Parity—Stick parity is a test mode that forces the parity bit to be generated and detected as the logical inversion of the USART Line Control Register bit 4 when USART Line Control Register bits 5 and 3 are set. For example, if bits 5, 4, and 3 are set then the parity bit is always generated and detected as a "0." If bits 5 and 3 are set and bit 4 is cleared then the parity bit is always generated and detected as a "1."

Break Detection—Break Detection is performed in the asynchronous mode only. If the receive data input is held spacing ("0"s) for more than a full character time (start bit, data bits, parity bit, and stop bits), the USART sets the BREAK DETECT status bit (bit 4) in the USART Line Status Register to "1" and, if bit 2 in the USART Interrupt Enable Register, RECV LINE STATUS is set to "1" an interrupt is generated.

Framing Error Detection—Frame errors are detected in only the asynchronous mode. In this mode, if a received character does not have a valid stop bit and a Break condition is not present, the USART reports a framing error by setting bit 3 of the USART Line Status Register. If bit 2 in the USART Interrupt Enable Register, RECV LINE STATUS, is set to "1" an interrupt is generated.

Receive Shift Register—The receive shift register provides serial-to-parallel conversion for the serial data entering the Receive Data (RxD) pin.

Receive FIFO—The receive FIFO is a 4-byte, 10-bit-wide buffer used for temporary storage of receive data from the receive shift register. The FIFO provides storage for 8 data bits, one special character flag, and one parity error flag. Parity and special character conditions are checked when data is loaded into the FIFO.

The presence of a character with either a special character or parity error flag is reported in the USART Line Status Register. Maskable interrupts are generated when an error condition is detected. Data is read out of the FIFO from the receive FIFO Data Register. Only the 8 data bits are accessible by the external processor or DMA controller.

While special character and parity error interrupts are generated when the character is loaded into the FIFO, the parity error present and special character available status bits (in the USART Line Status Register) are not set until the character is at the FIFO output. This allows the user to identify which character caused the interrupt.

The receive FIFO Data Register is the equivalent of the Receive Buffer Register in the 8250 UART. The presence of valid data in the receive FIFO Data Register is indicated by bit 0 (RECV DATA AVAIL bit) in the USART Line Status Register.

Receive Overrun Error Detection—If the receive FIFO is full when a newly received character is to be loaded into the FIFO, receive overrun error is reported via bit 1 (RECV BUFFER OVERUN) in the USART Line Status Register, and, if the RECV LINE STATUS bit in the USART Interrupt Enable Register is set to "1," an interrupt is generated.

Receive Character Length—If the USART Line Control Register is programmed to receive characters with fewer than 8 bits, the unused bit positions are filled with "0"s as the character is placed in the receive FIFO.

Receive FIFO Timeout—If the FIFO level is below the programmed threshold and no new characters are received within approximately 2048 receiver clocks, and, if the RECV FIFO TIMEOUT bit in the USART Interrupt Enable Register is set to "1," an interrupt is generated to in-

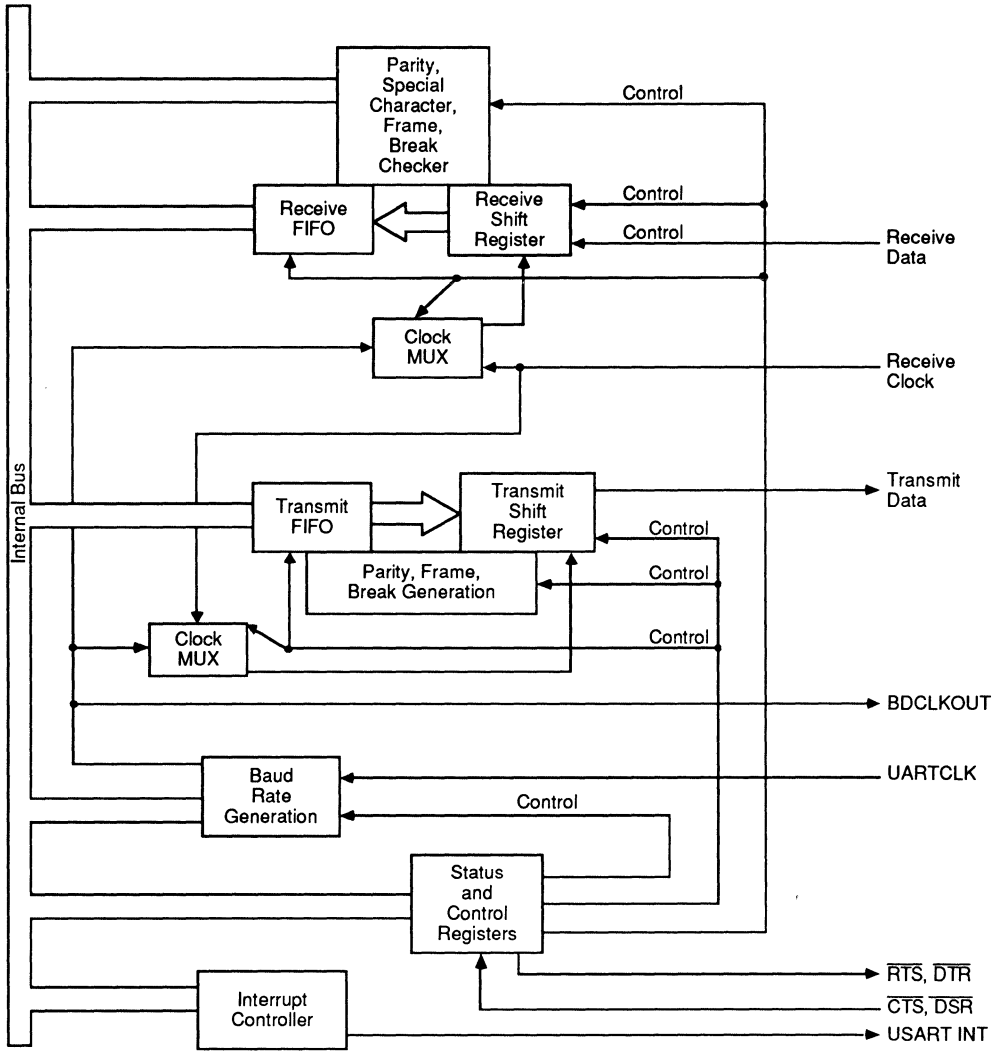


Figure 4. USART Simplified Block Diagram 09360B–7

dicate that data is available in the FIFO. This function is available only in the asynchronous mode.

Special Character Recognition Logic—Special character recognition is performed by using the lower order 7 bits of each received character as a pointer into a 128-bit RAM. If the addressed RAM bit has been programmed by the user to "1," the SPCHL CHAR IN FIFO status bit (bit 7) in the USART Line Status Register is set to "1," and, if the SPCHL CHAR bit (bit 5) in the USART Interrupt Enable Register is set to "1," an interrupt is generated. The SPCHL CHAR AVAIL status bit (bit 2) in the USART Status Register is set when the special character reaches the output of the receive FIFO.

Synchronous Mode—The synchronous mode of operation is enabled when the SYNC/ASYNC SELECT bit (bit 2) in the USART Control Register is set to "1." In this mode of operation, the receiver clocks in data bits on each rising edge of the selected clock. Each 8 bits received are placed in the receive FIFO without regard for framing or breaks. This mode allows for transparent reception of a data stream, regardless of protocol.

USART Transmitter

The USART Transmitter consists of the following major circuits:

Transmit FIFO
Parallel-to-Serial Shift Register
Frame, Break, and Parity Generation Logic

Transmit FIFO—The transmit FIFO serves as a buffer for data being moved from memory to the parallel-to-serial shift register. As the shift register becomes empty, the next character is moved from the FIFO to the parallel-to-serial shift register.

When the number of bytes in the transmit FIFO becomes less than or equal to the programmed transmit threshold level in the USART Control Register (bits 6 and 5), the XMIT TRSHLD REACHD status bit (bit 5) in the USART Line Status Register is set to "1," and , if the XMIT FIFO TRSHLD bit (bit 1) in the USART Interrupt Enable Register is set to "1," an interrupt is generated.

Bit 5 in the USART Line Status Register is the equivalent of the Transmitter Holding Register Empty in the 8250 UART.

Parallel-to-Serial Shift Register—The parallel-to-serial shift register converts parallel data from the FIFO into serial form for transmission.

The shift register clock can come from either the Baud Rate Generator (BDCLKOUT) or from the RxCLK pin. The clock source for the shift register should be 16 times the data rate in asynchronous mode and the same as the data rate in synchronous mode. Bit 6 (XMIT SHIFT REG EMPTY) of the USART Line Status Register is set to "1" when the transmit FIFO is empty and the last bit of the transmit shift register has been shifted out. An interrupt is generated by this condition if the SHFTREG EMPTY bit (bit 6) in the USART Interrupt Enable Register is set to "1."

Frame, Break, and Parity Generation Logic—Frame generation takes place only in the asynchronous mode of operation. The number of stop bits and character length are programmed via the USART Line Control Register. The same parameters apply to the receiver. Even, odd, and no parity can be selected via the USART Line Control Register.

Break Generation—The USART will generate a break condition (all "0"s) when bit 6 (BREAK) in the USART Line Control Register is set to "1." When this bit is set the USART transmits a minimum 10 consecutive "0"s immediately after completing any character transmission in progress, and continues sending this pattern until the bit is cleared. When the bit is set and the current character being transmitted clears the shift register the transmit FIFO is cleared. When the bit is reset, the TxD pin will be returned High for at least one bit time before a new character is transmitted. This allows the start bit of the new character to be generated.

Modem Control Unit

The USART contains modem handshake signals for use in controlling communications between the IDPC and a RS232 type terminal. The modem handshake signals include: $\overline{RTS}$, $\overline{CTS}$, $\overline{DSR}$, and $\overline{DTR}$.

$\overline{RTS}$ and $\overline{DTR}$ are outputs and are controlled by the local processor via bits 1 and 0 in the USART Modem Control Register, respectively. $\overline{CTS}$ and $\overline{DSR}$ are inputs and their status can be read via the USART Modem Status Register, bits 4 and 5 respectively. The Change in $\overline{DSR}$ and Change in $\overline{CTS}$ bits in the Modem Status Register indicate pin status since the USART Modem Status Register was last read. A maskable Modem Status Interrupt is generated if bit 3 of the USART Interrupt Enable Register and either bit 1 or bit 0 of the USART Modem Status Register are set. The Change in $\overline{DSR}$ and Change in $\overline{CTS}$ bits are cleared when the USART Modem Status Register is read.

Interrupt Controller

The USART interrupt controller issues an interrupt request to the external processor if an interrupt occurs and that particular interrupt is enabled in the USART Interrupt Enable Register. The interrupt request remains active until the source of the interrupt is cleared. Bits 1, 2, and 3 of the USART Interrupt Identification Register define the source of the interrupt. Bit 0, when cleared, indicates that an interrupt is pending.

Baud Rate Generator

The USART Baud Rate Generator is a programmable 16-bit divider that receives its input from the USARTCLK pin and can provide the clock to the USART transmitter and receiver. The Baud Rate Generator is configured by loading the USART Baud Rate Divisor LSB and MSB Registers. These registers are accessed by setting the DIV LATCH ACCESS BIT (bit 7 in the USART Line Control Register) and then writing to USART hexadecimal addresses 20 and 21 (USART Data Registers and Interrupt Enable Register addresses when the DIV LATCH ACCESS BIT is cleared).

In asynchronous mode, the Baud Rate Generator must be programmed to a value 16 times the data rate. The output of the Baud Rate Generator is fed to the USART transmitter and receiver and BDCLKOUT pin. Programming the Baud Rate Generator to divide-by-1 passes the USARTCLK unaffected. An internal divide-by-16 circuit generates the appropriate clock rate for the transmitter.

Clock Selector Unit

The sources of the transmitter and receiver clocks are independently selectable. For example, when bit 0 is set to "1" in the USART Control Register, the Receiver uses the output of the Baud Rate Generator for its clock. When bit 0 is cleared, the RxCLK input is used. The same options apply to the transmitter, except in this case, bit 1 in the USART Control Register specifies the clock source.

Dual-Port Memory Controller (DPMC)

The DPMC permits the use of shared memory in a multiprocessing environment. The local processor exchanges data with the host processor via shared memory and interprocessor hardware interrupts. The local processor also accesses any device on the IDPC external bus. The host processor can only access the RAM on the IDPC external bus. Any contention between the local processor and the host processor is arbitrated by the DMPC logic internal to the IDPC, providing for transparent access to shared memory. Both processors communicate via memory-resident data buffers and "mailboxes." An inter-processor interrupt scheme notifies the other processor when one of the processors has written data to a buffer or a command to a mailbox.

Functionally, the Dual-Port Memory Controller consists of the following major circuits:

Memory Cycle Arbitration and Control
Buffer/Latch Control
Inter-processor Interrupt Controller

A simplified block diagram of the Dual-Port Memory Controller is shown in Figure 5.

Memory Cycle Arbitration and Control

The DPMC generates the cycle timing for all accesses to shared RAM. The length of each cycle is fixed and independent of the cycle times of either the local or host processors. This logic generates memory cycles in response to processor requests. In case of conflicting requests, the logic arbitrates the conflict, granting the first memory access cycle to one processor while holding off the other processor via the appropriate ready line. The DPMC always arbitrates in favor of the local processor (L-port). If a request from the host port (H-port) is present during a local memory cycle, the host processor is granted the next memory cycle.

Buffer/Latch Control

The Buffer/Latch Control logic performs such functions as enabling RAM output drivers, enabling data bus latches, and generating RAM cycle timing.

Inter-Processor Interrupt Controller

The inter-processor interrupt controller provides for inter-processor interrupts via the Semaphore Register. The Semaphore Register is located in the IDPC and is used to coordinate inter-processor interrupts. The local processor can access the Semaphore Register directly, but the host processor cannot (access is provided via strobes to specific pins on the IDPC).

For a local-to-host-processor interrupt, the local processor writes a "1" in bit 0 of the Semaphore Register. When this bit is set, the Interrupt Controller activates the Host Interrupt Out (HINTOUT) pin. The host then responds by pulsing the Host Interrupt Acknowledge (HINTACK) line, clearing the bit and thus the interrupt.

A host-to-local-processor interrupt is initiated when the host pulses the Host Interrupt In (HINTIN) pin. When this happens, Bit 1 in the Semaphore Register is set, activating the Local Interrupt Out (LINTOUT) line. The local processor then acknowledges the interrupt by clearing bit 1 in the Semaphore Register.

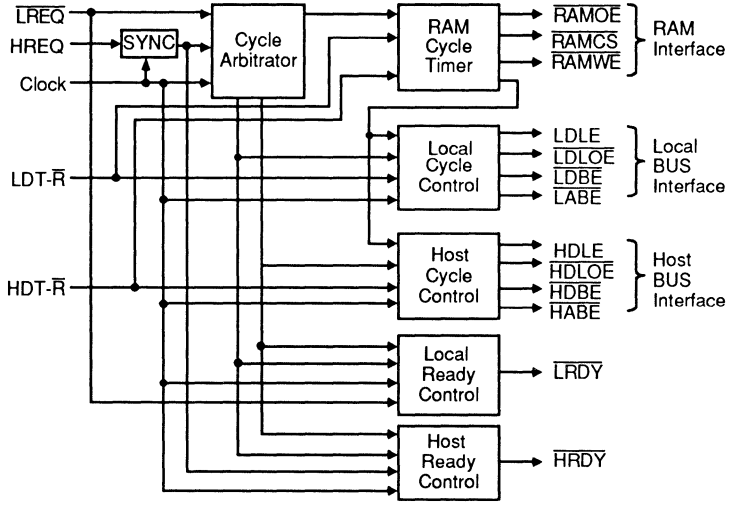
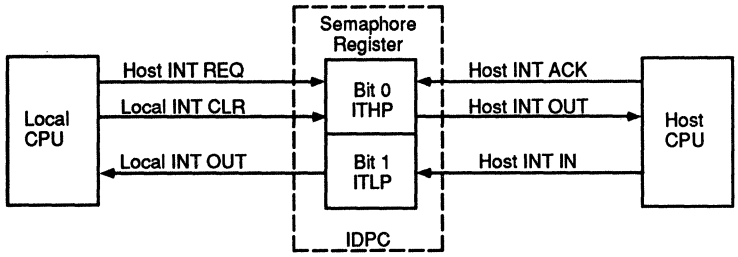


Figure 5. Dual-Port Memory Controller (DPMC) 09360B–8



Notes: Local Interrupt Clear and Host Processor Requests are writes to the Semaphore Register by the local processor.
ITLP = Interrupt to Local Processor
ITHP = Interrupt to Host Processor

09360B-9

Figure 6. Inter-Processor Interrupt Structure

Programmable Features

The IDPC is controlled via internal registers that are written and read by software running on the local processor connected to the IDPC external bus. These internal registers may be mapped into either memory or I/O space.

The internal registers occupy a 64-byte block located in the local processor's memory address space. The starting address of the memory block is determined by address decode logic (external to the IDPC) that is used to generate the IDPC chip select signal ($\overline{CS}$). The registers and their respective memory offset values are provided in Tables 1–4.

In systems containing more than one processor (e.g., a workstation application with host processor and local processor), only the local processor can access the IDPC registers. The host processor, however, can control IDPC operations indirectly by issuing requests to the local processor via shared memory supported by the IDPC's Dual Port Memory Controller.

The programmable registers are used for establishing modes of operation, configuring the IDPC, and monitoring/reporting status.

Table 1. IDPC Address Map

Offset (Hex)	Block
00–1F	DLC
20–2A	USART
2B–3E	Reserved
3F	DPMC

Data Link Controller (DLC)

DLC Transmitter

The programmable features associated with the DLC transmitter include:

Transmit Enable—the transmitter may be disconnected from the output pin (SBOUT), leaving other transmit functions intact.

Abort Generation—interrupts a frame and places the transmitter in the abort condition.

Flag/Mark Idle—either condition may be selected as an idle state between frames.

CRC Generation—may be enabled or disabled.

FIFO Threshold—user selectable threshold of 0 to 15 bytes. When the level of the transmit FIFO falls to this level or below, status is reported and a DMA request is generated.

Transparent Mode—transmit HDLC control functions (i.e., flag generation, CRC generation, abort generation, and zero bit insertion) are disabled and data is sent as received from the transmit FIFO.

Interrupts–the following transmitter-related interrupts can be selectively enabled or disabled:

- Valid Packet Sent
- FIFO Buffer Available
- Transmit Threshold Reached
- Transmit Underrun

DLC Receiver

The DLC receiver programmable features include:

Receiver Enable—when disabled, the receive data input pin (SBIN) is disconnected leaving other receiver functions intact.

CRC Check—selectively enables or disables the internal CRC compare operation.

CRC Pass-Thru—the FCS field can be placed into the receive FIFO with the data.

Address Recognition—program any combination of four unique one- or two-byte addresses and the broadcast address, performing address filtering on all incoming packets. In the 1-byte mode, either the first or second byte can be selected. In addition, the Command/Response bit (bit 1 of the first byte) can be ignored.

Minimum Packet Size—defines the minimum packet size in use. A short frame error is indicated if a packet is received containing fewer than the programmed number of bytes (0–15 bytes).

Maximum Packet Size—defines the maximum packet size in use. This prevents buffer overruns in the event of lost flags or protocol violations (65,536 bytes).

FIFO Threshold—select threshold of 2 to 32 bytes. When the level of the receive FIFO reaches this level or above, status is set (unless the last byte of a packet has already been read from the FIFO and status for that packet has not yet been read by the user) and a DMA request is generated. This forms an interlock that maintains synchronization between packet status and data.

Transparent Mode—receive HDLC related control functions (i.e., abort detection, CRC generation and checking, flag detection, and zero bit deletion) are disabled and data is received unaltered.

56 kb/s Mode—data is received at a 56 kb/s data rate instead of the usual 64 kb/s rate while in multiplex mode.

Interrupts—the following DLC receiver interrupts may be selectively enabled or disabled:

- Valid Packet Received
- Abort Received
- Non-Integer Number of Bytes Received (bit residual)
- Receive Data Available
- End-of-Packet in Receive FIFO

DLC Transmit/Receive Options

The following programmable options affect both the DLC transmitter and receiver:

Data Inversion—the output of the transmitter and the input of the receiver are inverted when this option is selected.

Channel Selection—up to 31 8-bit time slots for multiplexing transmitted serial data and demultiplexing received serial data may be chosen. In non-multiplexed mode, received serial data is continuous and the SFS/XMITCLK pin is used as a transmit clock input independent of the receive clock input.

Local Loopback—the DLC can be programmed to route transmitted data to the receiver for diagnostic purposes.

Remote Loopback—the DLC can be programmed to route received data to the transmit data output for remote testing capabilities.

Reset—a software reset can be generated to stop all functions, clear the FIFOs, and set all registers to their default values.

Delayed Status Reporting

The DLC contains several registers that report status in a delayed fashion (see Figure 7). The Receive Frame Status Register, Receive Byte Count Register, Receive Link Address bit field (bits 2–0 of the Interrupt Source Register), and the Receive Field of the Residue Status Control Register (bits 2–0 require this specific implementation in order to support the reception of contiguous (back-to-back) frames. These registers, and residue bit fields, maintain a "history" of frame status, and byte counts of up to three previously received frames while a fourth frame is actually being received. This allows status storage for up to 4 frames.

Each of these registers and bit fields are comprised of the following four stages: current, holding, master, and slave.

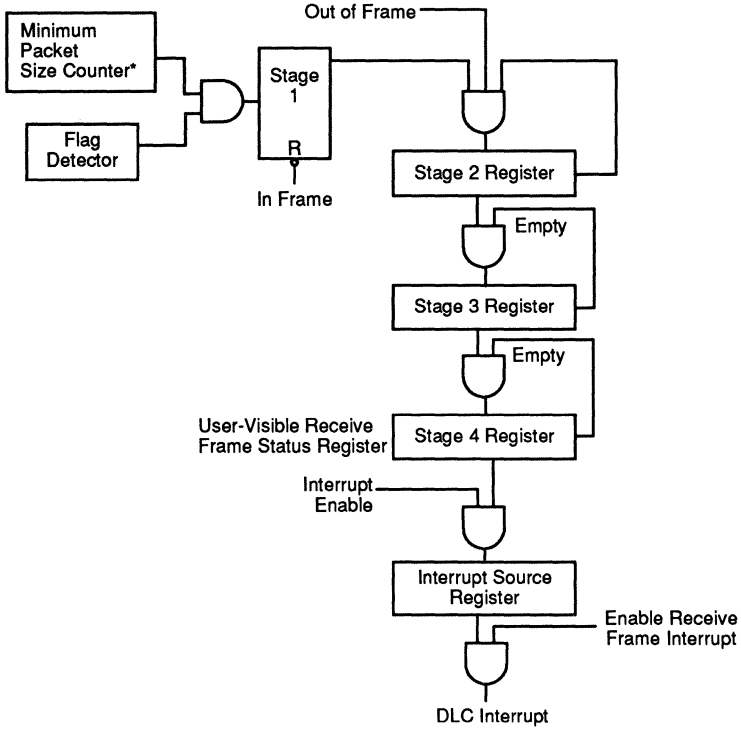
As shown in Figure 7, stage 1 holds the current status of the frame presently being received on the link. This stage is updated with the current DLC receiver status as changes occur in real-time.

Stage 1 is transferred to stage 2 when stage 2 is empty and the last byte of another packet is transferred to the DLC receive FIFO. If stage 2 is not empty, then stage 1 retains its status. When stage 1 has transferred to stage 2, stage 1 is free to acquire status of the next packet.

Stage 2 status is transferred to stage 3 as soon as stage 3 becomes empty, allowing status to move from stage 1 to stage 2.

Stage 3 status is transferred to stage 4 (which is the actual status register read by user software) when stage 4 is empty and the last byte of a packet is received. If stage 4 is not empty, then stage 3 retains its status. Stage 4 is cleared when read by software or when DLC reset occurs.

The frame status can back up. If stage 4 has not been read when an event occurs that would cause stage 3 to be transferred to stage 4, stage 3 is not transferred to stage 4. What happens is as follows: existing stages 1, 2,



*Minimum packet size is used as one possible example.

09360B–10

Figure 7. Four-Stage Delayed Status Mechanism

3, and 4 are not disturbed, the DLC receiver freezes stage 1 and data at the SBIN pin is ignored.

Additional received packets are lost until stage 4 is read. As soon as stage 4 is read, stage 3 is transferred to stage 4, stage 2 moves to stage 3, and stage 1 is transferred to stage 2. At this point, the DLC receiver logic enters a "hunt-for-flag" state, and frame reception can begin again.

The valid and invalid packet received bits of the DLC Interrupt Source Register are also reported in a delayed fashion.

Note: If the various delayed-stacked status registers have not been read since the Interrupt Source Register was last read, and the LSB of the Receive Byte Count Register is read, these status registers will be cleared. This ensures that the four-layer stack will remain in sync if a packet is received and the status registers are not read.

Detailed Description of User-Visible DLC Registers

The DLC contains 23 registers, as shown in Table 2.

Table 2. DLC Registers

Offset (Hex)	Register Name	Size (Bytes)	Type
00	Command/Control Register	1	Read/Write
01	Address Control Register	1	Read/Write
02	Link Address Recognition Register 0	2	Read/Write
04	Link Address Recognition Register 1	2	Read/Write
06	Link Address Recognition Register 2	2	Read/Write
08	Link Address Recognition Register 3	2	Read/Write
0A	Serial Bus Port Control Register	1	Read/Write
0B	Minimum Receive Packet Size Register	1	Read/Write
0C	Maximum Receive Packet Size Register	2	Read/Write
0E	Interrupt Source Interrupt Enable Register	1	Read/Write
0F	Receive Frame Interrupt Enable Register	1	Read/Write
10	Receive Link Interrupt Enable Register	1	Read/Write
11	FIFO Status Interrupt Enable Register	1	Read/Write
12	Transmit Byte Count Register	2	Read/Write
14	FIFO Threshold Register	1	Read/Write
15	Interrupt Source Register	1	Read Only
16	Receive Byte Count Register	2	Read Only
18	Receive Frame Status Register	1	Read Only
19	Receive Link Status Register	1	Read Only
1A	FIFO Status Register	1	Read Only
1B	Receive FIFO Data Register	1	Read Only
1C	Transmit FIFO Data Register	1	Write Only
1D	Residual Bit Control Status Register	1	Read/Write
1E–1F	Reserved	2	–

DLC Command/Control Register (00 HEX)

This register is used to control basic transmitter and receiver functions.

7	6	5	4	3	2	1	0
ENABLE FCS PASS-THRU	DLC RESET	ENABLE CRC GENER.	ENABLE CRC CHECK	FLAG/ MARK IDLE SELECT	RECVER ENABLE	XMIT ENABLE	SEND ABORT

Bit 7: FCS Pass-Thru Enable (Default = 0)—The Frame Check Sequence (CRC) bytes will be transferred to the receive FIFO if this bit is set to "1"; otherwise, they will be discarded by the DLC.

Bit 6: DLC Reset (Default = 0)—When this bit is set to "1," all DLC FIFOs, latches and status/control bits are forced to their default values. A delay of ten Master Clock (CLK) cycles is required before any DLC registers can be accessed, after resetting the DLC.

Bit 5: CRC Generate Enable (Default = 1)—If this bit is set to "1," the frame is terminated by appending the calculated CRC bytes and closing flag to the last byte of a packet in the transmit FIFO. If it is cleared to "0," the frame is terminated by appending the closing flag to the byte last byte of a packet.

Bit 4: CRC Check Enable (Default = 1)—If this bit is set to "1," then on reception of the closing flag, the result of the CRC check is transferred to the CRC Error bit (bit 2) in the Receive Frame Status Register; otherwise, the CRC result is ignored.

Bit 3: Flag/Mark Idle (Default = 0)—This bit determines what the transmitter sends when not in-frame. If it is set to "1," a flag pattern (01111110) is sent. If it is reset to "0," a mark pattern (11111111) is sent.

Bit 2: Receiver Enable (Default = 0)—When this bit is set to "1" data is clocked into the Serial Bus Port; otherwise, the DLC receiver is disabled and data on the SBIN pin is ignored.

Note that if this bit is cleared while the DLC receiver is in the process of receiving a frame (i.e., the receiver is in-frame), the SBIN pin will not be disabled until the closing flag of the frame being received is detected.

Bit 1: Transmitter Enable (Default = 0)—When set to "1," data from the DLC is shifted out the SBOUT pin under control of SCLK or SFS/XMITCLK; otherwise, the SBOUT pin is disabled.

Note that if this bit is cleared while in the process of sending a frame, the DLC will complete sending the frame before disabling the SBOUT pin.

Bit 0: Send Abort (Default = 0)—When set to "1," the DLC transmitter abort generator transmits abort characters (01111111, LSB on right). If this bit is set and cleared on two successive writes, the DLC will transmit at least one abort character. The transmitter will continue to send these abort patterns for as long as this bit is set. Abort characters are always sent in whole bytes.

Note that when this bit is set the DLC transmit FIFO, DLC byte counter, and the DLC Transmit Byte Count Register will be cleared.

DLC Address Control Register (01 HEX)

All bits in the DLC Address Control Register are set and cleared by software except when initialized to default values as the result of a reset.

The DLC Address Control Register can be written and read by the local processor. When all link address enable bits (bits 3–0) and the broadcast enable bit (bit 4) are cleared to "0," the DLC does not perform address detection, and passes all received frame bytes to the DLC receive FIFO. In this case, bits 7–5 are ignored.

If one or more of the link address enable bits (bits 4–0) are set, then a successful link address compare must occur before any frame bytes can be transferred to the DLC receive FIFO.

7	6	5	4	3	2	1	0
FIRST/ SECOND BYTE SELECT	ENABLE C/R BIT CMPARE	1–2 BYTE ADDR SELECT	ENABLE BRDCST ADDR DETECT	ENABLE ADDR DETECT 3	ENABLE ADDR DETECT 2	ENABLE ADDR DETECT 1	ENABLE ADDR DETECT 0

Bit 7: First/Second Byte Selection (Default = 0)—This bit is ignored unless bit 5 of this register is set to "1." When this bit is set, only the second byte is monitored by the address recognizers (first eight bits are don't cares). When this bit is cleared, only the first byte is examined.

Bit 6: C/R Address Enable (Default = 0)—At least one of the enable bits (4–0) must be set for this bit to have any effect on DLC operation.

If any of the enable bits are set, and the C/R address enable bit is cleared, then bit 1 of the first address byte of each received frame will be ignored.

If this bit is set, then bit 1 of the first received frame address byte must compare successfully along with the other address bits for address recognition to occur.

Bit 5: Address Size 1–2 (Default = 0)—At least one of the enable bits (4–0) must be set for this bit to have any affect on DLC operation.

If any of the enable bits are set and bit 5 is cleared, then the first two address bytes of each received frame will be compared.

If bit 5 is set to "1," only one byte is compared (bit 7 specifies whether the first or second byte is compared).

Bit 4: Broadcast Address Enable (Default = 1)—When set to "1," this bit enables comparison of a receive frame address with an all "1"s (broadcast address) register. The comparison is conditioned by bits 7–5 of this register. When bits 4–0 are cleared, address detection by the DLC is inhibited. If bit 4 is cleared to a zero and one or more of the enable bits (3–0) is set, then the all "1"s pattern is ignored.

Bit 3: Address Register 3 Enable (Default = 0)—Link address 3 enable.

Bit 2: Address Register 2 Enable (Default = 0)—Link address 2 enable.

Bit 1: Address Register 1 Enable (Default = 0)—Link address 1 enable.

Bit 0: Address Register 0 Enable (Default = 0)—Link address 0 enable.

Note: When set to "1," bits 3–0 enable comparison of a received frame address with the contents of the DLC Link Address Recognition Registers 0 through 3, respectively.

The comparison of a received frame address with the contents of all enabled Address Recognition Registers is conditioned by bits 7–5 of this register.

DLC Link Address Recognition Registers (02–03, 04–05, 06–07, 08–09 Hex)

These four registers are two bytes wide with the LSB having the lower address. The LSB of each pair corresponds to the second byte following the flag. The MSB corresponds to the first byte following the flag.

All of the bits in the four Link Address Recognition Registers are set and cleared by software except when initialized to "0"s by a DLC reset or IDPC reset.

Each of these four registers has a corresponding enable bit in the DLC Address Control Register (bits 3–0). If the corresponding enable bit is set, then the value in the Link Address Recognition Register is conditioned by bits 7–5 of the DLC Address Control Register. Default = Hex 0000.

DLC Serial Bus Port Control Register (0A HEX)

All bits in the Serial Bus Port Control Register are set and cleared by software, except when initialized to default values by a DLC reset or IDPC reset. This register can be written and read by the local processor.

7	6	5	4	3	2	1	0
ENABLE REMOTE LOOP BACK	ENABLE LOCAL LOOP BACK	INVERT DATA	CHAN SELECT MSB	CHAN SELECT —	CHAN SELECT —	CHAN SELECT —	CHAN SELECT LSB

Bit 7: Remote Loopback Enable (Default = 0)—This bit is set to enable loopback for diagnostic purposes. When set, the SBIN pin is connected directly to SBOUT. In this manner, receive data is presented to SBOUT as transmitted data. In this mode, the appropriate receive clock is SCLK. Receive data may be presented to the DLC receiver depending on the setting of the receive enable bit.

Bit 6: Local Loopback Enable (Default = 0)—This bit is set to enable loopback for diagnostic purposes. When set, the transmit data path (SBOUT) is connected internally to the receive data path (SBIN is disconnected). The selected transmit clock (either SCLK or SFS/XMITCLK) is used for both the transmit and receive clocks.

Bit 5: Data Invert (Default = 0)—If this bit is set to "1," the serial bit stream being sent or received is inverted at the SBOUT and SBIN pins, respectively. If it is cleared to "0" no inversion takes place.

Bits 4–0: Channel Select—These five bits select Serial Bus Port time slots for multiplexing transmitted serial bit streams/de-multiplexing received serial bit streams.

Bit 4	3	2	1	0	Channel Selection
0	0	0	0	0	Channel 0
0	0	0	0	0	Channel 1
0	0	0	1	0	Channel 2
		⋮			⋮
1	1	1	1	0	Channel 30
1	1	1	1	1	Non-Multiplex Mode

In non-multiplexed mode, a single channel is available with the receiver clocked by the SCLK pin and the transmitter clocked by the SFS/XMITCLK pin. For all settings except non-multiplexed, both the transmitter and the receiver are clocked by the SCLK pin.

If the SFS/XMITCLK pin is held active through the first bit time of time slot 1 in multiplex mode, data in time slot 0 and 1 can be transmitted, and received 16 bits at a time.

DLC Minimum Receive Packet Size Register (0B Hex)

This register specifies the Minimum Receive Packet Size.

7	6	5	4	3	2	1	0
56 kb/s MODE	0	0	0	MIN PKT SIZE MSB	MIN PKT SIZE —	MIN PKT SIZE —	MIN PKT SIZE LSB

Bit 7: 56 kb/s Mode Enable—This bit enables the DLC transmitter and receiver to transmit and receive data at a 56 kb/s data rate instead of the usual 64 kb/s rate.

Bits 6–4: Not used and must be cleared to "0."

Bits 3–0: Minimum Receive Packet Size (Default = Hex 5)—Bits 3–0 of this register are set and cleared by software except when initialized to a default value by a DLC reset or IDPC hardware reset. This register indicates the minimum packet length (exclusive of opening and closing flags) that can be received without generating a short frame error in the Receive Frame Status Register.

At the time that the short frame interrupt is generated, the Receive Byte Count Register reflects the number of bytes in the short frame.

Bit 3	2	1	0	Minimum Packet Size
0	0	0	1	1 Byte
0	0	1	0	2 Bytes
0	0	1	1	3 Bytes
	...	...		...
1	1	1	1	15 Bytes
0	0	0	0	Not Used

Note: Although reception of packets containing only 1, 2, or 3 bytes can be programmed, a minimum of 3 bytes must be received before data is moved into the FIFO and the packet is reported.

DLC Maximum Receive Packet Size Register (OC(LSB) – OD(HSB) Hex) (Default = 0000 Hex)

This register indicates the maximum packet length (exclusive of opening and closing flags) that can be received without generating a long frame error in the Receive Frame Status Register. The value programmed into the register should be equal to the desired packet size minus three.

Note that the receive byte counter is incremented on 8-bit boundaries and therefore is compared with the Maximum Receive Packet Size register each time a character is transferred from the receive shift register to the receive FIFO. The LONG FRAME ERROR status bit (bit 4) in the DLC Receive Frame Status Register will be set to "1" when the programmed value is exceeded. When this occurs, the state of the ENABLE FCS PASS-THRU ENABLE bit (bit 7) in the DLC Command/Control Register will determine the total number of characters passed to system memory. If this bit is set to "1," the total number of characters passed through to the receive FIFO and system memory will equal to the maximum count programmed plus 3. If this bit is reset to "0," the maximum characters transferred will equal the maximum count programmed plus 1.

DLC Interrupt Source Interrupt Enable Register (0E Hex)

7	6	5	4	3	2	1	0
ENABLE RECVR LINK STATUS	ENABLE FIFO STATUS	ENABLE RECV FRAME STATUS	ENABLE VALID PACKET SENT	ENABLE VALID PACKET RECVD	0	0	0

Bits 4 and 3 provide single-level interrupt enable/disable control for valid packet received and valid packet sent status conditions. For bits 7–5, the Interrupt Source Interrupt Enable Register contains the first-level enable of a two-level interrupt enable structure. Bits 7–5 enable three corresponding Interrupt Enable Registers:

Receive Frame Interrupt Enable Register
Receive Link Interrupt Enable Register
FIFO Status Interrupt Enable Register

The valid packet received and valid packet sent interrupts have a single-level interrupt enable structure (bits 3 and 4 of the Interrupt Source Interrupt Enable Register).

When an event occurs that causes a bit to be set in one of the three status registers (Receive Frame, Receive Link, and FIFO Status Registers), and both levels of status interrupt enable are set to "1," the DLC interrupt is generated and the bit corresponding to that register is set in the DLC Interrupt Source Register. Unless both levels of interrupt enable are set, no interrupt is generated.

Bit 7: Enable Receive Link Status (Default = 0)—This bit is set as the first level of enable for the Receive Link Status Enable Register. If a status bit is set in the Receive Link Status Register, and the corresponding bit is set in the Receive Link Status Interrupt Enable Register, and bit 7 of this register is set, an interrupt is generated and bit 7 of the Interrupt Source Register is set to indicate the interrupt originated in the Receive Link Status Register.

Bit 6: Enable FIFO Status Interrupt (Default = 0)—This bit is set as the first level of enable for the FIFO Status Interrupt Enable Register. If a status bit is set in the FIFO Status Register, and the corresponding bit is set in the FIFO Status Interrupt Enable Register, and bit 6 of this register is set, an interrupt is generated and bit 6 of the Interrupt Source Register is set to indicate the interrupt originated in the FIFO Status Register.

Bit 5: Enable Receive Frame Status Interrupt (Default = 0)—This bit is set as the first level of enable for the Receive Frame Interrupt Enable Register. If a status bit is set in the Receive Frame Status Register, and the corresponding bit is set in the Receive Frame Interrupt Enable Register, and bit 5 of this register is set, an interrupt is generated and bit 5 of the Interrupt Source Register is set to indicate the interrupt originated in the Receive Frame Status Register.

Bit 4: Enable Valid Packet Sent Interrupt (Default = 0)—If this bit is set and the valid packet sent bit is set in the Interrupt Source Register, a DLC interrupt is generated. If this bit is cleared, setting of the valid packet sent bit in the Interrupt Source Register does not generate an interrupt.

Bit 3: Enable Valid Packet Received Interrupt (Default = 0)—If this bit is set and the valid packet received bit is set in the Interrupt Source Register, a DLC interrupt is generated. If this bit is cleared, setting of the valid packet received bit in the Interrupt Source Register does not generate an interrupt.

Bits 2–0:—Not used and must be cleared to "0."

1

DLC Receive Frame Interrupt Enable Register (0F Hex)

The Receive Frame Interrupt Enable Register contains a bit-for-bit image of the Receive Frame Status Register. If a status bit is set in the Receive Frame Status Register corresponding to a set bit in the Receive Frame Interrupt Enable Register, and bit 5 of the first-level enable register (Interrupt Source Interrupt Enable Register) is set, a DLC interrupt is generated and bit 5 of the Interrupt Source Register is set indicating the interrupt originated in the Receive Frame Status Register.

7	6	5	4	3	2	1	0
0	0	ENABLE OVRRUN ERROR	ENABLE LONG FRAME ERROR	ENABLE SHORT FRAME ERROR	ENABLE CRC ERROR	ENABLE NON-INT # BYTES ERROR	ENABLE ABORT RECVD

Bits 7-6:—Not used and must be cleared to "0."

Bit 5: Enable Overrun Error Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 5) is set, setting this bit enables a DLC interrupt if the overrun error bit (bit 5) is set in the Receive Frame Status Register.

Bit 4: Enable Long Frame Error Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 5) is set, setting this bit enables a DLC interrupt if the long frame error bit (bit 4) is set in the Receive Frame Status Register.

Bit 3: Enable Short Frame Error Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 5) is set, setting this bit enables a DLC interrupt if the short frame error bit (bit 3) is set in the Receive Frame Status Register.

Bit 2: Enable CRC Error Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 5) is set, setting this bit enables a DLC interrupt if the CRC error bit (bit 2) is set in the Receive Frame Status Register.

Bit 1: Enable Non-Integer Number Bytes Received Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 5) is set, setting this bit enables a DLC interrupt if the non-integer number bytes received bit (bit 1) is set in the Receive Frame Status Register.

Bit 0: Enable Abort Received Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 5) is set, setting this bit enables a DLC interrupt if the abort received bit (bit 0) is set in the Receive Frame Status Register.

DLC Receive Link Interrupt Enable Register (10 HEX)

This register is used to enable/disable interrupts from the Receive Link Status Register (Default = 0).

7	6	5	4	3	2	1	0
0	0	0	0	0	ENABLE IN-FRAME ERROR	ENABLE FLAG IDLE RECVD	ENABLE MARK IDLE RECVD

Bits 7–3:—Not used and must be cleared to "0."

Bit 2: Enable Change In In-Frame Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 7) is set, setting this bit enables a DLC interrupt if the change in in-frame bit (bit 2) is set in the Receive Link Status Register.

Bit 1: Enable Change In Flag Idle Received Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 7) is set, setting this bit enables a DLC interrupt if the change in flag idle received bit (bit 1) is set in the Receive Link Status Register.

Bit 0: Enable Change In Mark Idle Received Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 7) is set, setting this bit enables a DLC interrupt if the change in mark idle received bit (bit 0) is set in the Receive Link Status Register.

DLC FIFO Status Interrupt Enable Register (11 HEX)

This register is used to enable/disable interrupts from the FIFO Status Register (Default = 0).

7	6	5	4	3	2	1	0
0	0	ENABLE EOP RECV FIFO	ENABLE XMIT UNDRUN REACHD	ENABLE XMIT BUFFER AVAIL	ENABLE XMIT TRSHLD REACHD	ENABLE RECV DATA AVAIL	ENABLE RECV TRSHLD REACHD

Bits 7-6:—Not used and must be cleared to "0."

Bit 5: Enable EOP In Receive FIFO Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 6) is set, setting this bit enables a DLC interrupt if the EOP in receive FIFO bit (bit 5) is set in the FIFO Status Register.

Bit 4: Enable Transmit Underrun Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 6) is set, setting this bit enables a DLC interrupt if the transmit underrun bit (bit 4) is set in the FIFO Status Register.

Bit 3: Enable Transmit Buffer Available Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 6) is set, setting this bit enables a DLC interrupt if the transmit buffer available bit (bit 3) is set in the FIFO Status Register.

Bit 2: Enable Transmit Threshold Reached Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 6) is set, setting this bit enables a DLC interrupt if the transmit threshold reached bit (bit 2) is set in the FIFO Status Register.

Bit 1: Enable Receive FIFO Data Available Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 6) is set, setting this bit enables a DLC interrupt if the receive FIFO data available bit (bit 1) is set in the FIFO Status Register.

Bit 0: Enable Receive Threshold Reached Interrupt (Default = 0)—If the first level of interrupt (Interrupt Source Interrupt Enable Register, bit 6) is set, setting this bit enables a DLC interrupt if the receive threshold reached bit (bit 0) is set in the FIFO Status Register.

DLC Transmit Byte Count Register (12 (LSB) – 13(MSB) Hex) (Default = 0)

This register is written by software when the number of bytes to be transmitted is different from the current value stored in the Transmit Byte Count Register (exclusive of opening and closing flags and FCS bytes).

The register contents are written to the transmit byte counter whenever software writes the least significant byte of this register pair (if the transmitter is out of frame), or when the last byte of a packet is loaded from the transmit FIFO into the parallel-to-serial shift register. If a write to this register occurs as the last byte of a packet is being loaded, the transfer to the transmit byte counter is delayed until the write is complete. The MSB of the this register must be written first because the transmit byte counter is loaded immediately after the LSB of this register is written. A transmit FIFO underrun error clears this register.

Transmit Byte Count Decode:

Bits																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Value Selected
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1 Byte
							...									...
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	65,535 kbytes
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Not Assigned

DLC FIFO Threshold Register (14 Hex)

This register is used to specify the transmit and receive FIFO threshold levels.

7	6	5	4	3	2	1	0
RECV TRSHLD MSB	RECV TRSHLD —	RECV TRSHLD —	RECV TRSHLD LSB	XMIT TRSHLD MSB	XMIT TRSHLD —	XMIT TRSHLD —	XMIT TRSHLD LSB

Bits 7–4: Receive FIFO Threshold (Default = Hex 8)—The receive FIFO threshold counts by two since the receive FIFO buffer is 32 bytes deep.

Bit 7	6	5	4	Receive Threshold
0	0	0	1	2 Bytes
0	0	1	0	4 Bytes
0	0	1	1	6 Bytes
	...	...		...
1	1	1	1	30 Bytes
0	0	0	0	32 Bytes

Bits 3–0: Transmit Threshold Value (Default = Hex 8)—The contents of this register are set and cleared under software control except when initialized by a DLC reset or IDPC reset or when an abort is issued.

Bit 3	2	1	0	Receive Threshold
0	0	0	1	1 Byte
0	0	1	0	2 Bytes
0	0	1	1	3 Bytes
	...	...		...
1	1	1	1	15 Bytes
0	0	0	0	16 Bytes

DLC Interrupt Source Register (15 HEX)

This register is used to identify the source of interrupting conditions and to report valid-packet-transmitted, valid-packet-received.

7	6	5	4	3	2	1	0
RECV LINK STATUS	FIFO STATUS	RECV FRAME STATUS	VALID PACKET SENT	VALID PACKET RECVD	RECV ADDR MSB	RECV ADDR —	RECV ADDR LSB

Bit 7: Receive Link Status (Default = 0)—This bit is set to "1" when any bit in the Receive Link Status Register is set and both of the corresponding bits in the Receive Frame Interrupt Enable Register and bit 7 (enable received link status interrupt bit) are set in the Interrupt Source Interrupt Enable Register.

It is cleared to "0" when the Receive Link Status Register is read by software, a DLC reset is executed, or an IDPC reset is received from the processor.

Bit 6: FIFO Status (Default = 0)—This bit is set to "1" when any bit in the FIFO Status Register is set and both of the corresponding bits in the Receive Frame Interrupt Enable Register and enable FIFO status interrupt (bit 6) are set to "1" in the Interrupt Source Interrupt Enable Register.

It is cleared to "0" when the FIFO Status Register is read by software, a DLC reset is executed, or an IDPC reset is received from the processor.

Bit 5: Receive Frame Status (Default = 0)—This bit is set to "1" when any bit in the Receive Frame Status Register and both of the corresponding bits in the Receive Frame Interrupt Enable Register and enable receiver frame interrupt bit (bit 5) are set in the Interrupt Source Interrupt Enable Register.

This bit is gated when stage 3 status is actually transferred to stage 4. (See description of delayed status reporting.)

Bit 5 is cleared to 0 when the Receive Frame Status Register is read by software, a DLC reset is executed, or an IDPC reset is received from the processor.

Bit 4: Valid Packet Sent (Default = 0)—This bit is set to "1" when the last bit before the closing flag has been transmitted by the DLC transmitter (transmit byte counter = 0 and no underrun and transmitter out of frame). It is cleared when the transmitter goes in-frame, this register is read, a DLC reset is executed, or an IDPC reset occurs.

Bit 3: Valid Packet Received (Default = 0)—This bit is reset to its default value when a DLC reset is executed or an IDPC reset is received. It is set to "1" when the last byte of a packet is read from the receive FIFO buffer and no receive error has been detected for that packet. It is cleared when software reads this register, or a DLC reset or IDPC reset occurs.

Bits 2–0: Receive Address Field (Default = 110 (0 = LSB))—The receive link address field is written by hardware whenever a packet is received (with or without errors). It is a delayed-stacked field.

The link address for up to four received packets can be stored at any given time. The address field for any packet is not presented to the user until the last byte of that packet is read from the FIFO.

Bits 2	1	0	Definition
0	0	0	Contents of Link Address 0 Recognized
0	0	1	Contents of Link Address 1 Recognized
0	1	0	Contents of Link Address 2 Recognized
0	1	1	Contents of Link Address 3 Recognized
1	0	0	Broadcast Link Address (All "1"s) Recognized
1	0	1	Not Used
1	1	0	Default Value—No Packet Received
1	1	1	Packet Received with no Address Recognized enabled (Bits 4–0 of DLC Address Control Register cleared to "0s")

DLC Receive Byte Count Register (16(LSB) –17(MSB)Hex) (Default = 0)

This 16-bit register indicates the number of bytes received in a packet, not including the opening and closing flags, whether the packet was received in error or not. The actual counter is incremented each time a byte is loaded into the FIFO.

This register is a "read-only" register, and is cleared to "0" when a DLC reset is executed or an IDPC reset is received from the processor.

This register presents information in a delayed fashion. When the last byte of a packet is read from the receive FIFO, the receive byte count is made available to the user. If a new packet is received before the status from the previous packet is read by the user, the status for the new packet is stacked behind the previous packet. Status for up to four packets can be stacked at any given time. When the four-deep stack is full, the DLC receiver ignores new packets until the status from at least one packet is read by the user.

There are two mechanisms that ensure synchronization between packet data and status: 1) data from one packet cannot be read from the FIFO until status from the previous packet is read; and 2) when the least-significant byte of the Receive Byte Count Register is read, all of the delayed stacked registers for that packet are cleared (Receive Byte Count Register, Receive Frame Status Register, Residual Bit Register, and the received address field of the Interrupt Source Register). For this reason, the LSB of the Receive Byte Count Register should always be read last.

Receive Byte Count Decode:

Bits 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Value Selected
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1 byte
							...									...
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	65,535 kbytes

DLC Receive Frame Status Register (18 HEX)

This is a "read-only" register. The setting of any bit in this register will result in the setting of bit 5 in the Interrupt Source Register if the corresponding bit is set in the Receive Frame Interrupt Enable Register, and the receive frame status bit is set in the Interrupt Source Interrupt Enable Register.

This register is a delayed-stacked register. Status is not reported until the last byte of the packet is read from the FIFO. At that time maskable interrupts are generated. Status for up to four packets can be stacked at any given time.

The bits of this register are cleared to "0" (default setting) when a DLC reset is executed, the IDPC reset pin is activated, or when this register or the LSB of the Receive Byte Count Register is read.

It is possible that more than one receive error may occur simultaneously on the same receive bit. However, only one bit in this register may be set to "1" at any time. The following table indicates the precedence of the various errors and exception conditions flagged by this register (listed in descending order of precedence):

If the Receive Frame Status Register is not read (not normally read for a valid packet) before the LSB of the Receive Byte Count Register, reading the Receive Byte Count Register will clear the Receive Frame Status Register to keep the register in sync (i.e., read Receive Byte Count Register LSB last).

7	6	5	4	3	2	1	0
0	0	OVRRUN ERROR	LONG FRAME ERROR	SHORT FRAME ERROR	CRC ERROR	NON-INT # BYTES RECVD	ABORT RECVD

Bits 7–6: Not used and must be cleared to "0."

Bit 5: Overrun Error (Default = 0)—This bit is set to "1" as a result of the DLC receive FIFO detecting an overrun condition (i.e., the receive FIFO contains 32 bytes when receive data needs to be moved into the FIFO from the Parallel-to-Serial Shift Register).

Bit 4: Long Frame Error (Default = 0)—This bit is set to "1" as a result of the DLC receiver detecting a long frame error.

Bit 3: Short Frame Error (Default = 0)—This bit is set to "1" as a result of the DLC receiver detecting a short frame error.

Bit 2: CRC Error (Default = 0)—This bit is set to "1" as a result of the DLC CRC checker detecting an error when CRC check is enabled in the DLC Command/Control Register.

Bit 1: Non-Integer Number Bytes Received (Default = 0)—This bit is set to "1" as a result of the DLC receiver flag detector recognizing a closing flag character with at least three bytes received when a non-integer number of bytes has been received in a non-short frame (i.e., at least one but less than eight bits were received after zero bit deletion in the byte immediately preceding the closing flag).

Bit 0: Abort Received (Default = 0)—This bit is set to "1" as a result of the DLC receiver abort detector detecting an abort character (seven "1"s while in-frame) while the DLC receiver is in-frame and at least three bytes have been received.

DLC Receive Link Status Register (19 HEX)

The Receive Link Status Register reflects the status of the data link at the receiver input. Three conditions are monitored: mark idle, flag idle, and in-frame. Bits 5–3 reflect the current status of the link and do not generate interrupts. Bits 2–0 reflect changes in the link since the register was last read; maskable interrupts are associated with these bits. At reset, bits 2–0 are cleared and bits 5–3 are cleared by the hardware that sets them.

Bits 7–6: Not used and must be cleared to "0."

Bit 5: In-Frame Received (Default = 0)—This bit is set to "1" when the receiver goes in-frame and is cleared when the receiver is not in-frame.

Bit 4: Flag Idle Received (Default = 0)—This bit is set to "1" to indicate a flag idle condition on the data link and is cleared when flag idle is not being received.

7	6	5	4	3	2	1	0
0	0	INFRAME RECVD	FLAG IDLE RECVD	MARK IDLE RECVD	CHANGE IN INFRAME	CHANGE IN FLAG IDLE	CHANGE IN MARK IDLE

Bit 3: Mark Idle Received (Default = 0)—This bit, is set to "1" to indicate a mark idle condition on the data link and is cleared when mark idle is not being received.

Bit 2: Change In In-Frame (Default = 0)—This bit, when set, indicates that the in-frame bit (bit 5) has changed (either set or cleared) since the last time the register was read. This bit is cleared by reading the register, a DLC reset, or an IDPC reset.

Bit 1: Change In Flag Idle (Default = 0)—This bit, when set, indicates that the flag idle bit (bit 4) has changed (either set or cleared) since the last time the register was read. This bit is cleared by reading the register, a DLC reset, or an IDPC reset.

Bit 0: Change in Mark Idle (Default = 0)—This bit, when set, indicates that the mark idle bit (bit 3) has changed (either set or cleared) since the last time the register was read. This bit is cleared by reading the register, a DLC reset, or an IDPC reset.

DLC FIFO Status Register (1A HEX)

Each of the FIFO Status Register bits are set and cleared by DLC hardware to indicate the real-time status of the various conditions.

The bits of this register will be set or cleared to their default values by either a DLC or IDPC reset.

Setting any bit in this register will set bit 6 of the Interrupt Source Register providing the corresponding enable bit is set in the FIFO Status Interrupt Enable Register and the enable FIFO status interrupt bit 6 is set in the Interrupt Source Interrupt Enable Register.

7	6	5	4	3	2	1	0
0	0	EOP IN RECV FIFO	XMIT UNDRUN	XMIT BUFFER AVAIL	XMIT TRSHLD REACHD	RECV DATA AVAIL	RECV TRSHLD REACHD

Bits 7–6: Not used and must be cleared to "0."

Bit 5: EOP In Receive FIFO (Default = 0)—This bit is set to "1" to indicate that the last byte of a packet has been loaded into the receive FIFO. It remains set until no EOP tags remain in the FIFO. This is the packet received indication, and is normally used in non-DMA applications to indicate that the FIFO requires servicing.

Bit 4: Transmit Underrun (Default = 0)—This bit is set to "1" during data transmission when the transmit FIFO goes empty and the transmit byte counter is not equal to zero, and is cleared when the FIFO Status Register is read.

An abort will automatically be transmitted in response to a transmit underrun condition.

Bit 3: Transmit Buffer Available (Default = 0)—This bit is set to "1" whenever the DLC FIFO Data Register is empty, and the transmit byte counter is not equal to zero (i.e., available to be written into). This bit remains active as long as the transmit FIFO is not full, and is cleared when the last byte of a packet is loaded in the FIFO. This prevents multiple packets from existing in the FIFO at the same time (non-DMA users).

Bit 2: Transmit Threshold Reached (Default = 0)—This bit is set to "1" when the number of bytes in the DLC transmit FIFO is less than or equal to the count in the transmit FIFO threshold bit field (bits 3–0 of the FIFO Threshold Register).

This bit is cleared to a "0" when the number of bytes in the transmit FIFO becomes greater than the transmit FIFO threshold bit field value. This status bit is used to condition the DLC transmit DMA data request signal.

Bit 1: Receive Data Available (Default = 0)—This bit is set to "1" whenever a byte is available in the DLC receive FIFO Data Register, and is cleared to "0" when a byte is read and the receive FIFO becomes empty. It is also cleared when the last byte of a packet is read from the FIFO. Under this condition, it is not re-enabled until the user reads the LSB of the Receive Byte Count Register. This, in conjunction with the packet received interrupt, notifies the non-DMA user when the last byte of a packet has been read.

Bit 0: Receive Threshold Reached (Default = 0)—This bit is set to "1" when the number of bytes in the DLC receive FIFO becomes equal to the value programmed in the receive FIFO threshold bit field of the DLC FIFO Threshold Register.

This bit is cleared to "0" when the number of bytes in the receive FIFO byte counter becomes less than the receive threshold value programmed in the DLC FIFO Threshold Register.

DLC FIFO Data Registers

The receive FIFO and transmit FIFO Data Registers are 8-bit registers.

The receive FIFO Data Register is read by DMA or software to remove a byte from the receive FIFO. If read by software, the user should first poll the receive FIFO data available status bit (bit 1 in the FIFO Status Register), unless data is being read in response to a threshold reached indication in which case the number of bytes to be read is known.

The transmit FIFO Data Register is written by DMA or software. If written by software, the user should first poll the transmit FIFO buffer available status bit (bit 3 in the FIFO Status Register) to ensure that a byte is available in the FIFO, (unless the data is being loaded in response to a threshold reached indication, in which case the number of bytes that can be loaded is known).

DLC Residual Bit Status/Control Register (1D Hex)

This read/write register controls the number of bits transmitted in the last byte of a packet and displays the number of valid data bits received in the last byte of a packet. It is also the register used to enable the Transparent mode.

7	6	5	4	3	2	1	0
TRANS. MODE ENABLE	TRANS. MODE SELECT	XMIT RESIDUE COUNT MSB	XMIT RESIDUE COUNT —	XMIT RESIDUE COUNT LSB	RECVD RESIDUE COUNT MSB	RECVD RESIDUE COUNT —	RECVD RESIDUE COUNT LSB

Bit 7: Transparent Mode Enable—This bit enables data to be sent or received without any HDLC protocol related format.

Bits 6: Transparent Mode Select—This bit determines whether data transmission and reception is controlled by their respective transmitter/receiver enable control signals or maximum byte counts programmed. This bit is ignored if bit 7 of this register is cleared to "0."

Bits 5–3: Transmitter Residue Count (Default = 000)—These three bits specify the number of residue bits to be transmitted in the last byte of a packet. This is a read/write field that is cleared under software control.

Bits			
5	4	3	Residue Bits
0	0	0	8 Bits
0	0	1	1 Bit
0	1	0	2 Bits
	...		...
1	0	0	7 Bits

Bits			
5	4	3	Residue Bits
0	0	0	8 Bits
0	0	1	1 Bit
0	1	0	2 Bits
	...		...
1	0	0	7 Bits

Bits 2–0: Received Residue Count (Default = 000) —These three bits form a "read-only" field displaying the number of residue bits received. This field is cleared to "0"s upon reset or by reading this register or reading the LSB of the receive byte counter. This field is a delayed-stacked field. Status for up to four packets may be stacked at any one time.

Detailed Description of User-Visible USART Registers

Universal Synchronous/Asynchronous Receiver/Transmitter (USART)

The programmable features of the USART include:

Character Length—character length is user-selectable (5-, 6-, 7-, or 8-bit characters).

Parity— even, odd, or no parity options are selectable. In addition, "stick" Parity test mode is provided.

Stop Bits—1 or 2 Stop Bits may be selected for 6-, 7-, or 8-bit characters; 1 or $1^1/_2$ Stop Bits may be selected for use with 5-bit characters.

Data Rates—USART supports data rates from 300 to 64 kb/s in both asynchronous and synchronous modes.

Handshake Lines—the USART provides $\overline{RTS}$ and $\overline{DTR}$ assertion through software control and allows for status checking of $\overline{CTS}$ and $\overline{DSR}$.

Operational Modes—the USART may be programmed for asynchronous or synchronous operation.

Baud Rate Generator—a programmable internal baud rate generator allows a selectable clock rate for asynchronous operation. Either mode allows selection of an external clocking source.

Break Generation—software break character generation.

FIFO Thresholds—each 4-byte transmit and receive FIFO has a selectable threshold value up to 4 bytes. Upon reaching the threshold value, the USART may be programmed to interrupt the external processor.

Special Character Recognition—the user may define one or more characters as special characters, and can enable when a special character is detected. Up to 128 characters can be selected as special. If 5-, 6-, or 7-bit character lengths are selected, any combination of characters may be selected as special. If 8-bit character length is used, characters with bit patterns of 0–127 may be selected as special.

Interrupts—any of the following interrupts may be selectively enabled or disabled:

- Change in $\overline{CTS}$
- Change in $\overline{DSR}$
- Parity Error
- Receive FIFO Threshold Reached
- Receive FIFO Timeout
- Transmit Shift Register Empty
- Break Detect
- Special Character Detect
- Framing Error
- Buffer Overrun

The USART contains 14 registers, as shown in Table 3.

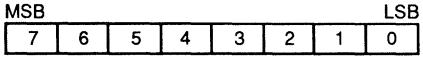
Table 3. USART Registers

Offset (Hex)	Register Name	Size (Bytes)	Type
20	Receive FIFO Data Register (DLAB = 0)*	1	Read Only
	Transmit FIFO Data Register (DLAB = 0)	1	Write Only
	Baud Rate Divisor LSB Register (DLAB = 1)	1	Read/Write
21	Interrupt Enable Register (DLAB = 0)	1	Read/Write
	Baud Rate Divisor MSB Register (DLAB = 1)	1	Read/Write
22	Interrupt Identification Register	1	Read Only
23	Line Control Register	1	Read/Write
24	Modem Control Register	1	Read/Write
25	Line Status Register	1	Read Only
26	Modem Status Register	1	Read Only
27	Control Register	1	Read/Write
28	Status Register	1	Read Only
29	Special Character Bit-Map Address Pointer Register	1	Read/Write
2A	Special Character Bit Map Command Register	1	Read/Write
2B–3E	Reserved	20	—

*Divisor Latch Access Bit (DLAB) in the Line Control Register.

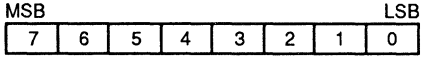
USART Receive FIFO Data Register (Default = 0) (20 Hex, DLAB =0)

The Receive FIFO Data Register is a "read-only" register. Data received by the USART is read from the receive FIFO by the CPU at this address.



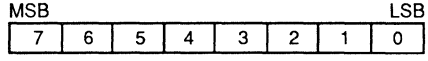
USART Transmit FIFO Data Register (Default = 0) (20 Hex, DLAB = 0)

The Transmit FIFO Data Register is a "write-only" input to the transmit FIFO. Data written in this 8-bit register is transmitted out of the TxD pin LSB first.



USART Baud Rate Divisor LSB Register (Default = 0) (20 Hex, DLAB = 1)

The Baud Rate Divisor LSB Register is an 8-bit register used to hold the LSB of the 16-bit baud rate divisor.



USART Baud Rate Divisor MSB Register (Default = 0) (21 Hex, DLAB = 1)

The Baud Rate Divisor MSB Register is an 8-bit register used to hold the MSB of the 16-bit baud rate divisor.

Note: Divide-by-one passes the USARTCLK unaffected. This allows the receiver and transmitter to operate from separate clocks in synchronous mode. A write to either the MSB or LSB Divisor Registers loads the baud rate generator with a 16-bit value.

Note: When executing a reset, the register pair is cleared to all zeros, but the baud rate generator will divides the USARTCLK by 64 until the Baud Rate Registers are programmed.

Bits																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Divide By:
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1 Byte
							...									...
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	65,535 kbytes

USART INTERRUPT ENABLE REGISTER (21 HEX, DLAB = 0)

The Interrupt Enable Register is an 8-bit read/write register used to enable specific interrupt sources (Default = 0). Setting a bit enables its corresponding interrupt. Clearing a bit disables the interrupt and resets the interrupt pin if the corresponding condition is present.

7	6	5	4	3	2	1	0
0	XMIT STATUS: SHFTREG EMPTY	USART STATUS: SPCL CHAR.	USART STATUS: Rx FIFO TIMEOUT	MODEM STATUS: CTS DSR	RECV LINE STATUS	XMIT FIFO TRSHLD	RECV FIFO TRSHLD

USART INTERRUPT IDENTIFICATION REGISTER (22 HEX)

The Interrupt Identification Register is an 8-bit read-only register that identifies which status register contains an interrupt condition. Unused bit positions (bits 7–4) return "0"s when this register is read.

7	6	5	4	3	2	1	0
0	0	0	0	INTR SOURCE MSB	INTR SOURCE —	INTR SOURCE LSB	INTR PEND

Bits 7–4: Not Used and must be cleared to "0."

Bits 3–1: Interrupt Source (Default = 000)—This 3-bit field identifies the highest priority source of all existing interrupts.

Bit 0: Interrupt Pending (Default = 1)—This bit is cleared to "0" if any interrupt is pending.

Interrupt Source Decode

Bits 3	2	1	Priority	Source	Reset By*
0	0	0	4th	CTS or DSR	Reading the Modem Status Register
0	0	1	3rd	Transmit FIFO Threshold Reached	Reading this Register and Interrupt Source = 001
0	1	0	2nd	Receive FIFO Threshold Reached	Reading this Register and Interrupt Source = 010
0	1	1	1st**	Overrun, Parity, Special Character Received, Framing, or Break	Reading Line Status Register
1	0	0	5th	Receive FIFO Timeout	Reading USART Status Register
1	0	1	6th	Transmit Shift Register Empty	Reading this Register and Interrupt Source = 101

* All bits are reset by a USART reset or an IDPC reset.

** Simultaneous receipt of a special character or a character with a parity error, and a threshold reached condition generates the interrupt for the special character or parity error before the threshold reached interrupt.

USART LINE CONTROL REGISTER (23 HEX)

This register controls access to the Baud Rate Generator Divisor registers and sets the mode of operation for the USART.

7	6	5	4	3	2	1	0
DIV LATCH ACCESS BIT	BREAK	STICK PARITY	EVEN/ ODD PARITY SELECT	PARITY ENABLE	STOP BIT LENGTH	CHAR LENGTH MSB	CHAR LENGTH LSB

Bit 7: Divisor Latch Access Bit—This bit is used to enable access to the Baud Rate Divisor Registers. If it is set to "1," access to these registers is enabled, but must be reset to "0" before accessing the receive/ transmit FIFO Data Registers and the Interrupt Enable Register.

Bit 6: Break—This bit is set to request that a break condition be transmitted. The USART will transmit the break pattern immediately after completing any character transmission in progress when this bit is set. The transmit shift register and transmit FIFO contents are discarded. The line returns to normal operation when the bit is cleared. Breaks are transmitted only in asynchronous mode.

Bit 5: Stick Parity—If parity is enabled (bit 3 set) and this bit is set to "1," parity is expected to be received opposite to that indicated by bit 4. Parity is transmitted with a value opposite that of bit 4.

Bit 4: Even/Odd Parity Select—This bit selects the parity sense used by the transmitter and receiver. If it is set to "1"; even parity is selected. If it is reset to "0"; odd parity is selected.

Bit 3: Parity Enable—When this bit is set to "1," parity generation and checking is enabled. When this bit is cleared, parity generation and checking is disabled.

Bit 2: Stop Bit Length—This bit selects the number of stop bits used in serial data transfers.

0 = 1 Stop Bit
1 = 1.5 Stop Bits (5-bit characters) or 2 Stop Bits (6-, 7-, or 8-bit characters)

Bit 1–0: Character Length—Bits 1 and 0 define the character length.

Bits 1	0	Character Length
0	0	5 Bits
0	1	6 Bits
1	0	7 Bits
1	1	8 Bits

USART Modem Control Register (24 HEX)

This register specifies modem control parameters (Default = 0).

7	6	5	4	3	2	1	0
0	0	0	LOCAL LOOPBK ENABLE	RESRVD	RESRVD	$\overline{\text{RTS}}$ PIN CONTR	$\overline{\text{DTR}}$ PIN CONTR

Bits 7–5: Not used and must be cleared to "0."

Bit 4: **Local Loopback Enable**—Setting this bit to "1" places the USART in a local loop back condition for diagnostic purposes.

Bits 3–2: Reserved

Bit 1: **$\overline{\text{RTS}}$ Pin Control**—When this bit is set to "1," $\overline{\text{RTS}}$ pin goes active-Low. This bit does not directly control the transmitter.

Bit 0: **$\overline{\text{DTR}}$ Pin Control**—When this bit is set to "1," the $\overline{\text{DTR}}$ pin goes active-Low. This bit does not directly control the transmitter or receiver.

USART Line Status Register (25 HEX)

The USART Line Status Register contains flag bits that are set to indicate the presence of a condition that can generate an interrupt if the appropriate interrupt enable bits are set in the Interrupt Enable Register. Bits 1 through 4 and 7 are cleared by reading this register. Bit 5 is cleared when the condition goes away, but the interrupt is cleared by reading the Interrupt Identification Register (when the Interrupt Identification Register is reporting this interrupt). Bits 0 and 6 are cleared when the associated conditions are no longer present.

7	6	5	4	3	2	1	0
SPCHL CHAR IN FIFO	XMIT SHIFT REG EMPTY	XMIT TRSHLD REACHD	BREAK DETECT	FRAMING ERROR	PARITY ERROR IN FIFO	RECV BUFFER OVRRUN	RECV DATA AVAIL

Bit 7: **Special Character In FIFO (Default = 0)**—This bit is set to "1" when a special character is transferred into the receive FIFO, and cleared when the USART Line Status Register is read.

Bit 6: **Transmit Shift Register Empty (Default = 1)**—This bit is set to "1" when the transmit shift register is empty (i.e. the last character transmitted) and cleared when the transmit shift register and FIFO are not empty.

Bit 5: **Transmit FIFO Threshold Reached (Default = 1)**—This bit is cleared when the number of bytes in the transmit FIFO rises above the programmed threshold, and is set to "1" when the FIFO level is equal to or below the threshold.

Bit 4: **Break Detected (Default = 0)**—This bit is set to "1" when a break condition is detected by the receiver, and is cleared when the USART Line Status register is read.

Bit 3: **Framing Error (Default = 0)**—This bit is set to "1" when an invalid stop bit is detected. A character with a framing error is not loaded into the FIFO, and is cleared when the USART Line Status register is read.

Bit 2: **Parity Error in FIFO (Default = 0)**—This bit is set to "1" when a character with a parity error is transferred into the receive FIFO from the receive shift register, and is cleared when the USART Line Status register is read.

Bit 1: **Receive Buffer Overrun (Default = 0)**—This bit is set to "1" when an overrun error results in lost receive data, and is cleared when the USART Line Status register is read.

Bit 0: **Receive Data Available (Default = 0)**—This bit is set to "1" when receive data is available in the receive FIFO Data Register.

USART Modem Status Register (26 HEX)

The 8-bit Modem Status Register is used to indicate the condition of the link handshake input signals and any change in their status. Bits 1 and 0 are cleared on reset; bits 5 and 4 reflect the input status.

7	6	5	4	3	2	1	0
RESRVD	RESRVD	$\overline{\text{DSR}}$ PIN STATUS	$\overline{\text{CTS}}$ PIN STATUS	RESRVD	RESRVD	$\overline{\text{DSR}}$ PIN CHANGE STATUS	$\overline{\text{CTS}}$ PIN CHANGE STATUS

Bits 7–6: Reserved

Bit 5: **$\overline{\text{DSR}}$ Pin Status**—This bit is set to "1" if the $\overline{\text{DSR}}$ input pin is active-Low and cleared to a "0" if $\overline{\text{DSR}}$ is inactive.

Bit 4: **$\overline{\text{CTS}}$ Pin Status**—This bit is set to "1" if the $\overline{\text{CTS}}$ input pin is active-Low and is cleared to a "0" if $\overline{\text{CTS}}$ is inactive.

Bits 3-2: Reserved

Bit 1: **$\overline{\text{DSR}}$ Pin Change Status (Default = 0)**—This bit is set to "1" when a change in the $\overline{\text{DSR}}$ input pin has occurred since this register was last read.

Bit 0: **$\overline{\text{CTS}}$ Pin Change Status (Default = 0)**—This bit is set to "1" if the input pin has changed state since this register was last read.

USART Control Register (27 HEX)

This 8-bit USART register is used to control all non-8250-UART functions. Additionally, this register contains the USART software reset bit.

7	6	5	4	3	2	1	0
RESET	XMIT FIFO TRSHLD MSB	XMIT FIFO TRSHLD LSB	RECV FIFO TRSHLD MSB	RECV FIFO TRSHLD LSB	SYNC/ ASYNC MODE SELECT	XMIT CLK SOURCE	RECV CLK SOURCE

Bit 7: Reset (Default = 0)—This bit is set to initiate a USART reset operation (identical to a reset initiated by hardware via the RST pin, except only the USART is affected). The software reset takes 2 master clock cycles, and this bit clears itself.

Bits 6–5: Transmit FIFO Threshold (Default = 00)—This field is used to hold a 2-bit count that reflects the transmit FIFO threshold. When the number of bytes remaining in the transmit FIFO is less than or equal to this level, Transmit FIFO Threshold Reached status is generated.

Bit 6	Bit 5	Transmit Threshold
0	0	0 Bytes
0	1	1 Bytes
1	0	2 Bytes
1	1	3 Bytes

Bits 4–3: Receive FIFO Threshold (Default = 11)—These two bits are used to select the receive FIFO threshold. When the number of bytes in the receive FIFO is greater than or equal to this value, the Receive FIFO Threshold Reached status is generated.

Bit 4	Bit 3	Receive Threshold
0	1	1 Byte
1	0	2 Bytes
1	1	3 Bytes
0	0	4 Bytes

Bit 2: Sync/Async Mode Select (Default = 0)—This bit determines the operating mode of the USART. If it is set to "1," the USART operates in synchronous mode where no start/stop bits are recognized and where the expected data rate is equal to the clock source programmed. If this bit is reset to "0," the USART operates in asynchronous mode and the clock source to the transmitter and receiver must be 16 times the expected data rate.

Bit 1: Transmit Clock Source (Default = 0)—The output of the internal baud rate generator is used as the clock source for the transmitter if it is set to "1"; otherwise, the RxCLK pin is used.

Bit 0: Receive Clock Source (Default = 0)—The output of the internal baud rate generator is used as the clock source for the receiver if this bit is set to "1"; otherwise, the RxCLK pin is used.

USART Status Register (28 HEX)

The USART Status Register reports status conditions that do not occur in an 8250 UART. This register also contains the Parity Error Character Available status bit. The default = 00010000.

Bits 4–1 are cleared when the corresponding condition no longer exists.

7	6	5	4	3	2	1	0
RECVR ENABLE	0	0	XMIT BUFFER AVAIL	RECV FIFO TRSHLD REACHD	SPCHL CHAR AVAIL	PARITY ERROR CHAR AVAIL	RECV FIFO TIMEOUT

Bit 7: Receiver Enable/Disable (Default = 0)—This bit is set to enable the USART receiver, and is cleared to disable the receiver.

Bits 6–5: Not used and must be reset to "0."

Bit 4: Transmit Buffer Available (Default = 1)—This bit is set to "1" whenever the FIFO Data Register is empty, and is cleared when the FIFO is full.

Bit 3: Receive FIFO Threshold (Default = 0)—This bit is set to "1" when the number of bytes in the receive FIFO is greater than or equal to the programmed receive FIFO threshold, and cleared when the number of bytes in the receive FIFO falls below the threshold value.

Bit 2: Special Character Available (Default = 0)—This bit is set to "1" when the special character reaches the output of the receive FIFO. It is cleared when the character is read from the FIFO.

Bit 1: Parity Error Character Available (Default = 0)—This bit is set to "1" when a character with the parity error reaches the output of the receive FIFO. It is cleared when the character is read from the FIFO.

Bit 0: Receive FIFO Timeout (Default = 0)—This bit is set to "1" when a receive FIFO timeout occurs. It is cleared when this register is read or when the receive FIFO becomes empty. The timeout occurs when the level in the receive FIFO is below the threshold and no characters are received in at least 2048 receiver clocks.

USART Special Character Bit-map Address Pointer Register (29 HEX)

This register is used to address the 128-bit special character bit map (Default = 0).

7	6	5	4	3	2	1	0
0	SPCHL CHAR MSB	SPCHL CHAR —	SPCHL CHAR —	SPCHL CHAR —	SPCHL CHAR —	SPCHL CHAR —	SPCHL CHAR LSB

Bit 7: Not used and must be reset to "0."

Bits 6–0: These bits represent the special character address in the bit map. A character is designated as a special character by first writing the address (which is the character itself) into bits 7–0 of the Special Character Bit-Map Address Pointer Register, and then by writing a "1" into bit 0 of the Special Character Bit-Map Command Register. A special character can be returned to normal status by writing a "0" into bit 0 of the Special Character Bit-Map Command Register (after the pointer is set).

USART Special Character Bit-map Command Register (2A HEX)

This register is used for setting and clearing the corresponding bit in the bit-map of the special character pointed to by the Special Character Bit-Map Pointer Register (Default = 0).

All bits in the bit-map are cleared on reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SET/ CLEAR BIT MAP

Bits 7–1: Not Used and must be reset to "0."

Bit 0: **Set/Clear Bit Map**—Writing a "0" or "1" to this bit clears or sets the Special character pointed to by the Special Character Pointer register. When this register is read bit 0 reflects the status of the special character pointed to by the Special Character Bit-MAP Pointer.

Note: When the receiver enable bit is set (bit 7 of USART Status Register), reading the Special Character Bit-Map Command Register returns all "1"s regardless of the actual state of the special character addressed. This is done to prevent simultaneous bit map accesses by the MPI and the internal logic.

A special character can be read or written to via the MPI only when the receiver enable bit (USART Status Register bit 7) is cleared.

Dual-Port Memory Controller (DPMC)

For multiprocessor applications, a common message area in RAM (i.e., a mailbox) is used for inter-processor communications.

When bit 0 of the Semaphore Register is set to "1," an interrupt (HINTOUT) is generated to the host processor indicating it has a message waiting in the mailbox. Bit 0 of the Semaphore Register is reset to "0" when the host processor acknowledges this interrupt by pulsing the HINTACK line.

When the host processor has completed placing a message in the mailbox for the local processor, it alerts the local processor by pulsing the HINTIN line. This results in setting bit 1 in the Semaphore Register which activates the interrupt line (LINTOUT) to the local processor. LINTOUT is de-activated when the local processor clears bit 1 of the Semaphore Register.

Table 4. DPMC REGISTERS

Offset (Hex)	Register Name	Size (Bytes)	Type
3F	Semaphore Register	1	Read/Write

Table 5. DPMC REGISTERS

Offset (Hex)	Register Name	Size (Bytes)	Type
3F	Semaphore Register	1	Read/Write

DPMC Semaphore Register

The Semaphore Register controls interrupt requests between the host processor and the local processor in a multiprocessor application. These interrupts coordinate processor-to-processor communication via shared memory. This register is cleared to "0"s by a hardware reset.

DPMC Semaphore Register (3F HEX)

7	6	5	4	3	2	1	0
0	0	0	0	0	0	INTR TO LOCAL PROC	INTR TO HOST PROC

Bits 7–2: Not used and must be reset to "0."

Bit 1: Interrupt to Local Processor (Default = 0)— This bit is set to "1" when the HINTIN pin from the host processor goes active (i.e., is pulsed). Setting this bit to "1" causes the LINTOUT pin to go active High. This bit is cleared by the local processor by writing a "0" to it. LINTOUT goes inactive when this bit is cleared.

Bit 0: Interrupt to Host Processor (Default = 0)— This bit is set to "1" by the local processor to initiate communications with the host processor. Setting this bit causes the HINTOUT pin to go active High. The bit is cleared by the HINTACK pin (from the host) going High. This bit can be read by the local processor.

APPLICATIONS

Overview

Most ISDN applications of the IDPC may be grouped into two categories:

Terminal Adapter (TA)
PC or Integrated Voice/Data Workstation (IVDW)

The major difference is the number of processors in the system. In the terminal adaptor application, a single local processor (such as the 80188) controls all system functions and the USART serves as the RS-232 interface between the ISDN network and the terminal. The IDPC provides an external bus for attachment of an external processor such as the 80188, memory, "S" Interface transceiver hardware, and other ISDN support hardware.

In the PC or IVDW application, two processors (local and host) are present. The local processor (with associated circuitry and software) and the IDPC form the heart of the communication processor. The local processor on the interface card exchanges ISDN transmit/receive data with the host processor using shared memory and interprocessor interrupts (instead of using the USART as in the TA application).

Terminal Adapter (TA) Application

A typical terminal adapter application is shown in Figure 8. For the TA application, the processor attaches to the IDPC bus and runs specialized communication software to allow "dumb terminals" to attach to an ISDN network and make ISDN data calls. The low-level programming interface required for ISDN communications is available from AMD. The arbitration software required for multiple processor communications is not used in the TA application.

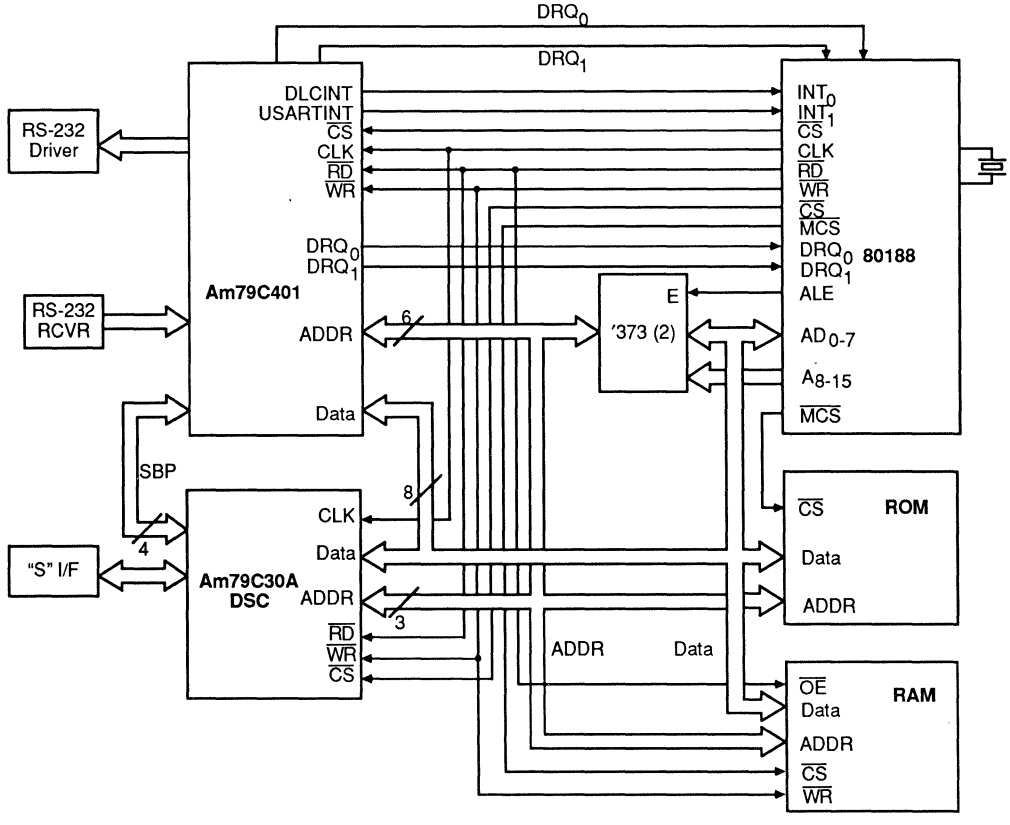


Figure 8. Typical TA Application (see 09360B–11)

09360B-11

PC or Integrated Voice/Data Workstation (PC/IVDW) Application

Typically, the PC/IVDW application uses an interface card that installs in the PC or workstation. In this application, two processors (local and host) are present. The local processor (with associated circuitry and software) performs layers 2 and 3 of the communications protocol.

The local processor on the interface card exchanges ISDN transmit/receive data with the host processor using shared memory and inter-processor interrupts (instead of using the USART as in the TA application). The Dual-Port Memory Controller on the IDPC provides for effective memory sharing and handling of inter-processor interrupts.

The local processor can access any device on the IDPC external bus, whereas the host processor can only access the RAM on the IDPC external bus. Any contention between the local processor and the host processor is arbitrated by the Dual-Port Memory Controller on the IDPC. This arrangement provides for transparent access to shared memory by both the local and host processors, allowing inter-processor communications via memory-resident data buffers and mailboxes.

A combination of IDPC interrupt-related pins and IDPC register bits (which can be read and written only by the local processor) implement the interrupt mechanism. Once the local processor or host processor has written data to a buffer or a command to a mailbox, it uses the inter-processor interrupt mechanism to notify the other processor.

A sample PC/IVDW interface design is shown in Figure 9. For the PC application, the local processor attaches to the IDPC bus and runs specialized communication software (offered by AMD) to allow PCs and integrated voice/data workstations to attach to an ISDN network and make ISDN data calls.

The local processor runs a ROM-based layer 2 and 3 ISDN communication package (offered by AMD) to control ISDN operations. The local processor and the host processor in the PC or workstation communicate via IDPC interrupts and the shared RAM on the ISDN interface card.

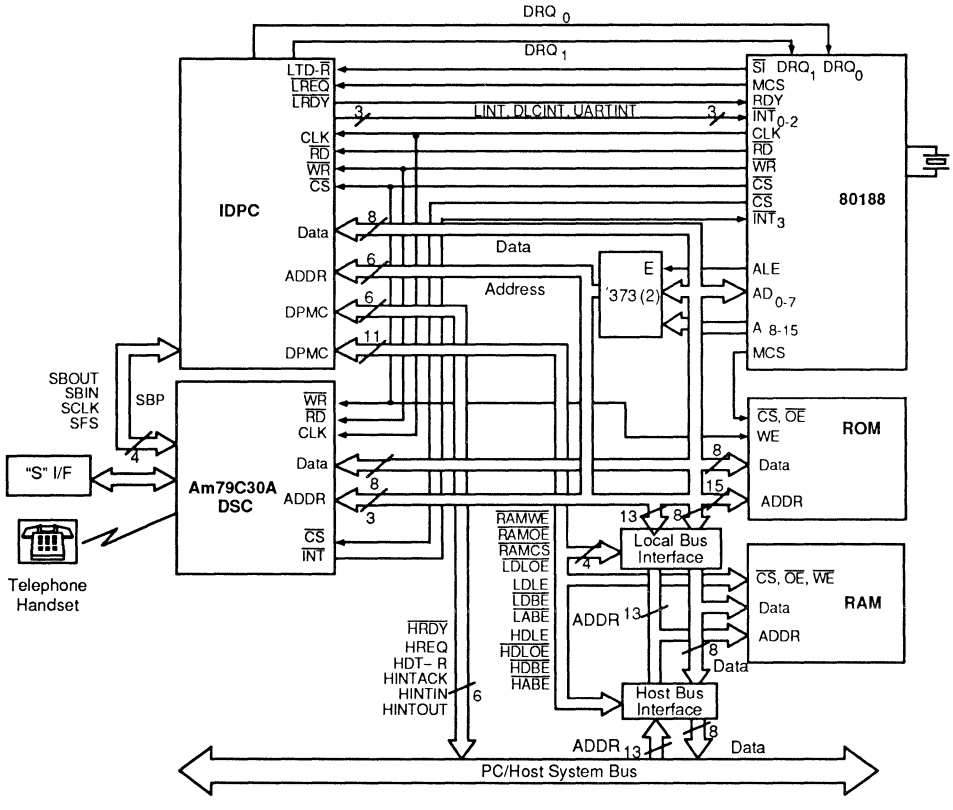


Figure 9. Typical PC Application 09360B-12

The RAM is shared (transparent to the software) using the bus arbitration logic in the IDPC's Dual-Port Memory Controller.

PC-Based SNA Terminal Emulation

The following example shows the use of the IDPC in non-multiplexed mode to provide a PC-based intelligent SNA terminal emulator. The SNA processor executes all of the SNA emulation code, with the possible exception of the SNA presentation services layer. The IDPC supports SDLC at speeds up to 2.048 Mb/s.

Similar to the IVDW-PC application, the IDPC's Dual-Port-Memory Controller provides a shared RAM area to allow the local and host processors to exchange information (see Figure 10).

Communications Protocol Overview

Most communication networks used for data transfer employ a set of rules and techniques called bit-oriented protocols. These protocols allow for the transfer of data in packets. The most common bit-oriented protocols include: HDLC, SDLC, LAPB, LAPD, and DMI.

Bit-Synchronous Message Concepts

All communications over the ISDN (or any network using the bit-oriented protocols, such as X.25 or SNA) make use of message-framing formats in which specific bit patterns (Flags) rather than control characters are used to delineate message blocks.

Frame Format

Each packet, plus its opening and closing flags, is called a frame. Each frame conforms to the following format:

Flag	Address (1–N Bytes)	Control (1 or 2 Bytes)	Information (Optional)	Frame Check Sequence (FCS)	Flag

- Flags: Used to bracket the packet (leading "0" bit followed by six "1" bits and a trailing "0" bit).
- Address: Identifies the data sender or receiver. All addresses are an integer number of bytes in length. In general, an address can be 1, 2, or "n" bytes long.

The length of an "n"-byte-long address is determined by the value of the least significant bit in each byte of the address. This bit, called the Extended Address (EA) Bit, identifies the last byte of the address. All of the bytes on

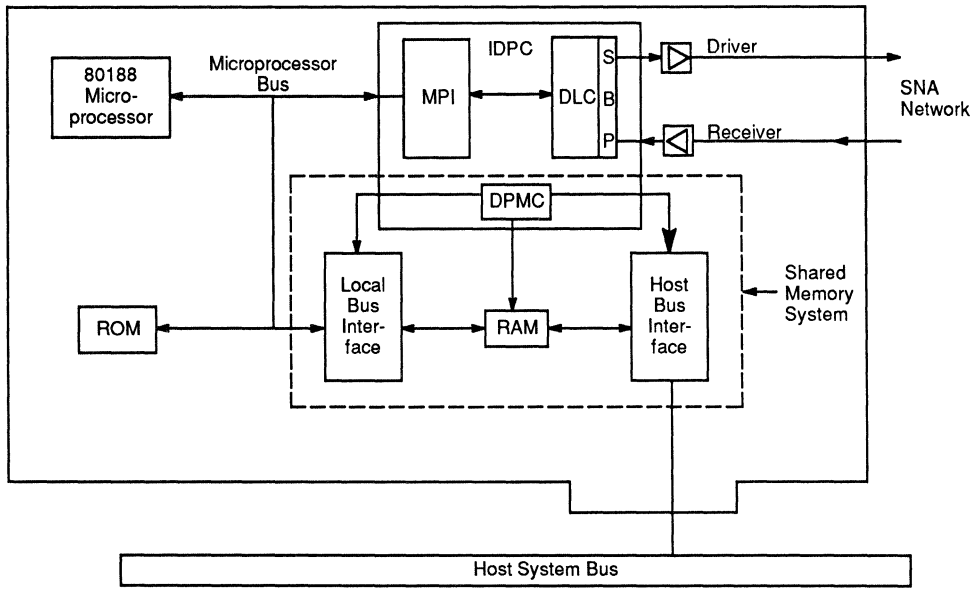


Figure 10. Typical SNA Application

an "n"-byte-long address will have the EA bit cleared to a 0 except for the last byte which has the EA bit set.

Note: The length of the address field affects the detection of a short frame (see description of short frame).

In some protocols, the second bit (bit 1) of the first byte of the address is used to indicate whether the frame is a command or a response. This bit, the Command/Response (C/R) bit, does not affect address recognition.

- Control Information: Used by higher levels of the protocol for data management (no action taken by the DLC in response to information in this field).
- Information Field (data): Variable length—up to 64K bytes long (minus address and control lengths). This field is optional and can be omitted for protocol control packets.
- Frame Check Sequence (FCS): The Frame Check Sequence is a 16-bit word that is produced by a Cyclic Redundancy Check (CRC) generator at the transmit side, reproduced by a similar circuit at the receive side, and checked against the transmitted code to determine if a bit has been dropped or picked up in error. The CRC is calculated using all bits after the Opening Flag up to the first bit of the Frame Check Sequence (excluding bits inserted for transparency).

Packet—A packet is defined as a frame minus the opening and closing flags.

Mark Idle—When frames are not being transmitted over the link, the link is idle. When the link is idle, the transmitter can be programmed to send an 'all 1s' pattern which is referred to as the mark idle condition (15 or more "1"s constitute mark idle).

Flag Idle—Prior to and between frames, the transmitter can be programmed to send back-to-back flags over the data link. This condition is referred to as flag idle.

In-Frame—The DLC receiver is said to be In-Frame when it is enabled and the first non-Flag, non-abort character is received after receiving at least one flag character.

The DLC transmitter is said to be In-Frame from the time it starts to send the first bit of the opening flag until the last bit of the closing flag has been transmitted (assuming the transmitter has not been commanded to send an abort character).

Abort Condition—The abort condition is an action that takes place in a response to the detection of an abort character while the DLC receiver is In-Frame. An abort causes the termination and discarding of the packet being received. Aborts are asynchronous events in that they can be transmitted and detected on bit boundaries.

Abort Character—The abort character is any pattern of at least seven contiguous "1"s. The DLC transmitter sends a pattern of seven "1"s and a "0" as an abort character.

Zero Insertion/Deletion (Data Transparency)—In order to prevent the data characters from appearing as flags, aborts, or mark idles , a technique called bit stuffing is employed. The contents of each packet is examined bit-by-bit (beginning with the first bit after the opening flag to the last bit of the Frame Check Sequence), and a "0" is inserted in the bit stream for any pattern containing five contiguous "1"s.

At the receiver side, "0"s are removed following the receipt of five contiguous "1"s.

Short Frame—Bit-Oriented Protocols (i.e., SDLC, HDLC) specify minimum lengths for valid packets (usually 4, 5, or 6 bytes). Any frame that is received with fewer than the minimum acceptable number of bytes is labeled as a short frame and is reported as an error.

Long Frame—In order to prevent buffer overrun, it is common practice to specify the maximum packet length for a data transfer (typically varies with each data call). Any received frame that exceeds the specified maximum frame length is terminated, identified as a long frame, and reported as an error.

Non-Integer Number of Bits Received—If a closing flag is detected and a non-integer number of bytes has been received (character preceding the closing flag contained fewer than 8 bits), a non-integer number of bytes condition is reported. This is not an error when bit-residue is allowed.

Order of Bit Transmission—All bytes are transmitted in ascending numerical order. Within a byte, the least significant bit (LSB) is transmitted first (except for the Frame Check Sequence which is transmitted MSB first).

ABSOLUTE MAXIMUM RATINGS

Storage Temperature . . . . . –60 to +150° C
Ambient Temperature Under Bias . . . . –55 to +125° C
Voltage from Any Pin to Vss . . . –0.25 to Vcc +0.25 V
Voltage from Vcc to Vss . . . . . –0.25 to +7 V
Lead Temperature (Soldering, 10 sec) . . . . . +300° C

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) 0 to +70° C
Supply Voltage (V_{CC}) +4.75 to +5.25 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise noted

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{IH}	Input High Logic Level		2.0	V_{CC}	V
V_{IL}	Input Low Logic Level		–0.2	0.8	V
V_{OL}	Output Low Logic Level	I_{OL} = 2 mA		0.4	V
V_{OH}	Output High Logic Level	I_{OH} = –400 µA I_{OH} = –10 µA	2.4	 V_{CC} – 10	V V
I_{OL}	Output Leakage Current In	0 ≤ V_{OUT} ≤ V_{CC}		±10	µA
I_{IL}	Input Leakage Current	0 ≤ V_{IN} ≤ V_{CC}		±10	µA
I_{CC}(S)	V_{CC} Supply Current (Standby)	V_{CC} + 5.25 V		800	µA
I_{CC}(A)	V_{CC} Supply Current (Active)	T_A = 70° C		30	mA

Capacitance*

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
C_I	Logic Input Capacitance			10	pF
C_O	Logic Output Capacitance			15	pF

*Parameters are not "tested."

SWITCHING CHARACTERISTICS over operating range unless otherwise noted

No.	Parameter Symbol	Parameter Description	Min.	Max.	Unit
CLOCKS					
1	t_{CH}	CLK High Time	25		ns
2	t_{CL}	CLK Low Time	25		ns
3	t_{CLCL}	CLK Cycle Time	80	1000	ns
4	t_{CHCL}	CLK High-to-Low Transition		15	ns
5	t_{CLCH}	CLK Low-to-High Transition		15	ns
6	t_{SCH}	SCLK or SFS/XMITCLK High Time	50		ns
7	t_{SCL}	SCLK or SFS/XMITCLK Low Time	50		ns
8	t_{SCLCL}	SCLK or SFS/XMITCLK Cycle Time	488		ns
9	t_{SCHCL}	SCLK or SFS/XMITCLK High-to-Low Transition		1000	ns
10	t_{SCLCH}	SCLK or SFS/XMITCLK Low-to-High Transition		1000	ns
RESET					
11	t_{RES}	RST Active	10		Master CLK Cycles
12	t_{RESDL}	Delay after RST Low before cycle	8		Master CLK Cycles
13	t_{PD}	$\overline{PD}$ Active	10		Master CLK Cycles
14	t_{PDDL}	Delay after $\overline{PD}$ High before cycle	8		Master CLK Cycles
DMA REQUEST					
15	t_{DRQ0}	Last DMA Cycle to DRQ_0 Inactive		40	ns
16	t_{DRQ1}	Last DMA Cycle to DRQ_1 Inactive		40	ns
HOST MESSAGE AVAILABLE					
17	$t_{HIAVHIDL}$	HINTACK Active to HINTOUT Inactive		30	ns
USART ASYNCHRONOUS OPERATION					
18	t_T	Clock Low to Data Valid		50	ns

SWITCHING CHARACTERISTICS

No.	Parameter Symbol	Parameter Description	Min.	Max.	Unit
USART SYNCHRONOUS OPERATION					
19	t_{URDVL}	Data Setup to Clock High	15		ns
20	t_{URDH}	Data Hold from Clock High	15		ns
21	t_{UTCDV}	Clock Low to Data Valid		50	ns
SERIAL BUS PORT					
22	t_{SDC}	Data Setup to CLK High	15		ns
23	t_{SDHLD}	Data Hold from CLK High	15		ns
24	t_{CSDV}	Data Valid from CLK Low		50	ns
LOCAL CPU WRITE					
25	t_{AVAL}	Address Valid to $\overline{CS}$ Low	15		ns
26	t_{WA}	$\overline{CS}$ Valid to $\overline{WR}$ Active	15		ns
27	t_W	$\overline{WR}$ Active Width	130		ns
28	t_{DHLD}	Data Hold from $\overline{WR}$ High	15		ns
29	t_{WD}	$\overline{WR}$ Low to Data Valid		1	Master CLK Cycle
30	t_{WR}	$\overline{WR}$ Low to $\overline{RD}$ Low	4		Master CLK Cycle
30a	t_{DCE}	$\overline{WR}$ High to $\overline{RTS}$/$\overline{CTS}$ Transition		3	Master CLK Cycle
LOCAL CPU READ					
31	t_{AVAL}	Address Valid to $\overline{CS}$ Low	15		ns
32	t_{RA}	$\overline{CS}$ Valid to $\overline{RD}$ Active	15		ns
33	t_R	$\overline{RD}$ Active Width	120		ns
34	t_{RICSI}	$\overline{RD}$ High to $\overline{CS}$ High	0		ns
35	t_{RDHLD}	Data Hold from $\overline{RD}$ High	10		ns
36	t_{RDDLY}	$\overline{RD}$ Low to Valid Data	0	80	ns
37	t_{RDHDZ}	$\overline{RD}$ High to Data Bus Hi-Z		35	ns
37a	t_{INT}	$\overline{RD}$ High to INT High		80	ns

SWITCHING CHARACTERISTICS

No.	Parameter Symbol	Parameter Description	Min.	Max.	Unit
DUAL-PORT MEMORY CONTROLLER					
38	t_{LRQS}	$\overline{LREQ}$ Low Setup to CLK Low	10		ns
39	t_{TRS}	LDT-$\overline{R}$, HDT-$\overline{R}$ Setup to CLK High	10		ns
40	t_{TRH}	LDT-$\overline{R}$, HDT-$\overline{R}$ Hold from CLK High	15		ns
41	t_{WEADLY}	$\overline{RAMWE}$ Active Delay from CLK High		35	ns
42	t_{WEIDY}	$\overline{RAMWE}$ Inactive Delay from CLK Low		35	ns
43	t_{CSADLY}	$\overline{RAMCS}$ Active Delay from CLK Low		40	ns
44	t_{CSIDLY}	$\overline{RAMCS}$ Inactive Delay from CLK Low		40	ns
45	t_{DEADLY}	$\overline{RAMOE}$ Active Delay from CLK High		40	ns
46	t_{DEIDLY}	$\overline{RAMOE}$ Inactive Delay from CLK Low		40	ns
47	t_{BADLY}	$\overline{LDBE}$, $\overline{HDBE}$ Active Delay from CLK High		40	ns
48	t_{BIDLY}	$\overline{LDBE}$, $\overline{HDBE}$ Inactive Delay from CLK Low		40	ns
49	$t_{LDLEADLY}$	LDLE, $\overline{HDLE}$ Active Delay from CLK High		40	ns
50	$t_{LDLEIDLY}$	LDLE, $\overline{HDLE}$ Inactive Delay from CLK Low		40	ns
51	$t_{DLOADLY}$	$\overline{LDLOE}$, $\overline{HDLOE}$ Active Delay from CLK High		40	ns
52	$t_{LDLOIDLY}$	$\overline{LDLOE}$ Inactive Delay from LREQ/ High		20	ns
53	t_{ABADLY}	$\overline{LABE}$, $\overline{HABE}$ Active Delay from CLK High*		40	ns
54	t_{ABIDLY}	$\overline{LABE}$, $\overline{HABE}$ Inactive Delay from CLK Low		40	ns
55	t_{LRADLY}	$\overline{LRDY}$ Active Delay from $\overline{LREQ}$ Low**		35	ns
56	t_{LRIDLY}	$\overline{LRDY}$ Inactive Delay from CLK Low**		35	ns
57	t_{HRADLY}	$\overline{HRDY}$ Active Delay from HREQ High**		35	ns
58	t_{HRIDLY}	$\overline{HRDY}$ Inactive Delay from CLK Low		35	ns
59	$t_{HDLDIDLY}$	$\overline{HDLOE}$ Inactive Delay from HREQ Low		20	ns
60	t_{HREQSU}	HREQ Setup	10		ns

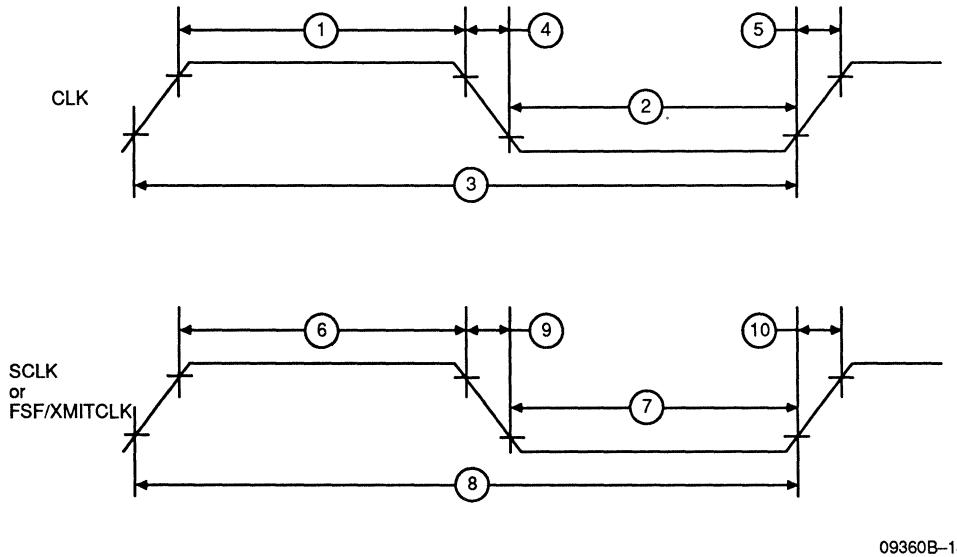
* CLK or $\overline{LABE}$/$\overline{HABE}$, whichever happens last
** With 1k pull-up resistor

SWITCHING WAVEFORMS

KEY TO SWITCHING WAVEFORMS

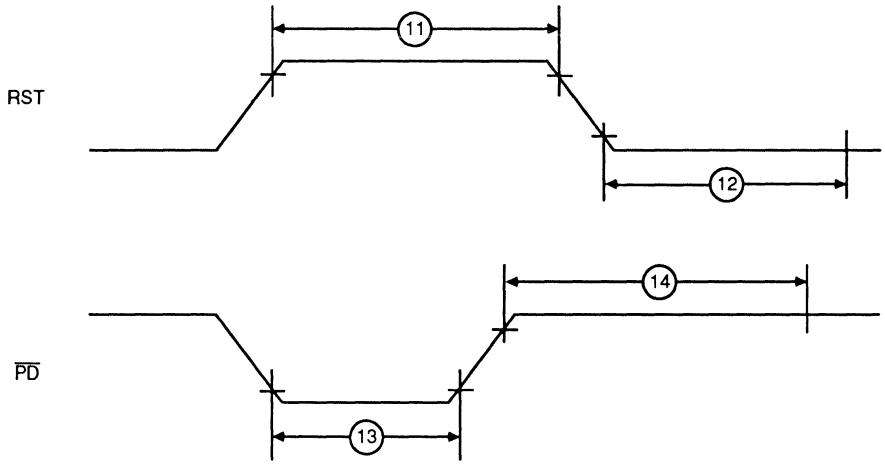
Waveform	Inputs	Outputs
	Must Be Steady	Will Be Steady
	May Change From H to L	Will Be Changing From H to L
	May Change From L to H	Will Be Changing From L to H
	Don't Care, Any Change Permitted	Changing, State Unknown

KS000010



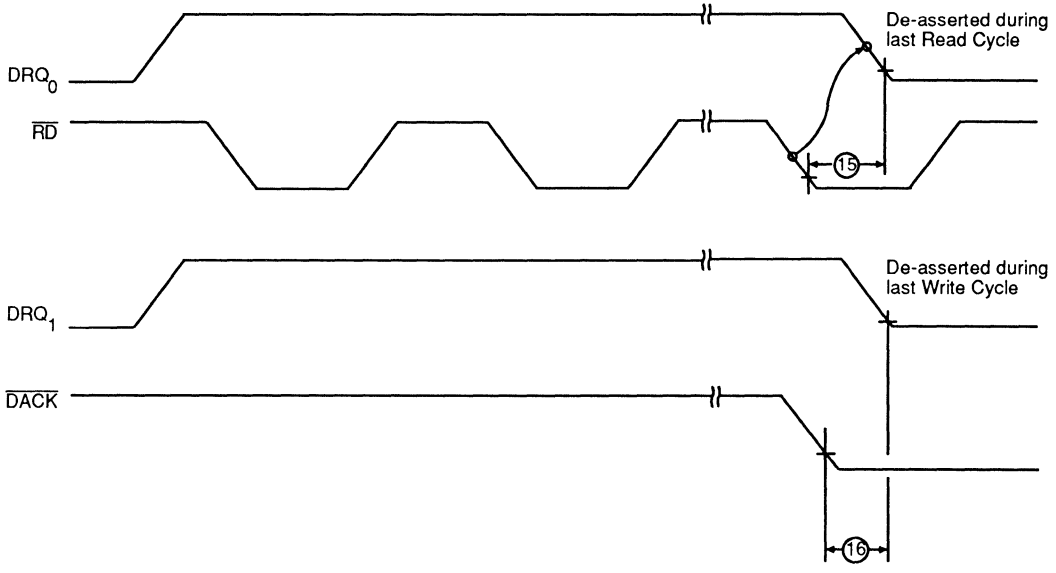
09360B–14

Clock Timing



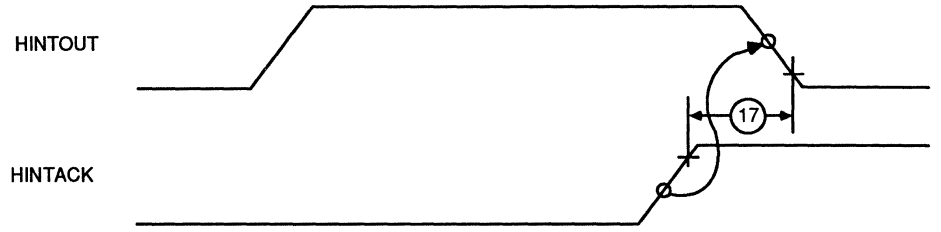
09360B–15

Reset Timing

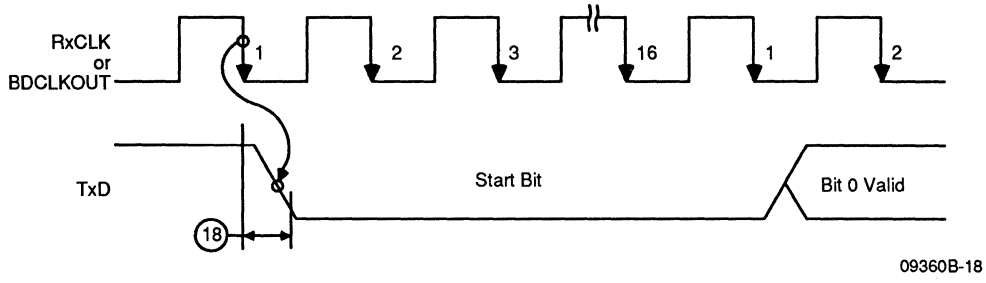


09360B–16

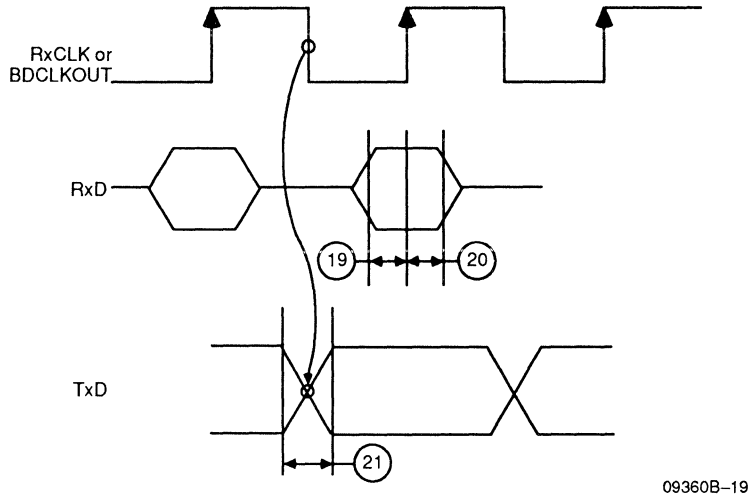
DMA Request Timing



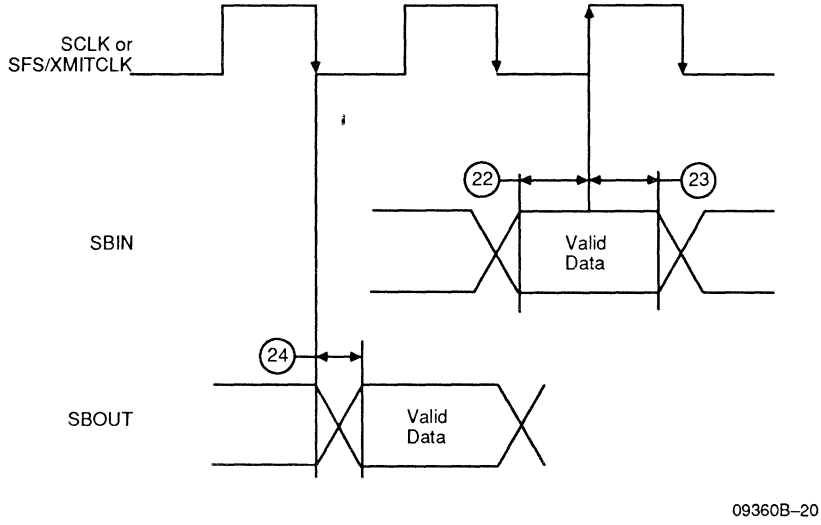
Host Message Available Timing



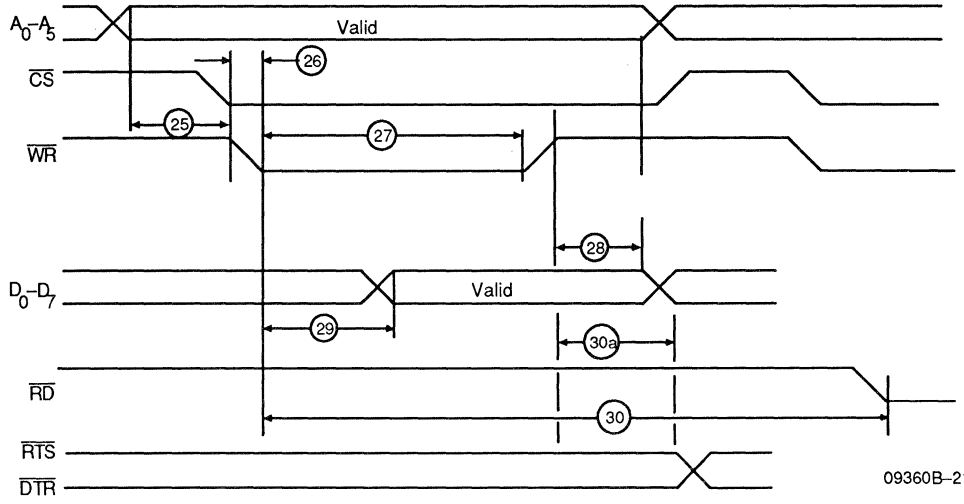
USART Asynchronous Operation



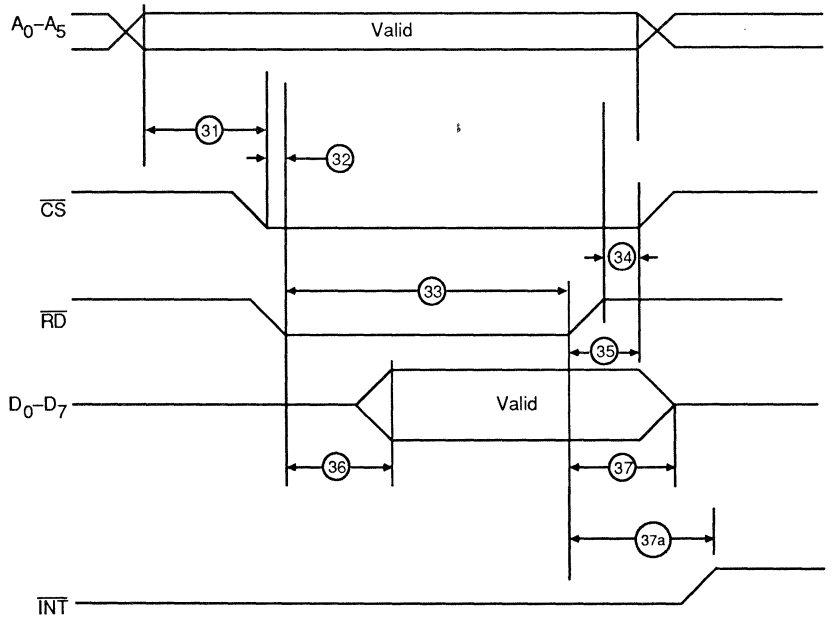
USART Synchronous Operation



Serial Bus Port Timing

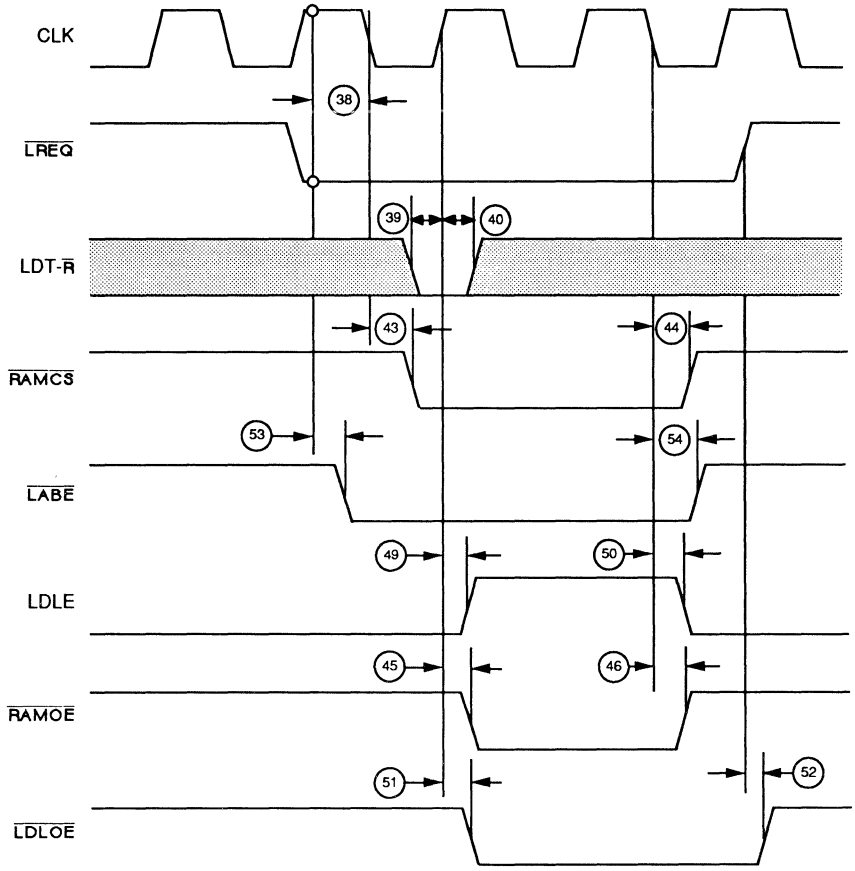


Local CPU Write Timing

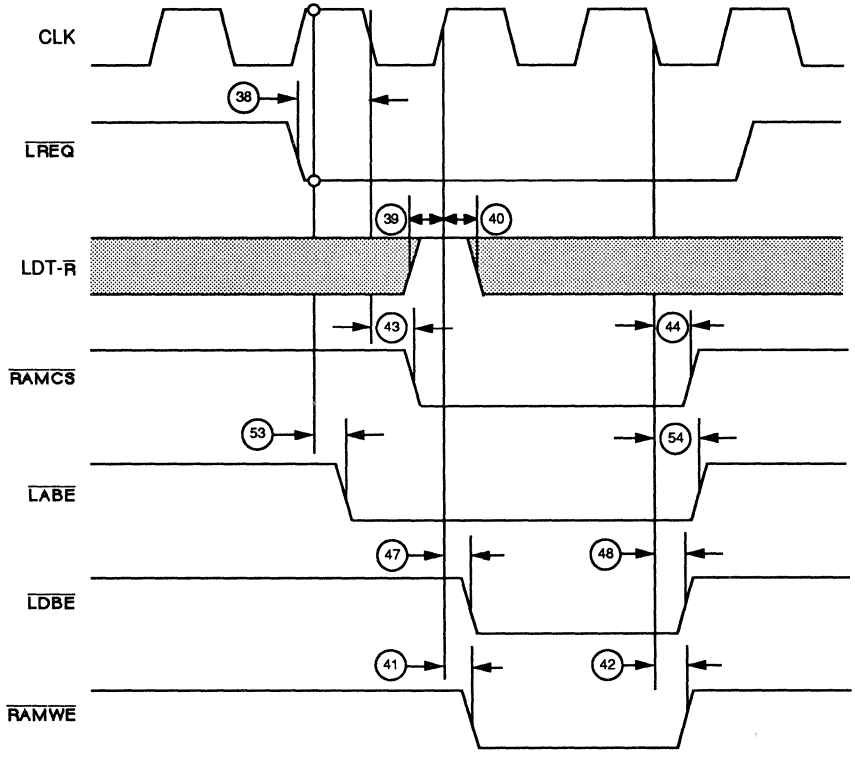


Local CPU Read Timing

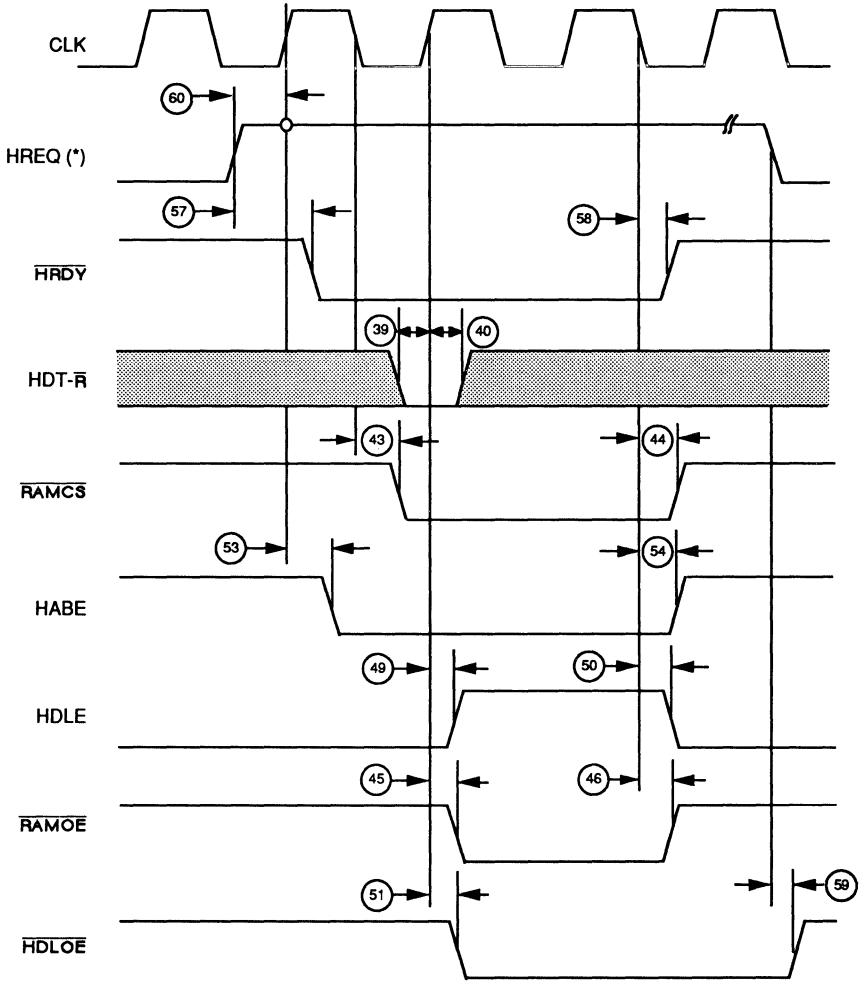
09360B–22



Dual-Port Memory Local Proc. Read Cycle (No contention)

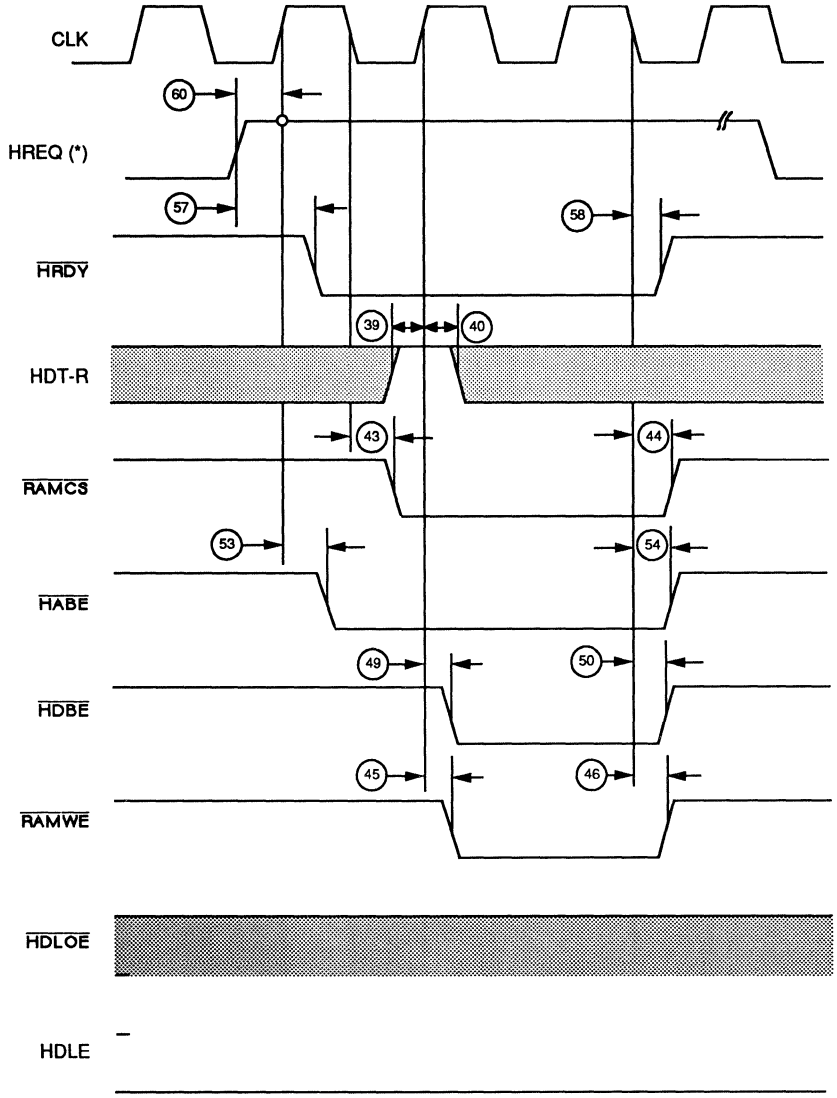


Dual-Port Memory Local Proc. Write Cycle (No contention)



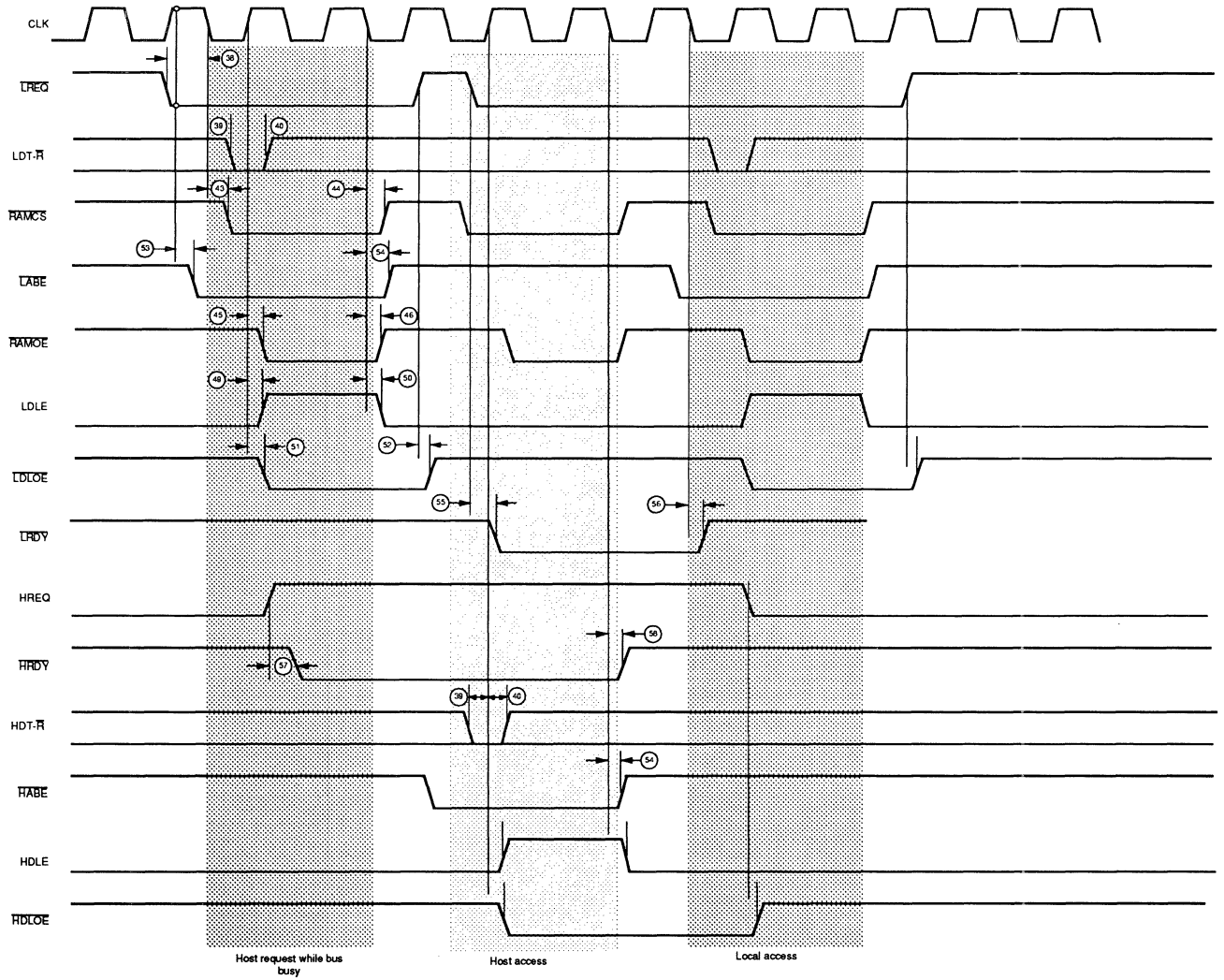
(*) HREQ is internally synchronized by the IDPC. If the setup time to the rising edge is not met, the memory cycle starts on the next rising edge of CLK.

Dual-Port Memory Host Proc. Read Cycle (No contention)



(*) HREQ is internally synchronized by the IDPC. If the setup time to the rising edge is not met, the memory cycle starts on the next rising edge of CLK.

Dual-Port Memory Host Proc. Write Cycle (No contention)

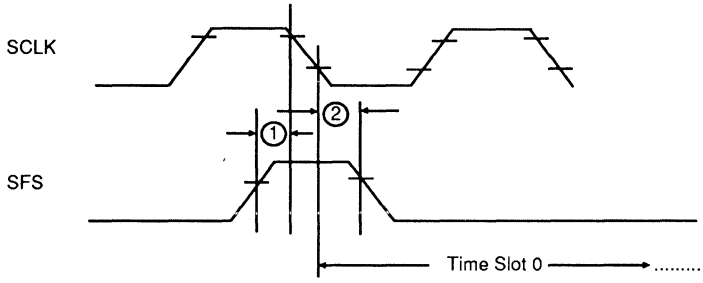


Dual-Port Memory Contention: Local Read—Host Read—Local Read

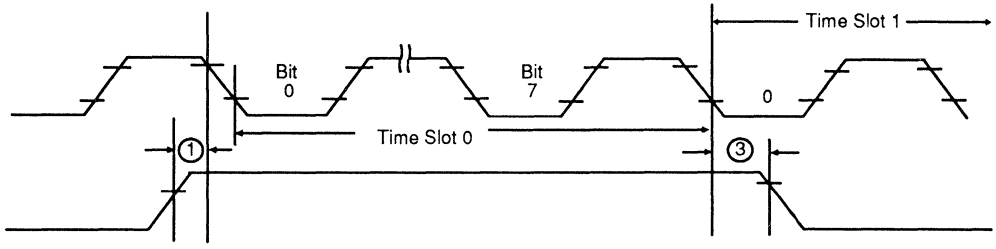
Multiplexed Mode

No.	Symbol	Description	Min.	Max.	Unit
1	t_{FSS}	Frame Sync set up time	50	t_{SCLOL} – 30	ns
2	t_{FSH1}	Frame Sync hold time	25	—	ns
3	t_{FSH2}	Frame Sync hold time	50	—	ns

*$t_{SCHC}L$ = SCLK cycle time.

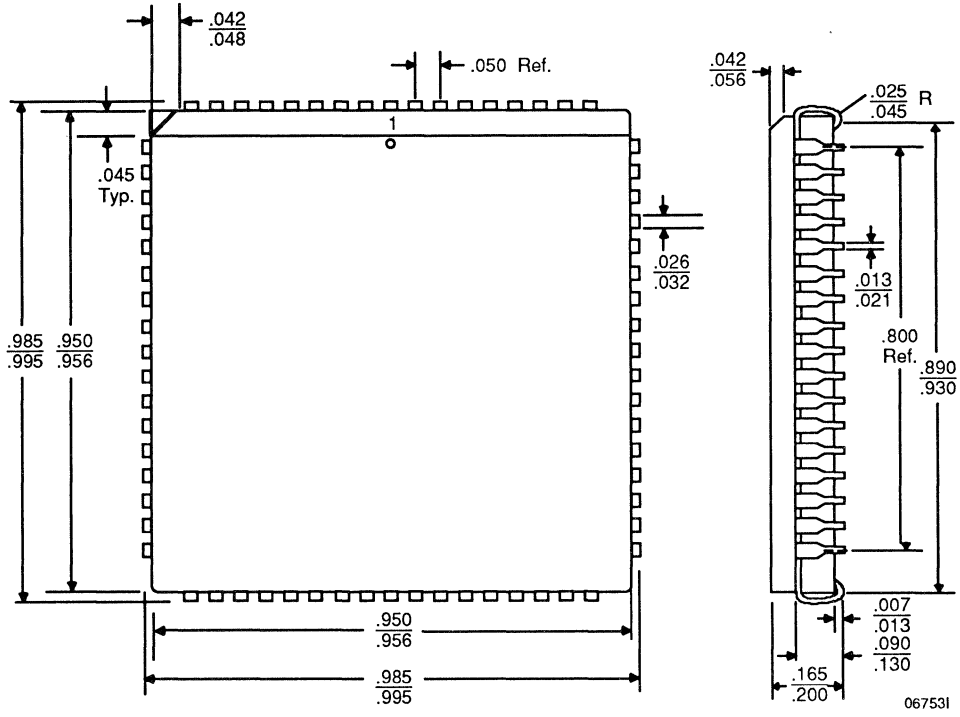


SFS Timing for Multiplexed Mode



SFS Timing for Extended Mode

PHYSICAL DIMENSIONS
PL068



Preliminary

Am2085

ISDN Subscriber Access Controller (ISAC-S)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- S-Bus transceiver according to CCITT I.430
- Recovery of clock and frame
- Frame alignment for trunk line termination
- Access to Echo bit
- Implementation of activation/deactivation procedure according to CCITT I.430
- Support of LAPD protocol
- FIFO buffer (2 × 64 byte) for efficient transfer of D-channel packets
- Serial interfaces for various types of B-channel sources/destinations (SLD, SSI)
- Switching functions for B-channels
- Watchdog timer
- Switching of test loops
- 8-bit microprocessor interface
- Advanced CMOS technology
- Low power consumption

GENERAL DESCRIPTION

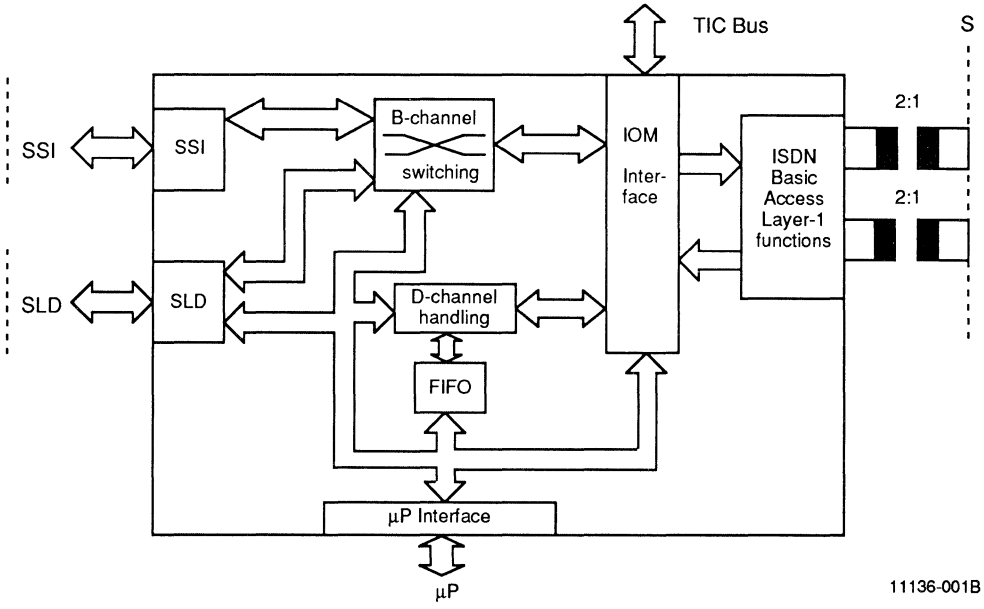
The Am2085 ISAC-S™ is a transceiver circuit able to interface voice/data communication equipment to the four-wire CCITT S-bus. It supports the LAPD protocol in hardware. For an efficient transfer of D-channel packets, FIFO structures are used.

The device is mounted in a 40-pin CMOS package.

The power consumption of the device in the active state is 80 mW (8 mW in Power Down State).

1

BLOCK DIAGRAM



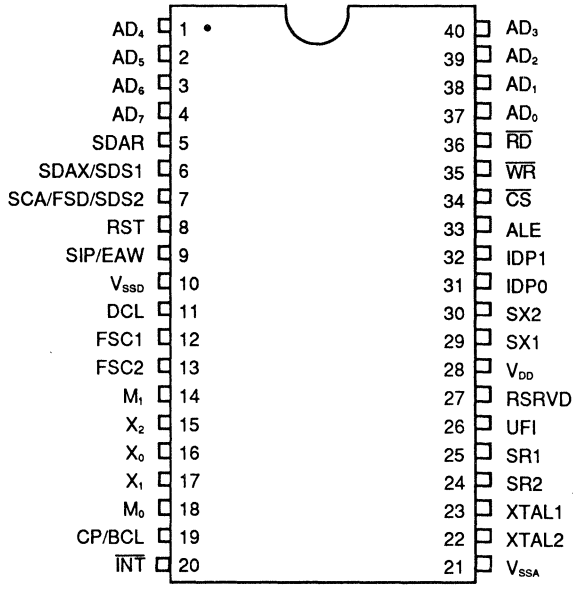
11136-001B

Publication #	Rev.		Amendment
11136	C	&	/1

Issue Date: July 1989

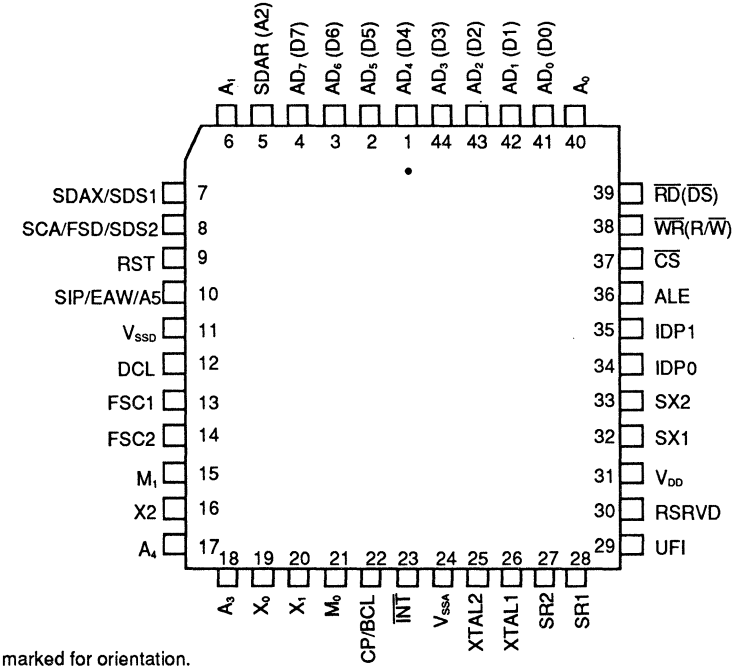
CONNECTION DIAGRAMS
Top View

40-Pin DIP



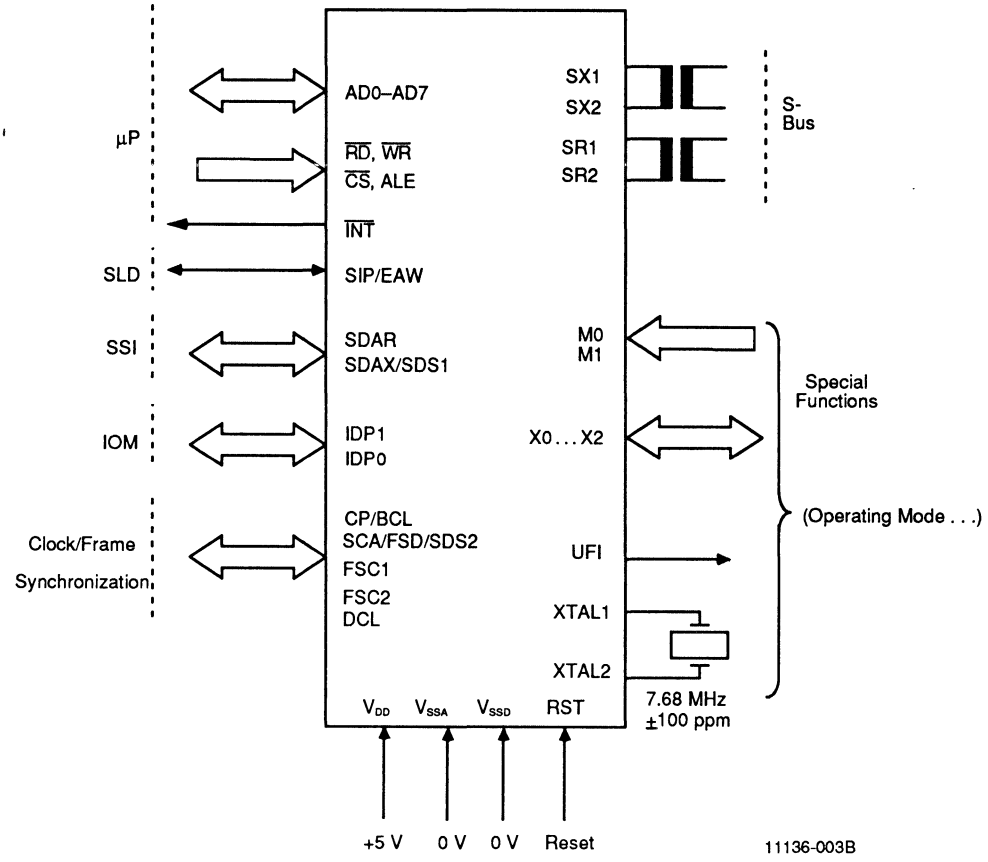
11136-002B

44-Pin PLCC



Note: Pin 1 is marked for orientation.

LOGIC DIAGRAM

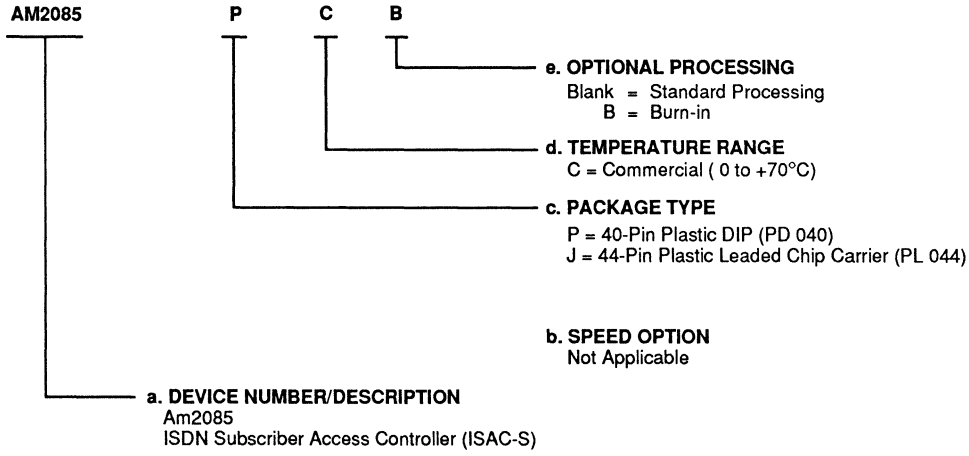


ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM2085	PC, JC, PCB, JCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

AD_0–AD_7

Address Bus (Input)

The multiplexed Address/Data Bus transfers data and commands between the microprocessor System and the ISAC-S.

ALE

Address Latch Enable (Input)

A High on this line indicates an address on the external address/data bus, selecting one of the ISAC-S internal sources or destinations.

CP

(Output)

Synchronized clock output.

$\overline{CS}$

Chip Select (Input; Active Low)

A Low on this line selects the ISAC-S for a read/write operation.

DCL

Clock (Input/Output)

This pin supplies the device clock.

FSC1

Frame Sync 1 (Input/Output)

Supplies the synchronization signal 1: input (LT-S/NT) or output (TE/LT-S).

FSC2

Frame Sync 2 (Input/Output)

Supplies the synchronization signal 2: input (LT-S/NT) or output (TE/LT-T).

$\overline{INT}$

Interrupt (Output; Active Low)

The signal is activated when the ISAC-S requests an interrupt. It is an open-drain output.

M_1, M_0

(Input)

Setting of operating mode.

$\overline{RD}$

Read (Input; Active Low)

This signal indicates a read operation.

$\overline{RST}$

Reset (Input; Acitve High)

A High on this input focus forces the ISAC-S into reset state.

SCA/FSD

Serial Clock Port A/Frame Sync. Delayed (Output)

Depending on the programmed timing mode, this output supports either a 128-kHz clock signal for the SSI port or a delayed 8-kHz frame synchronization signal for IOM interface.

SDAR

Serial Data Port A Receive (SSI) (Input)

This line receives serial data at standard TTL or CMOS levels. An integrated pull-up circuit enables connection of an open-drain/open-collector driver without an external pull-up resistor.

SDAX

Serial Data Port A Transmit (SSI) (Output)

This line transmits serial data at standard TTL or CMOS levels.

SDI

(Input/Output)

Serial Data In, IOM interface.

SDO

(Output)

Serial Data Out, IOM interface.

SIP/EAW

SLD Interface Port (Input/Output)

This line transmits and receives serial data at standard TTL or CMOS levels. When the terminal-specific functions are selected, this line serves as the subscriber awake line.

SR1

(Output)

S-bus receiver, 2.5 V reference output.

SR2

(Input)

S-bus receiver, signal input.

SX1

(Output)

Positive output S-bus transmitter.

SX2

(Output)

Negative output S-bus transmitter.

UFI

(Output)

Connection for an optional external RC circuit.

V_{DD}

Power supply (+5 V ±5%).

V_{SSA}

Analog ground.

V_{SSD}

Digital ground.

$\overline{WR}$

Write (Input; Active Low)

This signal indicates a write operation.

X_2, X_1, X_0

(Input/Output)

Operating mode-specific functions.

XTAL1

(Input)

Connection for crystal or external clock input.

XTAL2

(Output)

Connection for external crystal. Left unconnected if external clock is used.

OPERATIONAL DESCRIPTION

The ISAC-S, designed for the user area of the ISDN basic access, can be used for the following applications corresponding to the appropriate basic operating mode of the ISAC-S:

• **Terminal equipment type 1**	
ISDN feature telephone, extended ISDN terminal	TE Mode
• **Network termination 2**	
PABX, including the functions for the following:	
–line termination on S(LT-S), just as in a digital subscriber line module	LT-S mode (INFO 2 and 4 will be generated automatically)
	NT mode (Info 2 and 4 must be initiated by software)
–line termination on T(LT-T), just as in a digital line trunk module	LT-T mode

The operating mode of the ISAC-S must be selected by pin-strapping, as described in Operating Modes, before Power On Reset.

Reset

After Reset, Layer 1 will have reached the following state:

- G1 deactivated in LT-S/(NT) mode
- F3 standby in TE/LT-T mode

according to CCITT I.430 (see Control of Layer 1). F3 standby state means that the internal oscillator is active, and the DCL clock and the FSC1/FSC2 frame signals are delivered as output signals. The F3 power down state, with a minimum power consumption of not more than 8 mW, can be achieved by programming the CFS-bit = "1" in the ADFR register.

1

Table 1. Subset of ISAC-S Registers with Defined Reset Values (in Hex)

Register	Value After Reset	Meaning
ISTA	00	No interrupts
MASK	00	All interrupts enabled
EXIR	00	No interrupts
STAR	48	XFIFO is ready to be written to RFIFO is ready to receive at least 16 octets of a new message
CMDR	00	No command
MODE	00	Auto mode 1-octet address field External timer mode Receiver inactive IOM interface, monitor channel not used (TE: pt-pt, LT-S/(NT): point-to-multipoint)
RFBC	00	No frame bytes received
SPCR	00	SDI pin = High SIP/SAW pin "High Impedance" (SLD interface deactivated) Timing mode 0 (terminal) IOM interface test loop deactivated SLD B-channel loop is selected SDAX pin = High
STCR	00	Serial interface port for the SLD interface selected TIC bus address 0 No Synchronous Transfer
CIXR	BF	C/I code = 1111 TE-channel data = 1 TIC bus is not requested for transmitting a C/I code
ADFR	00	No prefilter Active clock signals (Standby) in TE mode Adaptive timing (point-to-multipoint) in NT/LT-S FSC1/FSC2 frame signals are not inverted in TE mode Interframe time fill = consecutive 1

Initialization

During initialization phase the appropriate registers must be programmed according to the application and the desired features, as listed in Table 2.

Table 2. Programming for Initialization Phase

Function	Register	Effect
Special Functions	ADFR	Prefilter Disabled Layer-1 functions Standby/power-down selection in TE mode, fixed (point-to-point, S interface), adaptive (S-bus) timing in LT-S mode B1/B2-channel assignment (SSI) in TE mode Interframe time fill in HDLC port mode
Masking Interrupts	MASK	Masking of selective interrupt sources
D-channel (HDLC port)	MODE	Message transfer mode 2-octet/(1-octet) address Timer mode: external/internal (auto mode only) IOM interface mode: IOM: point-to-point and point-to-multipoint (LT-S,NT) IOM: point-to-multipoint (TE) HDLC port: (TEM-bit 1)
	TIMR	N1 and T1 in internal timer mode (TDM-bit in mode) T2 in external timer mode
	XAD1	SAPI: LAPD transmit address octet (auto mode only)
	XAD2	TEI:
	SAP1/SAP2	SAPI: LAPD receive address octet for the internal address recognition
	TEI1/TEI2	TEI:
Serial interface B-channel switching, Terminal-specific functions	SPCR	SLD port inactive/active Timing mode 0, 1 IOM interface loop (reduced timer resolution) B-channel switching
	STCR	Terminal-specific functions/SLD interface TIC bus address
	CIXR	Subscriber/Exchange Awake, Watchdog Timer

Table 3. Characteristics and Typical ISDN Applications of the Message Transfer Modes

	Message Transfer Modes				
	Auto Mode	**Non-Auto Mode**	**Transparent Mode**	**Ext. Transparent Mode 1**	**Ext. Transparent Mode 0**
Characteristics	One logical link (SAP1, TEI1) can be handled autonomously Window Size (WZ) = 1 Full address recognition (SAPI, TEI)	Full address recognition (SAPI, TEI)	SAPI address recognition	SAPI address recognition	No address recognition fully transparent
Typical Application	Terminal: WZ = 1 Exchange: point-to-point configuration WZ = 1	Terminal: WZ ≥ 1 Exchange: point-to-point configuration WZ ≥ 1	Exchange: bus configuration	Exchange: bus configuration	Diagnostic

Processing

Assuming the ISAC-S has been initialized for typical applications in the user area of the ISDN basic access, the ISAC-S is now ready to transmit and receive messages in the D-channel (LAPD support). As a prerequisite for that, the Layer 1 must be previously activated.The control of the data transfer phase is mainly done by commands from microprocessor to ISAC-S via the CMDR register and by interrupt indications from ISAC-S to microprocessor (ISTA and EXIR register).

The two B-channels (B1/B2) can be switched in a highly flexible manner between the S interface (IOM interface) and the SSI and SLD interface.

Control of Layer 1

The management commands of Layer 2, programmed in the CIXR register, trigger certain procedures in Layer 1. The responses from Layer 1 can be read from CIRR register after a CIC interrupt (ISTA). An example of activation and deactivation with the respective commands and indications is depicted in Figure 4.

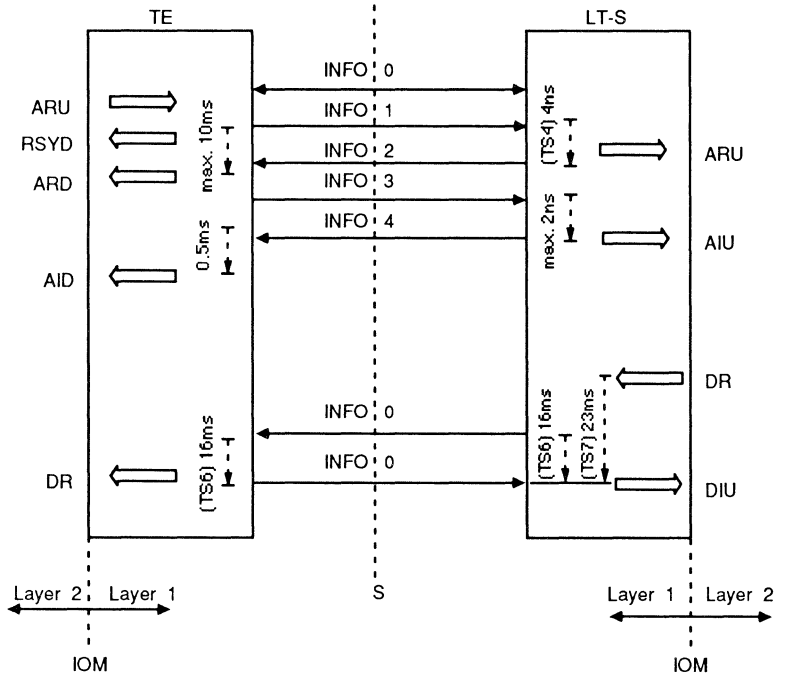


Figure 4. Example of Activation/Deactivation

Tables 4 to 6 contain the command/indication codes in the different operating modes:

Table 4. Commands and Indications in LT-S Mode

Command (downstream)	Abbr.	Code	Remark
Deactivate request	DR	0000	(x)
Send continuous zeros	SCZ	0001	Transmission of AMI pulses at a frequency of 96 kHz (x)
Send single zeros	SSZ	0010	Transmission of AMI pulses at a frequency of 2 kHz (x)
Activate request	ARD	1000	
Activate request loop	ARL	1010	Activate request for loop 2
Deactivate indication	DID	1111	Deactivation acknowledgment, quiescent state
Indication (upstream)			
Lost signal level	LSL	0001	No receive signal
Lost framing	RSYU	0100	Receiver is not synchronous
Activate request	ARU	1000	Info 1 received
Activate indication	AIU	0100	Synchronous receiver
Deactivate indication	DIU	1111	Timer TS6 or TS7 expired after deactivation command

Note: (x) = unconditional commands

Table 5. Commands and Indications in NT Mode

Command (downstream)	Abbr.	Code	Remark
Deactivate request	DR	0000	(x)
Send continuous zeros	SCZ	0100	Transmission of AMI pulses at a frequency of 96 kHz
Activate request	ARD	1000	Transmission of info 2
Activate request loop	ARL	1010	Transmission of info 2, switching of test loop 2
Deactivate indication	DID	1111	Deactivation acknowledgment, quiescent state
Activate indication	AID	1100	Transmission of info 4
Activate indication loop	AIL	1110	
Indication (upstream)			
Timing	TIM	0000	Clocks are required
Lost signal level	LSL	0001	No receive level
Lost framing	RSYU	0100	Receiver is not synchronous
Error indication	EI	0110	RST and SCZ both active
Activate request	ARU	1000	Info 1 received
Activate indication	AIU	1100	Synchronous receiver
Deactivate indication	DIU	1111	Timer TS6 or TS7 expired after deactivation command

Note: (x) = unconditional commands

Table 6. Commands and Indications TE/LT-T

Command (downstream)	Abbr.	Code	Remark
Timing	TIM	0000	Clocks are required
Reset	RS	0001	(x)
Send single zeros	SSZ	0010	Transmission of AMI pulses at a frequency of 2 kHz (x)
Activate request, priority 8	AR8	1000	Activation command set D-channel priority to 8
Activate request, priority 10	AR10	1001	Activation command set D-channel priority to 10
Activate request	ARL	1010	Activation of test loop 3 (x)
Deactivate indication	DIU	1111	IOM interface can be switched into idle state
Indication (upstream)			
Power up	PU	0111	IOM clocking is provided
Deactivate request	DR	0000	Deactivation request by S
Slip detected	SD	0010	Wander is larger than 18 µs peak-to-peak
Disconnected	DIS	0100	Pin CON connected to GND
Error indication	EI	0110	(RST = 1 and CFS-bit = 0) or RS
Level detected	RSY	0100	Signal received, receiver
Activate request	ARD	1000	Info 2 received
Test indication	TI	1010	Test loop 3 activated or continuous zeros transmitted
Activate indication with priority class 8	AI 8	1100	Info 4 received, D-channel priority is 8 or 9
Activate indication with priority class 10	AI 10	1101	Info 4 received, D-channel priority is 10 or 11
Deactivate indication	DID	1111	Clocks will be in disabled quiescent state

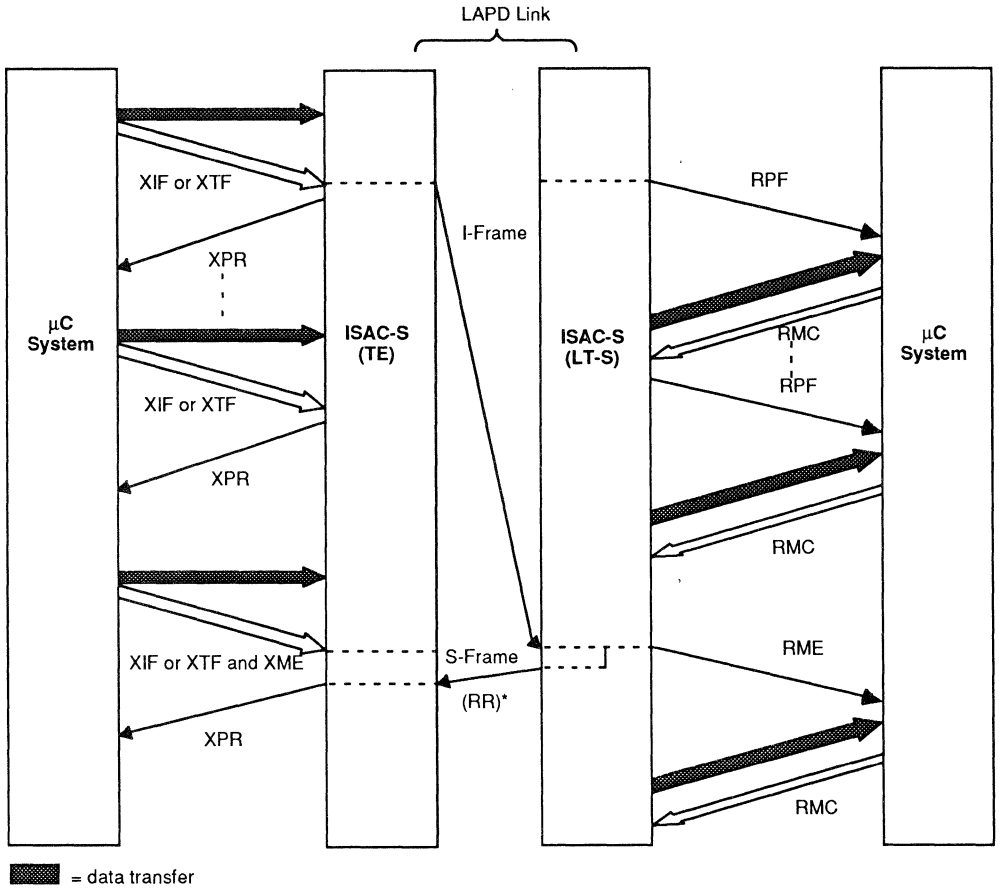
Note: (x) = unconditional commands

1

Transfer of LAPD frames in the D-Channel

When Layer 1 of the S interface is activated, the ISAC-S is able to transmit and receive LAPD frames via this international standardized interface in a highly sophisticated manner.

The LAPD protocol support depends on the selected message transfer mode (Table 5 and Layer 2 Functions section). The powerful FIFO structure of the ISAC-S, which consists of a 2 × 32 byte receive and a 2 × 32 byte transmit FIFO, as well as an intelligent FIFO controller, builds a flexible connection between the LAPD controller of the ISAC-S and upper layer protocol functions in the microcontroller system via the microprocessor interface. Assuming a normally running communication link (Layer 1 activated, Layer 2 link established, TEI assigned, and so on), Figure 5 demonstrates the transfer of an I-frame via the D channel.



*In auto mode, the "RR" response will be transmitted autonomously (provided operation is normal)

11136-005B

Figure 5. Transmission of an I-Frame in the D-channel (Subscriber to Exchange)

The following table summarizes the commands which can be programmed by setting appropriate bits in the CMDR register to Control Layer 2 (see also Detailed Register Description).

Table 7. CMDR Register Bits for Layer 2 Control

Com.	HEX	Bit 7–0	Meaning
RMC	80	1000 0000	Receive message complete
RHR	40	0100 0000	Reset HDLC receiver
RNR	20	0010 0000	Receiver not ready (auto mode)
STI	10	0001 0000	Start timer
XTF	08	0000 1000	Transmit transparent frame without closing the frame
XIF	04	0000 0100	Transmit "auto mode" I-frame without closing the frame
XTFC	0A	0000 1010	Transmit transparent frame and close frame
XIFC	06	0000 0110	Transmit "auto mode" I-frame and close frame
RHX	01	0000 0001	Reset HDLC Transmitter

Interrupt List

In the following table, all interrupts of the ISAC-S are listed together (see also Detailed Register Description).

Table 8. Meaning of ISAC-S Interrupts (Layer 1 and Layer 2)

	Interrupt			Meaning	Reaction (ISDN)
Layer 2 receive	RPF	ISTA	Receive Pool Full	Request for reading received octets of a LAPD frame from RFIFO.	Read the octets and acknowledge with RMC command.
	RME	ISTA	Receive Message End		
	RFO	EXIR	Receive Frame Overflow	A frame has been lost. The microcontroller has failed the minimum reaction time.	Error report for statistical purposes only.
	PCE	EXIR	Protocol Error	S- or I-frame with incorrect N(R) or S-frame with I-field received, (in auto mode only).	Error report Data link release indication to Layer 3.
	TIN	ISTA	Timer interrupt	External timer expired or, in auto mode, internal timer (T200) and repeat counter (N200) both expired.	Error report Data link release indication to Layer 3 (no acknowledgment from peer entity).
Layer 2 transmit	XPR	ISTA	Transmit Pool Ready	Acknowledgment that further octets of an LAPD frame can be written to the XFIFO.	Write further octets to the XFIFO and subsequently request (further) transmission with X_F or X_FC.
	RSC	ISTA	Receive Status Change	A status change from peer has been received, RR/RNR frame.	Read STAR register, check RRNR-bit and report it.
	XMR	EXIR	Transmit Message Repeat	Frame must be repeated due to a transmission error and/or a received negative acknowledgment.	Transmission of this frame must be repeated. No indication to Layer 3.
	XDU	EXIR	Transmit Data Underrun	Frame has been aborted because the XFIFO holds no further data (messages greater than 32 octets).	
Layer 1	CIC	ISTA	C/I Code Change	A change of indication from layer-1 has been detected.	Read CIRR register and report reason to management entity.
	MOR	EXIR	—	Not used in the ISAC-S.	
Synchronous transfer	SIN	ISTA	Synchronous Transfer Interrupt	Synchronization of microprocessor and data transfer via serial interfaces.	Access to B-channel registers BCX1/2, BCR1/2, or SFCR. Has to be confirmed by setting the appropriate ST0/(1)-bit in STCR within a time limit.
	SOV	EXIR	Synchronous Transfer Overflow	The SIN interrupt was not confirmed on time by setting the appropriate ST0/(1)-bit in STCR.	Abnormal error condition. Revise access software.
Terminal-specific functions	SAW	EXIR	Subscriber Awake	Indicates a falling edge on SAW line (terminal-specific functions are selected.	Switch into "Power Up" state and start data link establish procedure.
	WOV	EXIR	Watchdog Timer Overflow	Watchdog timer has been expired (terminal-specific functions are selected).	Worst error condition. Restart system software.
	EXI	ISTA	Extended Interrupt	An interrupt indicated in EXIR has occurred.	Read EXIR and determine interrupt source.

FUNCTIONAL DESCRIPTION

The Am2085 performs the Layer 1 functions of the ISDN basic access as well as B-channel switching and wide-spread functional support for Layer 2.

General Functions and Device Architecture

The detailed block diagram of the ISAC-S is shown in Figure 6.

The left side of the diagram contains the Layer 1 functions, according to CCITT I series recommendations:

- S-bus transmitter and receiver
- Timing recovery and synchronization by means of digital PLL circuitry
- Activation/Deactivation
- D-channel access

Furthermore the following diagnostic tests are implemented:

- Test loop 2 (NT/LT-S) and 3 (TE/LT-T) close to the S-bus
- Send single AMI pulses at 2 kHz and send continuous AMI pulses at 96 kHz

The right side consists of Layer 2 functions to support LAPD and provides B-channel switching capabilities.

In a special operating mode, the auto mode, the ISAC-S processes information transfer and procedure handshakes (I- and S-frames) of the LAPD protocol autonomously.

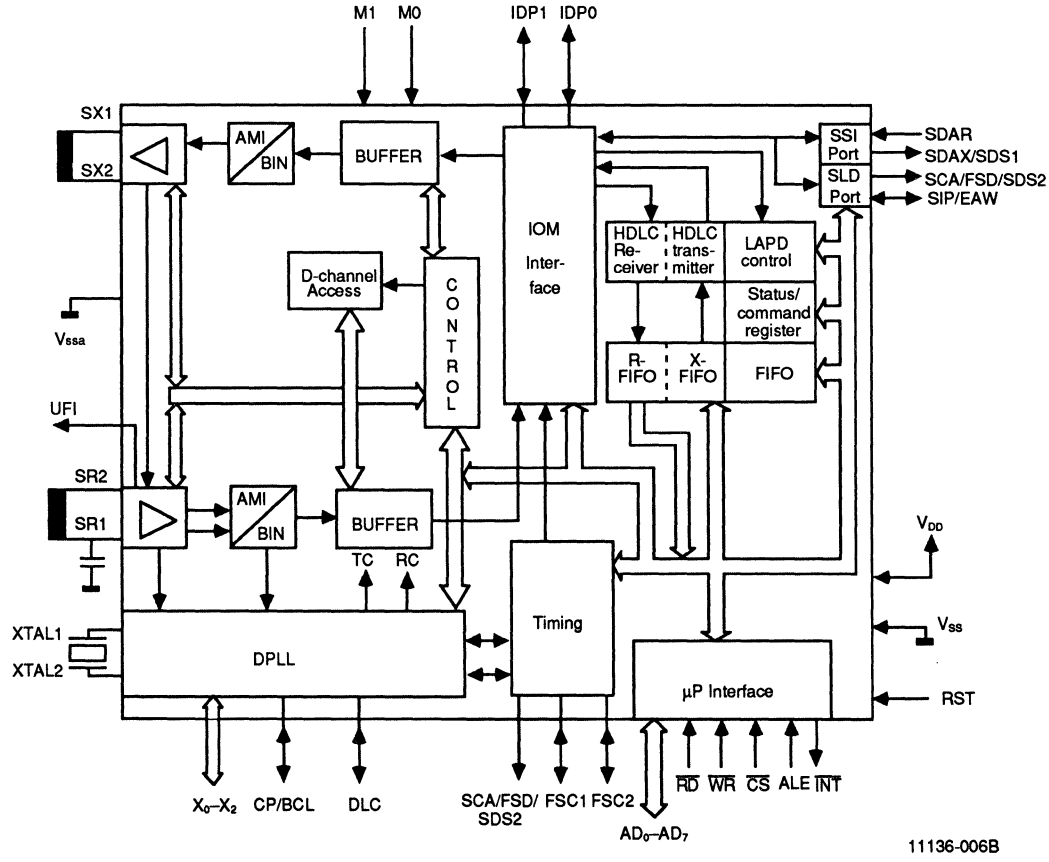


Figure 6. ISAC-S Device Architecture

Two serial interfaces to B-channel sources/destinations are realized:

- The standard SLD interface which is a bidirectional (ping-pong) 256-kb/s interface primarily optimized for telecommunication applications
- The full-duplex 128-kb/s interface, SSI, which can serve as a general interface in TEs to transfer the two B-channels

Control and monitor functions as well as data transfers (D-channel messages, transparent B-channel data) by the user's CPU is performed through a standard 8-bit microprocessor interface. A highly sophisticated 2 × 64 byte FIFO structure for both directions enables a flexible D-channel message information exchange between the LAPD (HDLC) controller and a microcontroller system.

The timing unit is responsible for the system clock and frame synchronization. Pin-strapping determines its operating mode.

Operating Modes

The Am2085 is configurable for the following applications:

- ISDN terminals (TE)
  —> TE (mode)
- ISDN subscriber line termination (LT-S)
  —> LT-S/(NT) mode
- ISDN trunk line termination (PABX connection to central office)
  —> LT-T mode

Configuration is performed by pin-strapping (pins M_2, M_0), yielding different meanings to the multifunctional pins (X_0, X_1, X_2) as well as the clock and framing signal pins (DCL, FSC1, FSC2, CP).

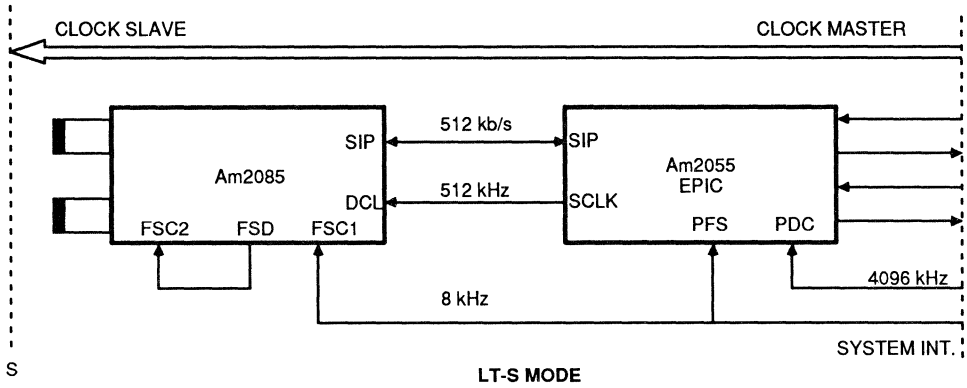
Table 9. Operating Modes and Functions of Mode-Specific Pins of Am2085 ISAC-S

Application	M2	M0	DCL	FSC1/2	CP	X2	X1	X0
TE	0	0	O: 512 kHz*	O: 8 kHz*	O: 1536 kHz*	O: ECHO	O: 3840 kHz	I: CON
LT-T	0	1	I: 512 kHz	I: 8 kHz	O: 512 kHz*	I: fixed at 0	I: fixed at 0	I: CON
LT-S1	1	0	I: 512 kHz	I: 8 kHz	I: fixed	I: fixed at 0	O: 7680 kHz at 0	I: fixed at 0
NT	1	1	I: 512 kHz	I: 8 kHz	I: SCZ	I: SSZ	I: fixed at 0	_

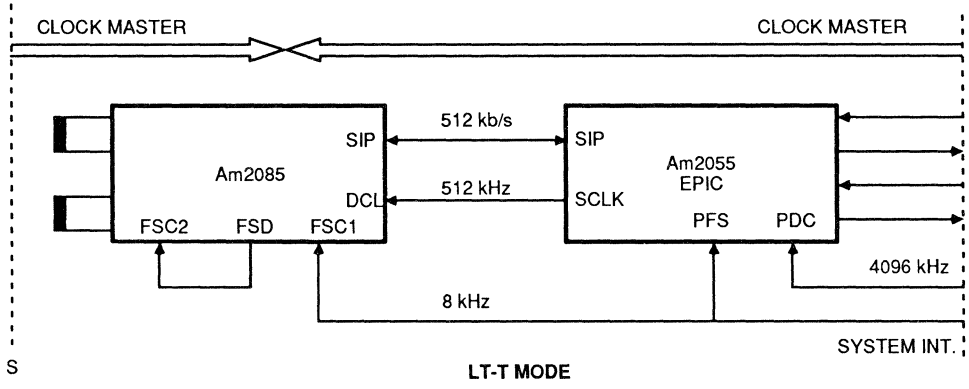
*synchronized to S I = Input O = Output

Notes:
- ECHO: Reproduces the E-bits received from the S interface synchronously to IOM frame "D"-bits (bit positions 24 and 25 of IOM frame). All other bit positions are binary 1.
- CON: Connected to S-bus
- SCZ: Send continuous binary zeros (96 kHz)
- SSZ: Send single binary zeros (2 kHz)

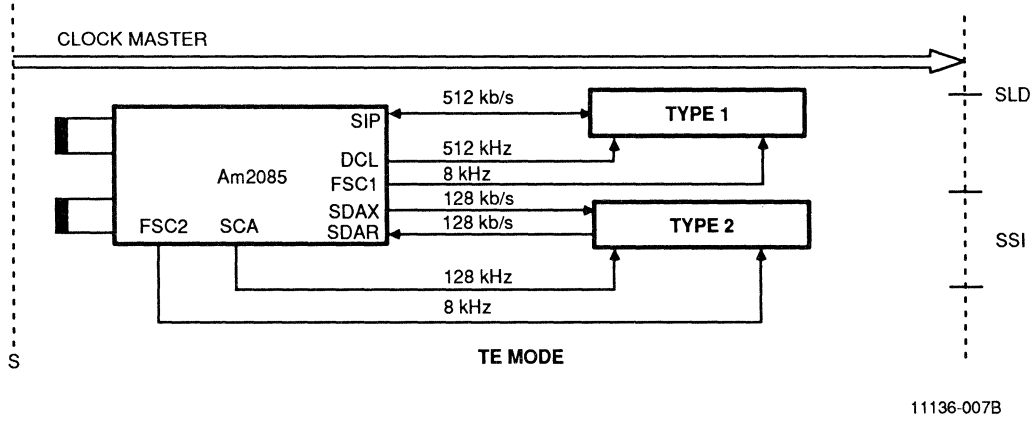
The different operating modes in relation to the timing recovery are illustrated in Figure 7.



LT-S MODE



LT-T MODE



TE MODE

11136-007B

Figure 7. Operating Modes of ISAC-S

Interfaces

The ISAC-S serves three different user-oriented interface types:

- parallel processor interface to higher layer functions
- SSI and SLD as interfaces for B-channel sources/destinations
- IOM interface; to the Layer 1 functions of the ISDN basic access (TIC bus)

Microprocessor Interface

The microprocessor interface consists of bus transceiver, address register, and bus control logic. Via this interface, the ISAC-S can be connected to the multiplexed address/data bus of a microcontroller system. The following functions can be performed by writing and reading special registers in the ISAC-S (see Detailed Register Description section):

- Transfer of data packets in the D-channel
- Control of Layer 2 functions for the ISDN basic access
- Switching of B-channels
- Access to the B-channels
- Control of Layer 1 functions for the ISDN basic access
- Support of diagnostic functions

In the case of special events in the ISAC-S, the processor is notified by interrupt. The interrupt source can be determined and acknowledged by means of the ISAC-S registers ISTA, EXIR, and MASK (see Detailed Register Description).

Serial Synchronous Interface (SSI)

The serial port SSI serves as a full-duplex connection to B-channel sources/destinations in terminal equipment with a data rate of 128 kb/s. SSI consists of one data line for each direction (SDAX and SDAR), the 8-kHz frame synchronization signal (FSC1 and/or FSC2), and the 128-kHz clock signal (SCA/FSD). This serial interface allows the possible connection of serial synchronous transceiver devices (USART Am82520 HSCC) and various CODEC filters directly to the ISAC-S, as illustrated in Figure 8.

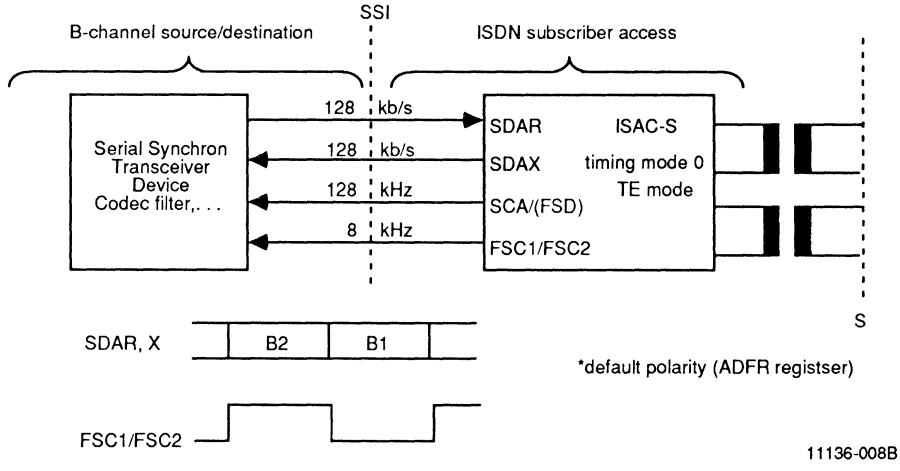


Figure 8. Connection of B-channel Sources/Destinations to the ISAC-S via SSI

Programming the FSC1/FSC2-bit in the ADFR register makes it possible to independently program the strobe signals FSC1/FSC2 so that either B1 or B2 is selected for further processing by the terminal device. The microcontroller system has access to B-channel data via the ISAC-S registers BCR1/BCR2 and BCX1/BCX2.

The microprocessor access must be synchronized to the serial transmission process by means of the Synchronous Transfer Interrupt (STCR; see Detailed Register Description).

SLD Interface

The standard SLD interface is a three-wire interface with a 512-kHz clock (DCL), an 8-kHz frame direction signal (exchange: FSC1 only, terminal: FSC1 and FSC2), and a serial ping-pong data lead (SIP) with an effective full-duplex data rate of 256 kb/s.

The SLD interface can be used in:

- Terminal configurations (timing mode "0") as a full-duplex time-multiplexed (ping-pong) connection to B-channel sources/destinations.
- Digital exchange configurations (timing mode "1") as a full-duplex time-multiplexed connection of B-channel sources/destinations, in this case the ISAC-S itself, to a peripheral board controller. In a typical line-card application the PBC performs time slot assignment of the B-channels to PCM highways, building a system interface to a switching network and a central processor as shown in Figure 10.

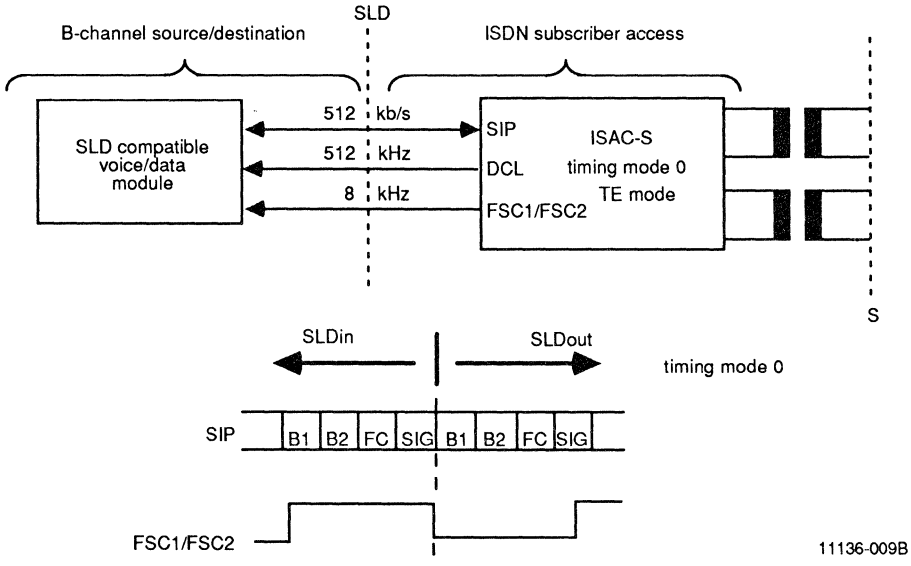


Figure 9. Connection of B-channel Sources/Destinations to the ISAC-S via SLD

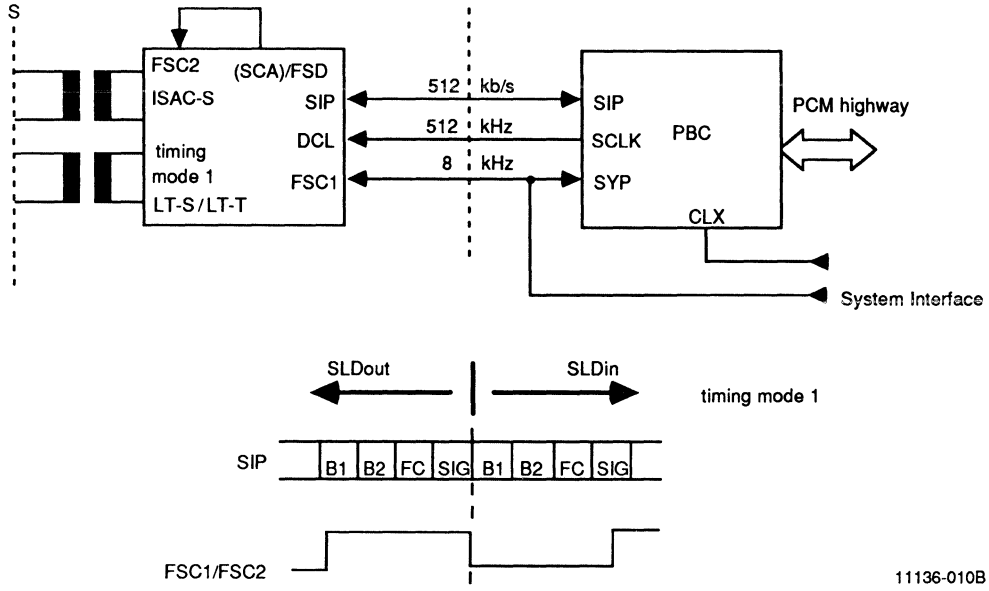


Figure 10. Connection of the ISAC-S as B-channel Source/Destination to a Peripheral Board Controller (PBC)

1

The microcontroller system has access to B-channel data, the Feature Control Byte (FC) and the Signaling Information (SIG) via the ISAC-S registers:

- BCR1/2 and BCX1/2 <—B1/B2
- SFCR <—FC
- SSGR and SSGX <—SIG

The microprocessor access to BCR1/2, BCX1/2 and SFCR must be synchronized to the serial transmission process by means of the Synchronous Transfer Interrupt (STCR) and the BVS-bit (STAR).

ISDN Oriented Modular (IOM) Interface

Although the ISAC-S combines the Layer 1 functions of an Am2080 S-bus transceiver (SBC) with the HDLC functions on one chip, the IOM interface is externally still available for Telecom IC (TIC) bus applications. The TIC bus allows the possible connection of up to seven additional IOM-compatible Communications Controllers via the Layer 1 functions of the ISAC-S to CCITT's S interface as shown in Figure 11.

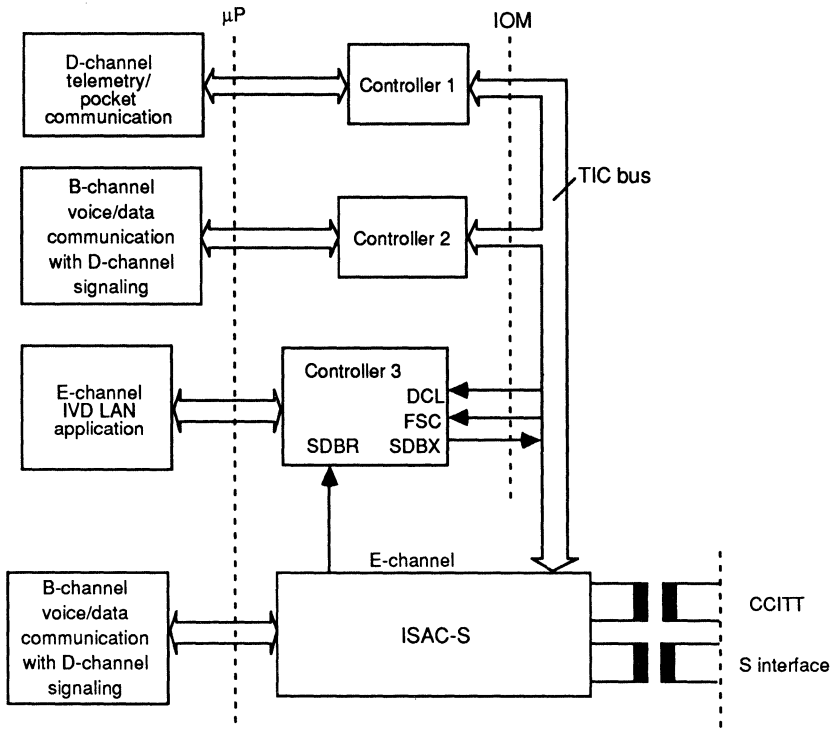


Figure 11. Typical TIC Bus Application

The IOM interface consists of one data line per direction (SDBR and SDBX). Three additional signals define the data clock (DCL) and the frame synchronization (FSC1/FSC2) at this interface. These signals are internally derived from the S interface and are delivered by the ISAC-S in timing mode 0 (terminal).

In timing mode 1 (exchange), the clock and a synchronization signal are provided by the system. In this case, the IOM interface is synchronization by a synchronization signal SCA/FSD, delayed in time with respect to the frame synchronization signal supplied by the system. This reduces the round-trip delay time (see Figure 12).

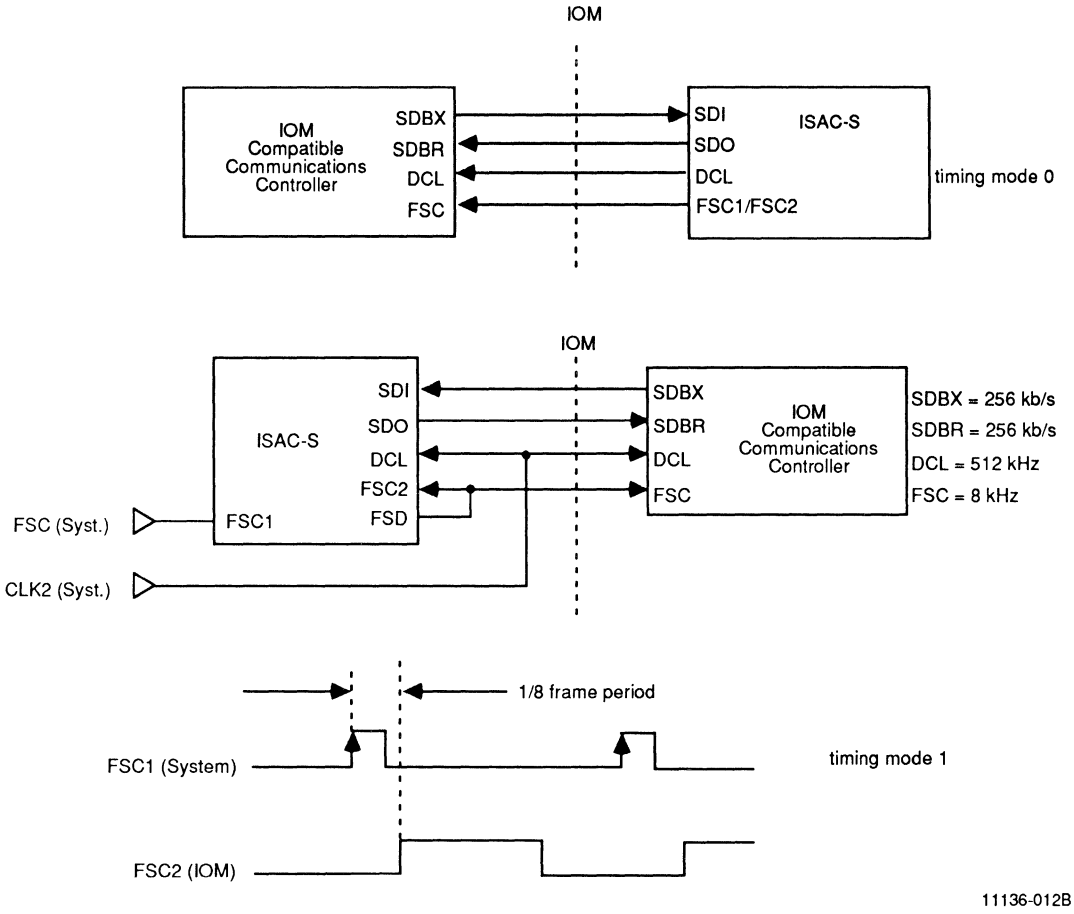


Figure 12. IOM Interface Signals

The IOM interface has two different clocking states:

- Idle state <— FSC1/FSC2 and DCL are disabled and both data lines are logical High (power down)
- Clocked state <— FSC1/FSC2 and DCL are enabled (stand by)

Unlike digital exchange configurations in which the IOM interface always remains in the synchronized state, in terminal equipment both clock states can be selected.

The transition from idle state to clocked state will be automatically initiated by an incoming call from network side. An activation of the IOM interface from the subscriber end has to be programmed in this case by setting and resetting the SPU-bit in the SPCR register, before the IOM interface can be used (that is, for the activation/deactivation procedure at the S interface, see Operational Description section).

The IOM Channel Structure

The channel frame structure of the IOM interface and the related channel capacity are defined in Figure 13 as follows:

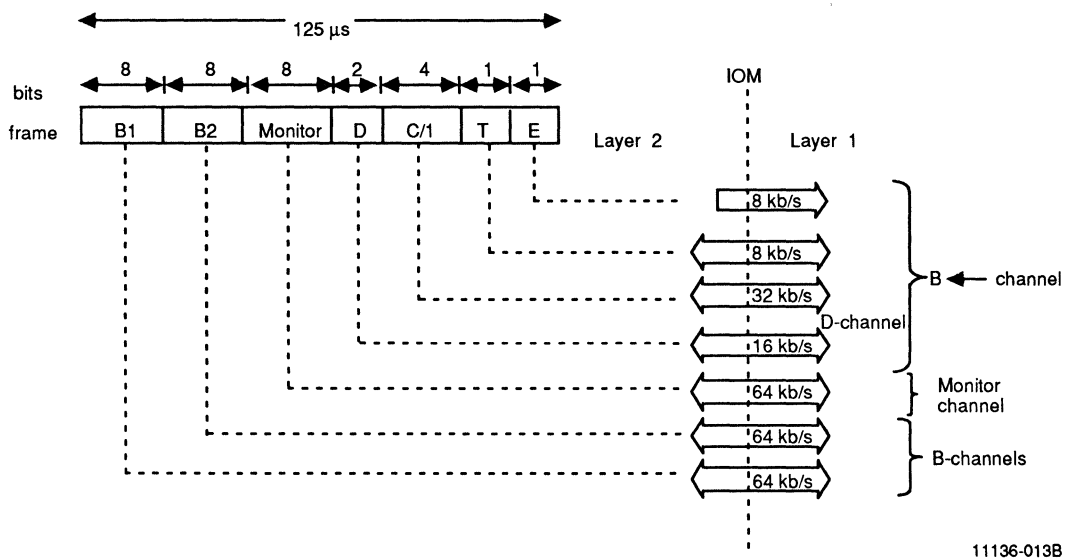


Figure 13. IOM Channel Capacity

In the ISAC-S, the monitor channel supports the TIC bus access mechanism as well as the indication of the S-bus status (D-channel access). The ISAC-S indicates by means of monitor bit 3 (BAC) whether or not it occupies the TIC bus (B-channel).

The control of the Layer 1 functions, especially the activation/deactivation procedure at the S interface, will be done by the exchange of special 4-bit C/I-codes in the C/I channel.

The T-channel, which is fully transparent, and the E-channel are reserved for future use.

TIC Bus

The TIC bus is operated in point-multipoint configuration. It uses a wired-OR connection of the data outputs to enable the TIC bus access mechanism. Therefore the ISAC-S contains an internal pull-up resistor at the SDI pin.

The TIC bus is controlled by a collision resolution mechanism in the monitor channel similar to the D-channel access on the S-bus.

Individual Functions

Distinctive functions for the ISDN basic access realized in the ISAC-S are:

- B-channel switching
- Layer 1 functions
- Layer 2 functions
- Terminal-specific functions
- Test functions

B-Channel Switching

The ISAC-S contains two serial synchronous interfaces which can serve as interfaces to B-channel sources/destinations.

- SSI —> 128 kb/s data rate, B1 and B2
  —> one data line per direction
- SLD —> 256 kb/s data rate, B1 and B2 (and also FC and SIG)
  —> one data line for both directions (ping-pong)

Both channels B1 and B2 can be switched independently of one another to the IOM interface and to the four-wire S interface. Furthermore, it is possible to program a loop for B-channel data received from the IOM or SLD interface. The microcontroller can select the B-channel routes in the SPCR register and has access to the B-channels by writing or reading the BCR/BCR2 and BCX1/BCX2 registers (Figure 14). Synchronization to the 8-kHz frame is done by means of a Synchronous Transfer programmed in the STCR register (SIN Interrupt indicated in the ISTA register and SOV interrupt in EXIR).

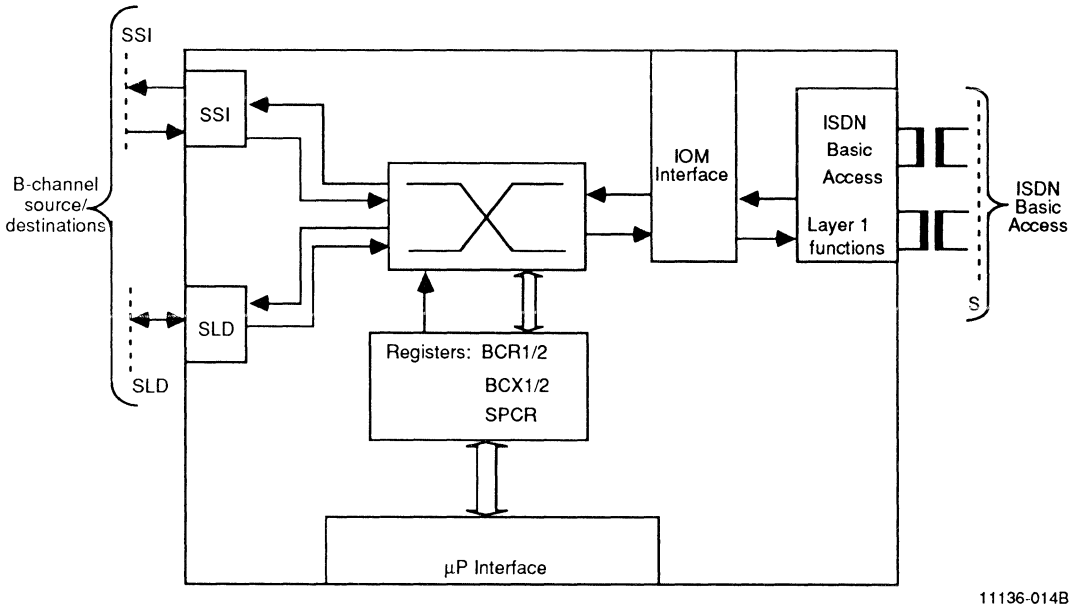
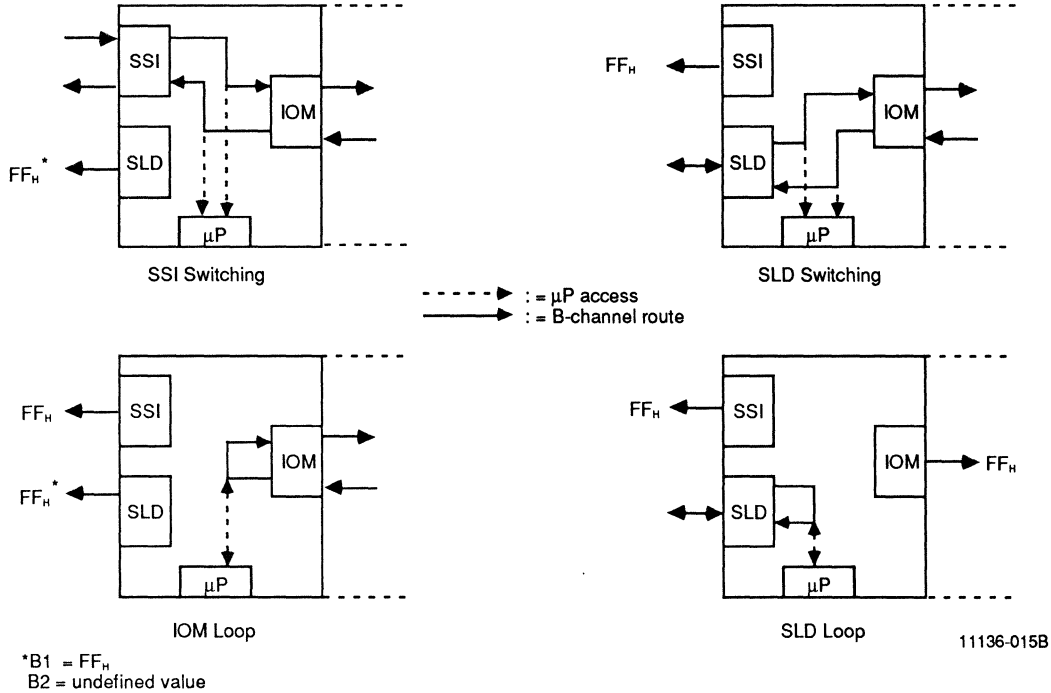


Figure 14. Principle of B-channel Switching

In Figure 15 all possible selections of the B-channel routes and access to B-channel data via the micoprocessor interface are illustrated.



*B1 = FF_H
B2 = undefined value

Figure 15. B-channel Routes and Access to B-channel Data

Layer 1 Functions for the ISDN Basic Access

The S-bus interface circuit in the ISAC-S performs the Layer 1 functions for the S/T interface of the ISDN basic access according to CCITT I.430. The distinctive functions are listed below:

- S-bus transceiver according to CCITT I.430
- Recovery of clock and frame in all applications
- Frame alignment for trunk line termination
- Implementation of activation/deactivation procedures
- Switching of test loops
- Level detection in power-down state

The wiring configurations in user premises, in which the ISAC-S can be used, are illustrated in Figure 16.

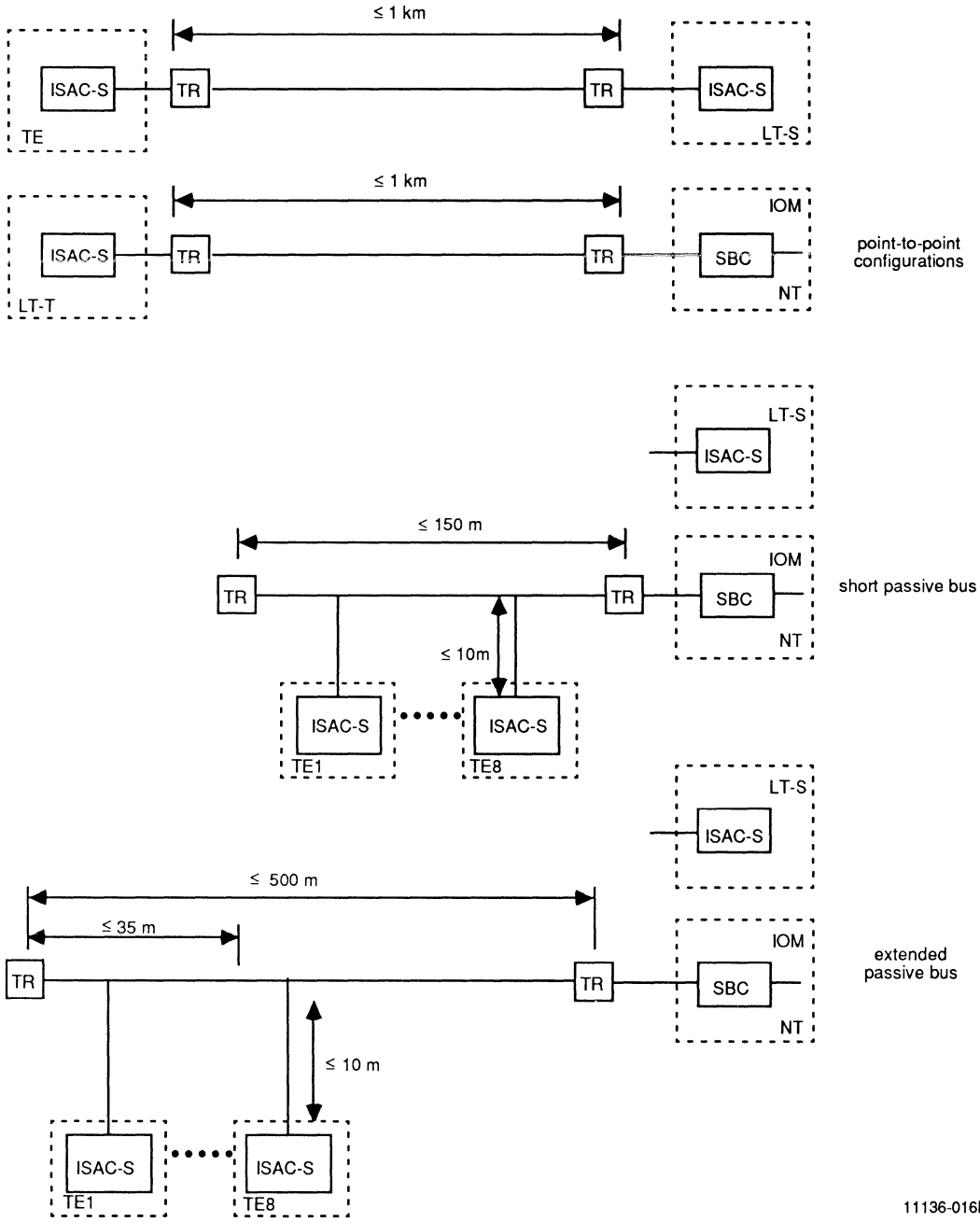


Figure 16. Wiring Configurations in User Premises

"S" Interface

According to CCITT recommendation I.430, a modified AMI code with 100% pulse width is used on the S interface. A logical 1 corresponds to a neutral level (no current), whereas logical 0s are coded as alternating positive and negative pulses. An example of a modified AMI code is shown in Figure 17.

One S-frame consists of 48 bits at a nominal bit rate of 192 kb/s. Thus each frame carries two octets of B1, two octets of B2, and four D-bits, according to the B1+B2+D structure defined for the ISDN basic access (total useful data rate: 144 kb/s). Frame beginning is marked using a code violation. The frame structures (from network to subscriber and subscriber to network) are shown in Figure 18.

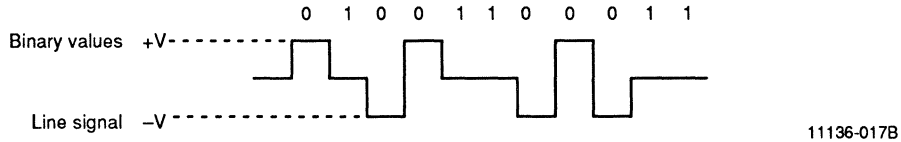


Figure 17. Modified AMI Code

ISDN APPLICATIONS

System Integration

The basic architecture for the ISDN basic access according to CCITT I series recommendations consists of

- an exchange and trunk line termination in the central office (ET, LT)
- a remote network termination in the user area (NT)
- a two-wire loop (U interface) between NT and LT
- a four-wire link (S interface) which connects subscriber terminals, PABX system and the NT in the user area as depicted in Figure 18.

The NT equipment simply serves as a link between the U interface on the exchange and the S interface on the user side. The NT itself may consist of either an NT1 only or an NT1 together with an NT2 connected via the T interface which is physically identical to the S interface. The NT1 is a direct transformation between Layer 1 of S and Layer 1 of U. NT2 also includes the complex functions like multiplex and exchange functions in higher layer OSI functions.

The ISAC-S is specifically designed for the user area of the ISDN basic access, especially Terminal and PABX exchange equipment. Figure 19 illustrates the general subscriber access architecture in the user area of the ISDN basic access including Terminal and PABX equipment.

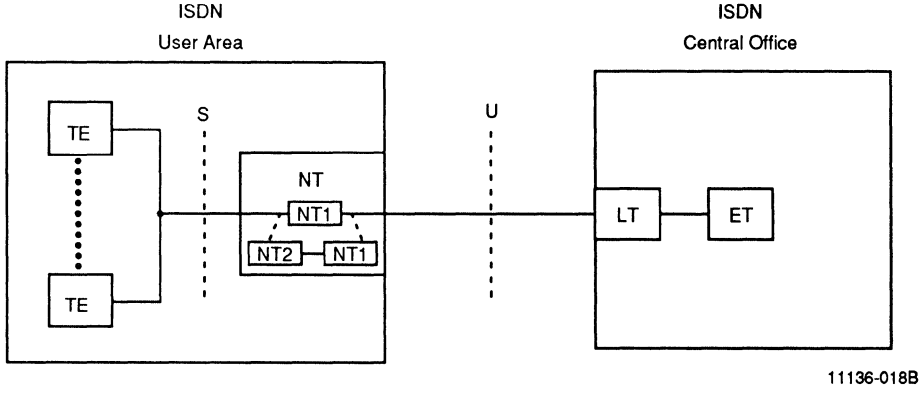


Figure 18. ISDN Architecture for the Basic Access

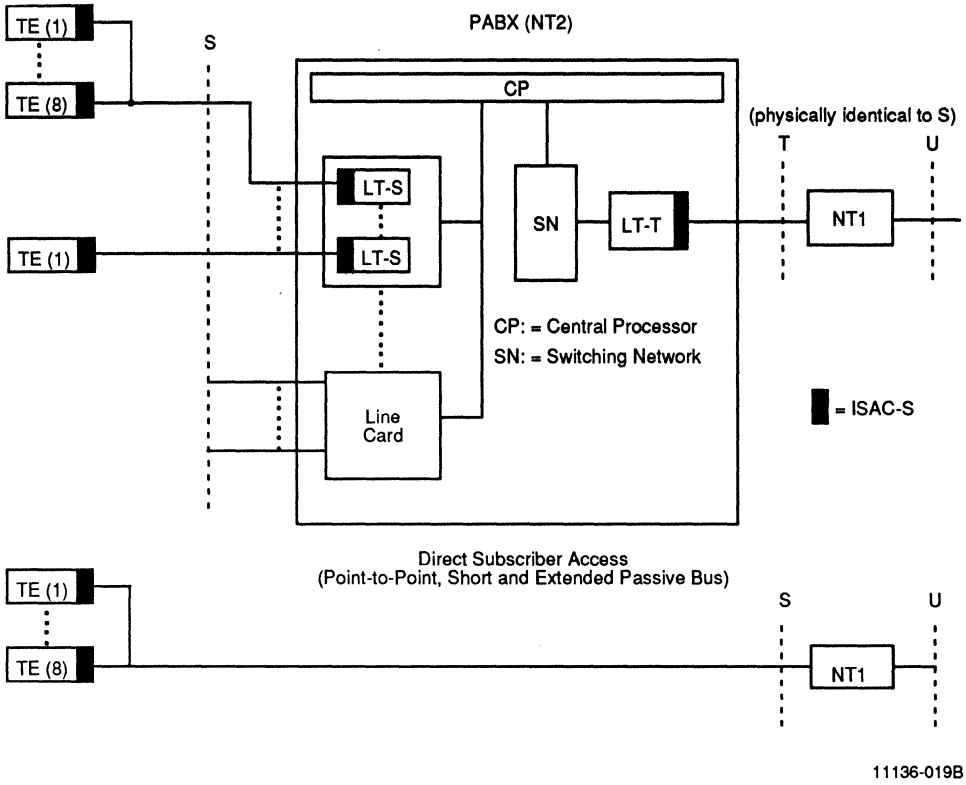


Figure 19. ISAC-S Applications in the User Area (ISDN Basic Access)

The concept of the ISDN basic access is based on two circuit-switched 64 kb/s B-channels and a message-oriented 16 kb/s D-channel for packetized data, signaling, and telemetry information.

The two serial interfaces of the ISAC-S, SLD, and SSI can be used as interfaces for B-channel sources/destinations, and the IOM interface in TIC-bus configuration provides the possibility of connecting further D-channel link entities to the S-bus.

Via the microprocessor interface, the microcomputer system can select the B-channel switching, can transmit/receive data packets in the D-channel, and has control over various functions (Layer 2, Layer 1, diagnostic, . . .).

To get a general idea, the following figures illustrate the ISAC-S integration into typical ISDN applications.

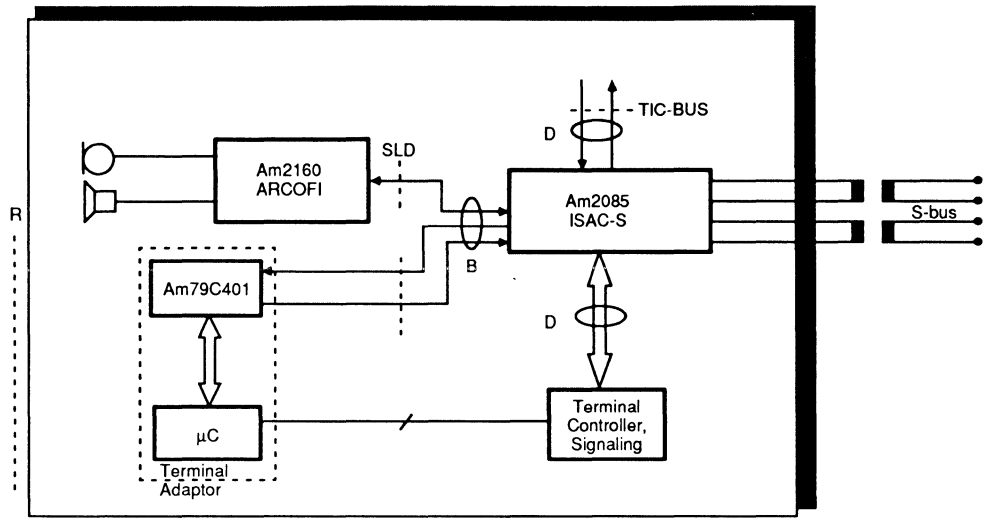
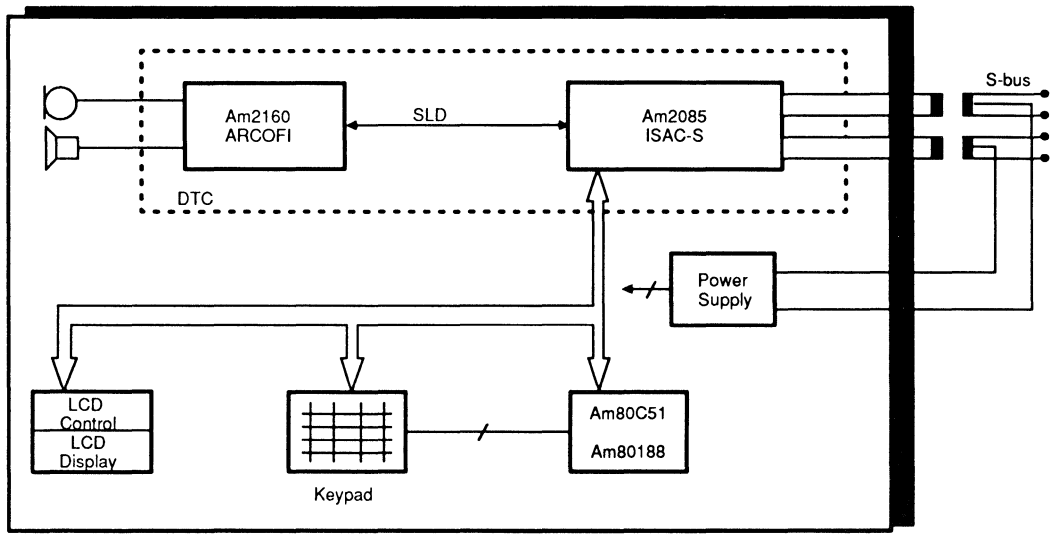


Figure 20. Extended ISDN Terminal (Voice/Data Workstation)

11136-020B



11136-021B

Figure 21. Basic ISDN Feature Telephone

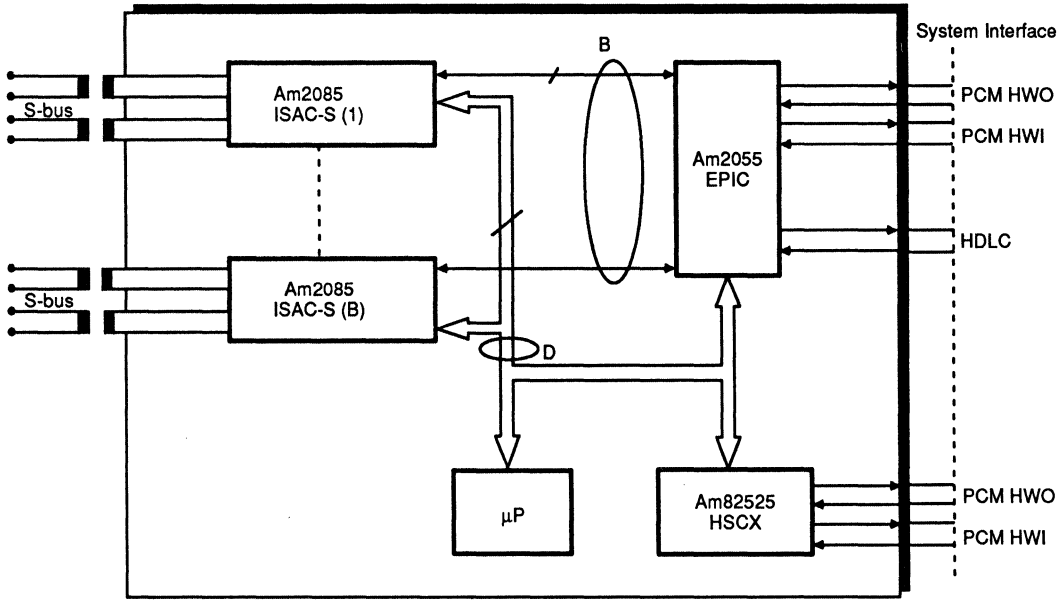


Figure 22. ISDN Line Card for PABX

Microprocessor Environment

The ISAC-S is especially suitable for cost-sensitive applications with single-chip microcontrollers (that is, Am8031, Am8051). Due to its bus structure (8-bit multiplexed address/data bus) and non-critical bus timing, it also fits perfectly into almost every 8-bit microprocessor system environment (that is, 8085, 8088, 80188).

With minimum hardware or software expense, it is also possible to use the ISAC-S with 16-bit microprocessors (that is, 8086, 80186). Figure 23 gives an example of the integration of ISAC-S in an 80C51 or 80188 system.

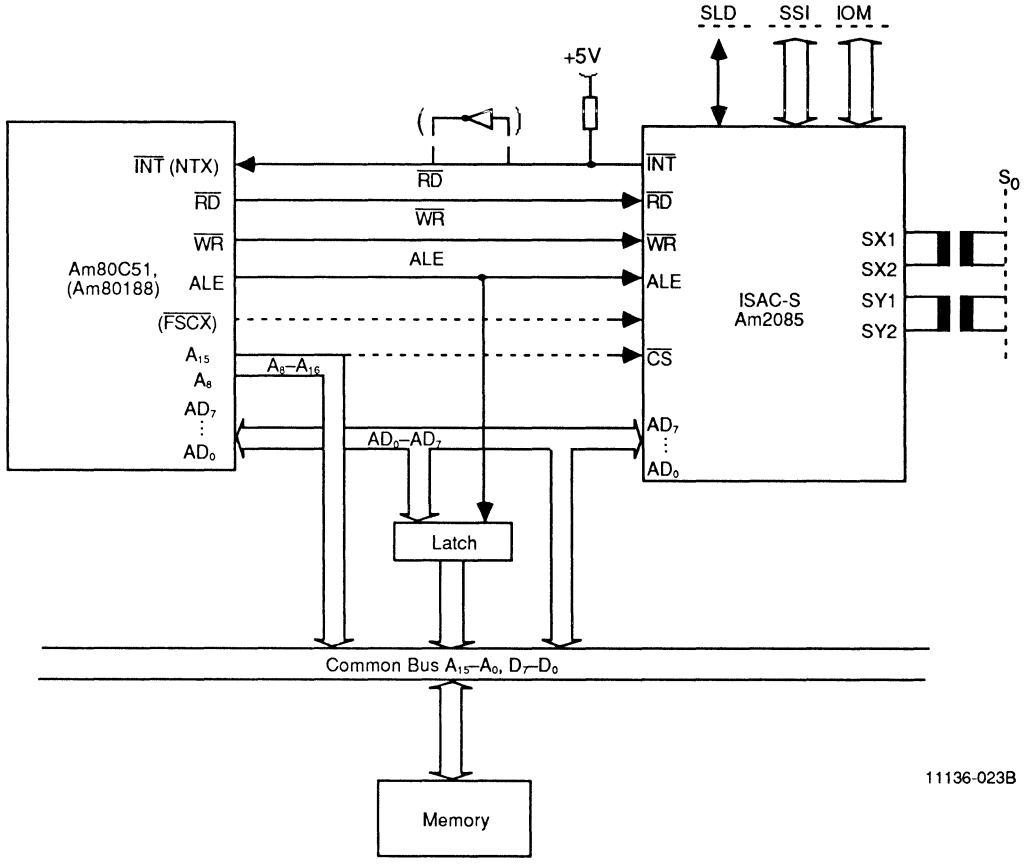
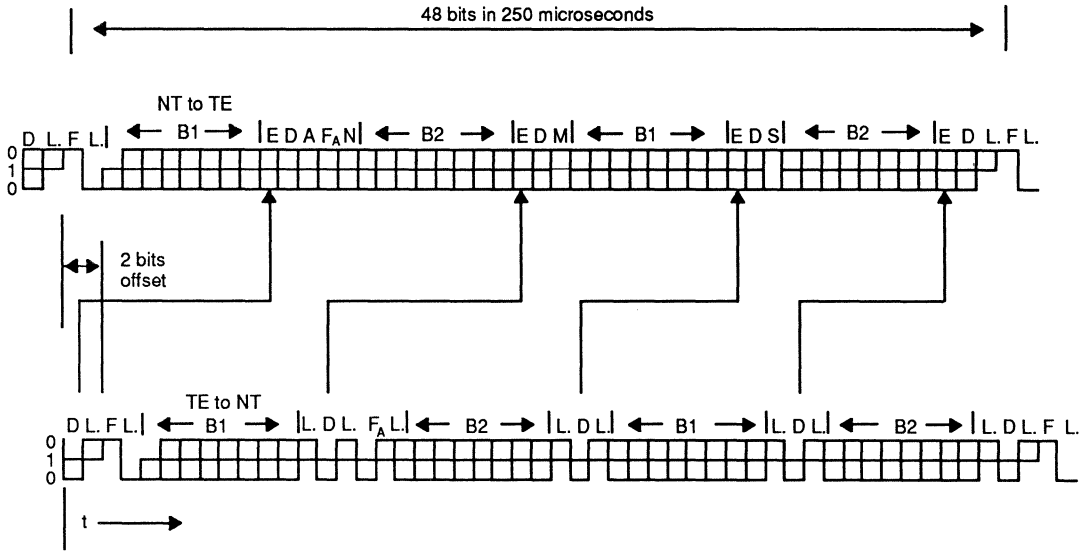


Figure 23. Example of Integration of ISAC-S in an Am80C51 or Am80188 System



A = bit used for activation
B1 = bit within B channel 1
B2 = bit within B channel 2
D = D-channel bit
E = D-echo-channel bit
F = framing bit
F_A = auxiliary framing bit or Q-bit
L = DC balancing bit
M = multiframing bit
N = bit set to a binary value N = F_A
S = reserved for future standardization

11136-024B

Note: Dots demarcate those parts of the frame that are independently DC-balanced.

Figure 24. Frame Structure at Reference Points S and T (CCITT I.430)

Analog Functions

The full-bauded AMI pulse shaping is achieved with the integrated transmitter which is realized as a voltage limited current source. A current of 7.5 mA is delivered over SX1-SX2, which yields a voltage of 1.5 V over 200 ohms.

The receiver is designed as a threshold detector with adaptively switched threshold levels. Pin SR1 delivers 2.5 V as an output, which is the virtual ground of the input signal on pin SR2.

An external transformer of ratio 2:1 is needed in both receive and transmit directions to provide for isolation and to transform voltage levels according to CCITT recommendations.

Level Detection Power Down (TE mode)

In power-down state, only an analog level detector is active. All clocks, including the IOM interface, are stopped. The data lines are High, whereas the clocks are Low.

An activation initiated from the exchange side (Info 2 on S-bus detected) will have the consequence that a clock signal is provided automatically. From the terminal side, an activation must be started by setting and resetting the SPU-bit in the SPCR register (see Detailed Register Description section).

Timing Recovery

A DPLL circuitry working with a frequency of 7.68 MHz ±100 ppm serves to generate the 192-kHz line clock from the reference clock delivered by the network and to extract the 192-kHz line clock from the receive data stream.

The 7.68-MHz clock may be generated with the help of an external crystal between pins XTAL1 and XTAL2. It may also be provided by an external oscillator.

The buffer memory serves to adapt the different bit rates of the S and the IOM interface. In addition, in trunk line applications it absorbs the possible deviation between two system clocks, according to CCITT Q.503.

Activation/Deactivation

An incorporated finite state machine controls ISDN Layer 1 activation/deactivation according to CCITT.

D-Channel Access

The D-channel access procedure according to CCITT I.430, including priority management, is fully implemented in the ISAC-S. When used in LT-S (NT) mode in a multipoint configuration, the device generates the echo bits necessary for D-channel collision detection.

Q-Channel Support

In terminal applications (TE), the Q-channel as specified by I.430 is supported. In case the ISAC-S in the terminal has received a binary one in F_A-bit position, it will reflect this binary one in the next S frame (also F_A-bit position) from TE to NT. This allows another terminal to use the extra transmission capacity.

Control of Layer 1

The control of the Layer 1 functions, especially the activation/deactivation procedure at the S interface, will be done by the exchange of special 4-bit Command/Indication codes in the C/I channel (see Operational Description section).

Layer 2 Functions for the ISDN Basic Access

LAPD, Layer 2 of the D-channel protocol (CCITT I.441) includes functions for :

- Provision of one or more data link connections on a D-channel (multiple LAP). Discrimination between the data link connections is performed by means of a data link connection identifier (DLCI = SAPI + TEI).
- HDLC-framing
- Application of a balanced class of procedure in point-multipoint configuration.

The simplified block diagram in Figure 25 shows the functional blocks of the ISAC-S which support the LAPD protocol.

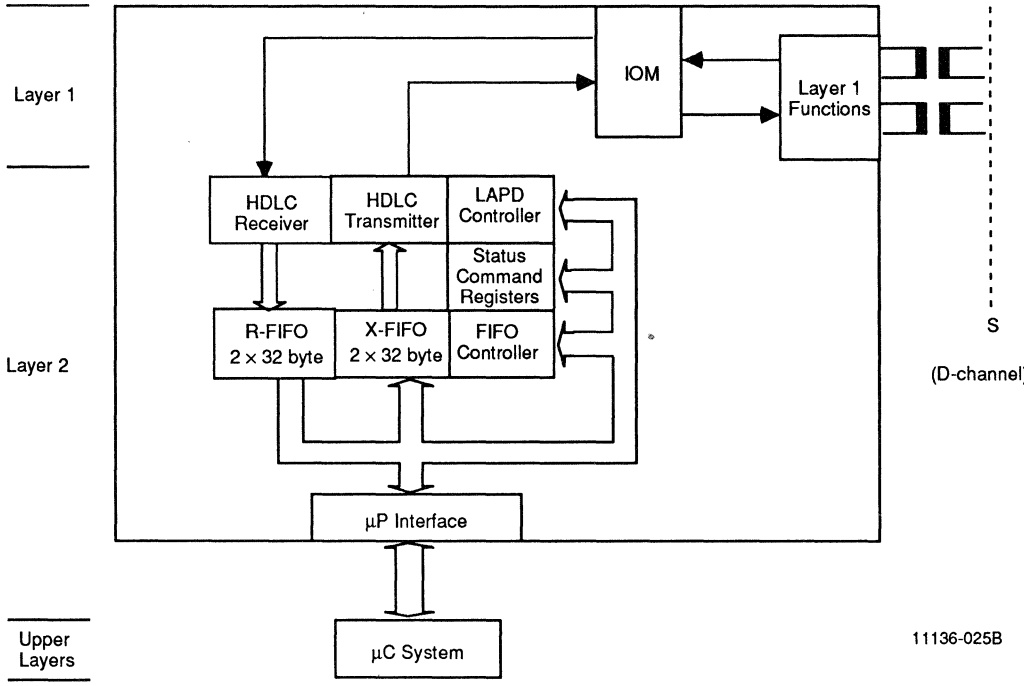


Figure 25. D-Channel Processing of the ISAC-S

For the support of LAPD, the ISAC-S contains an HDLC transceiver which is responsible for flag generation/recognition, bit stuffing, CRC-check and address recognition. In the auto mode, the LAPD controller handles the control field utilization and parts of LAPD procedures (information transfer in multiple frame operation with window size of 1).

A FIFO structure with two 64-byte pools for transmit and receive directions and an intelligent FIFO controller permits flexible transfer of protocol data units to and from the microcontroller system. Programming of the several modes and control of message transfer is done via status, command, and mode registers (see Operational Description and Detailed Register Description sections).

For the address recognition, the ISAC-S contains the registers SAP1, SAP2 for an individual SAPI address (fixed value for Group SAPI) and TEI1, TEI2 for an individual TEI address. The C/R-bit interpretation can be programmed according to network or user side in the SAP1 register (CRI-bit). The control field format for the optional modulo 128 operation can be selected in the SAP2 register (MCS-bit).

Message Transfer Modes

The ISAC-S supports Layer 2 of the D-channel protocol (LAPD) with different capabilities depending on the selected message transfer mode.

- **Auto mode**

  The ISAC-S processes all S- and I-frames of a logical link fully autonomously, according to CCITT I.441.

  During the "communication procedure," dialogue between the ISAC-S and processor is not necessary. The ISAC-S reports the status of the procedure to the processor. The Layer 2 software remaining in the microcontroller system is used for initialization and error recovery. As a prerequisite for this mode, window size 1 must be used between transmitted and acknowledged frames.

- **Non-auto mode**

  In this mode the control field and the information field of an HDLC frame is forwarded directly to the processor. The Layer 2 address recognition is still performed.

- **Transparent modes**

  The address field is either partly checked by the ISAC-S (SAPI) or completely forwarded to the processor. The Layer 2 headers are either stored in special purpose registers (transparent mode) or, together with the information field, in the FIFO buffer (extended transparent mode).

The three major types of message transfer mode and the corresponding Layer 2 functions in the ISAC-S are illustrated in Figure 26 in relation to the ISO's OSI reference model.

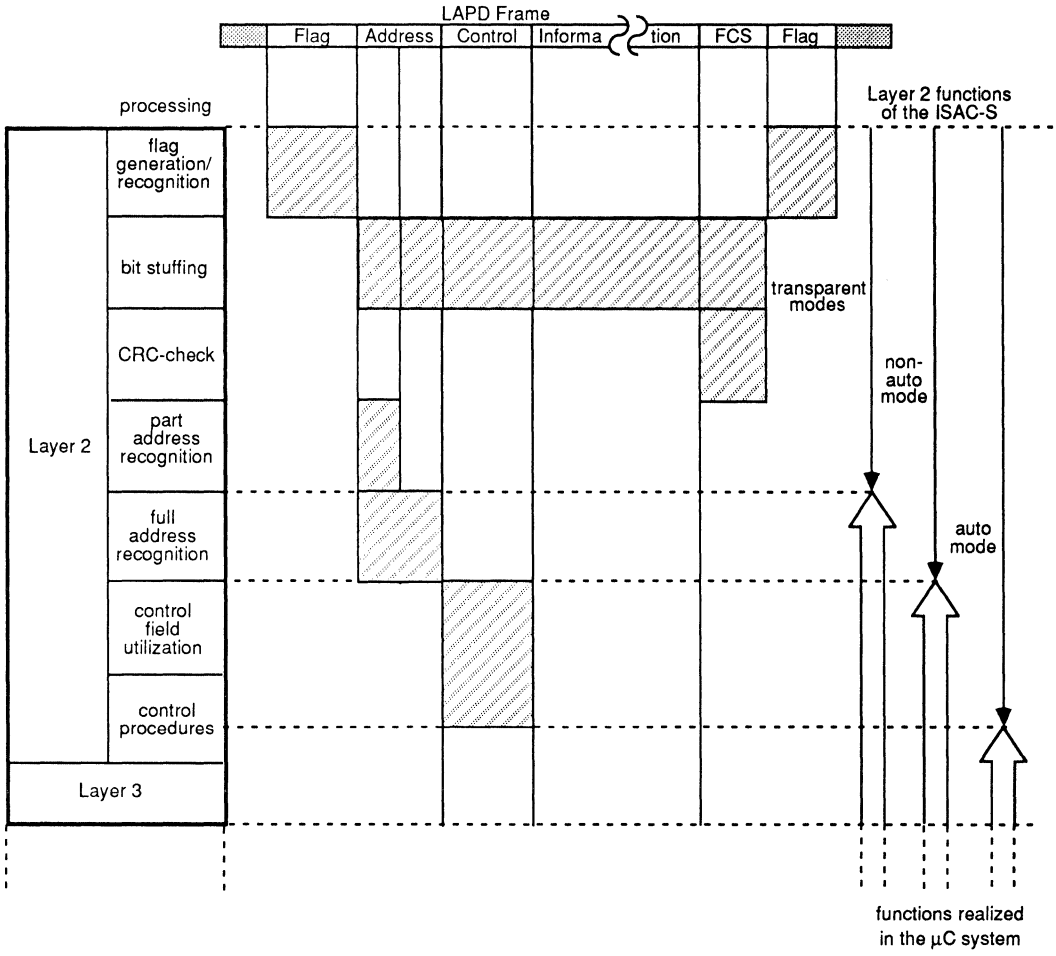


Figure 26. D-channel Protocol Support of the ISAC-S

11136-026B

Reception of Frames

Depending on the selected message transfer mode, the protocol data (address, control and information field) of a received frame will be stored in the RFIFO as well as in additional registers (RHCR, TEI1). The processor will be informed by an appropriate interrupt and must react within a corresponding reaction time.

The RFIFO and the FIFO-controller have been so designed that the maximum microprocessor reaction time is 16 ms for messages of more than 32 octets. The processor will be informed about the receiver operation (status) chiefly by means of the two interrupts:

- RPF (Receive Pool Full)
- RME (Receive Message End) —> ISTA (see Detailed Register Description)

When one of the two receive pools is filled up completely, the ISAC-S generates an RPF interrupt.

If the processor wants to save the frame, it should, as a consequence, react by reading the 32 bytes out of the RFIFO within the maximum reaction time.

During a handshake procedure between ISAC-S and microcontroller system, the processor has to acknowledge the reading by means of the RMC (Receive Message Complete) command.

When the end of a frame is detected, an RME interrupt will be generated, indicating that the remainder of the current frame is now available in one of the two receive pools. The reception of the last part of a frame must also be completed with an RMC command (see Operational Description).

With respect to the RME interrupt, the ISAC-S provides additional information about the received frame in its internal registers, according to Table 10.

Transmission of Frames

The processor initiates the transmission of a message with one of the two commands:

- XTF (Transmit Transparent Frame)
- XIF (Transmit I-frame, in auto mode only) —> CMDR

after it has written up to 32 bytes in one of the two 32-byte pools of the XFIFO.

When one pool is empty an XPR interrupt alerts the processor. The processor can then write further data to the XFIFO and enable the continuation of frame transmission according to a handshake procedure with the appropriate transmit command XIF of XTF (see also Interrupt List).

The microcontroller must indicate the message end with an XME command (together with the appropriate transmit command XIF or XTF) for the following cases:

- The message length is shorter than or equal to 32 bytes and all data is entered into the XFIFO or
- The last part of a longer message is written into the XFIFO

Table 10. Received Frame Information Saved in the Internal Registers

Information	Register	Bit	Mode
SAPI of LAPD address field	TEI1		Extended transparent mode 0
TEI of LAPD address field	TEI1		Extended transparent mode 1
			Transparent mode
	RHCR		Extended transparent mode 0
LAPD control field	RHCR		Auto mode
			Non-auto mode
			Transparent mode
			Extended transparent mode 1
Type of frame (Command/Response)	RSTA	C/R	—
Result of CRC-check (positive/negative)	RSTA	CRC	—
Data available in RFIFO (yes/no)	RSTA	RDA	—
Abort condition detected (yes/no)	RSTA	RAB	—
Data overflow during reception of a frame (yes/no)	RSTA	RD0	—
Recognition of Data Link Connection	RSTA	SA0	Auto mode
Identifier (DLCI = SAPI + TEI)		SA1	Non-auto mode
		TA	
Number of bytes received in RFIFO	RFBC	5–0	—
Message length (≤ 223)	RFBC	7–0	—

An XPR interrupt will also be generated subsequent to:

- The complete transmission of a transparent frame after the XTF and XME command or
- The reception of a positive acknowledge after the XIF and XME command or
- When no frame transmission is in progress and the microprocessor gives the XRES command

The message will be aborted automatically:

- When there is no more data in the XFIFO ready to be transmitted or
- When the processor gives an XRES command

The ISAC-S then transmits an abort sequence and generates an XDU interrupt.

Collisions that occur on the S-bus (D-channel) up to the 32nd data byte of a frame are treated without microprocessor interaction. The ISAC-S will retransmit the frame automatically. If the collision is detected later than the 32nd data byte of a frame, the ISAC-S aborts the frame and requests the processor to repeat the frame with an XMR (Transmit Message Repeat) interrupt.

In every mode, the start flag will be inserted automatically, as are likewise the end flag and the frame check sequence (CRC-16 according to LAPD) which are appended after an XME command.

When auto-mode I-frames are transmitted, the LAPD controller of the ISAC-S generates in addition to the delimiting flags and the FCS field, the address and control field autonomously. In this mode the XFIFO contains only the data for the information field.

The status of the XFIFO can also be read from the STAR register (XFW-bit: XFIFO write enable). This allows for a polling procedure instead of or in addition to the XPR interrupt and can be useful when there is no need to transmit a high quantity of messages (referred to the D-channel capacity).

Layer 2 Functions in the Auto Mode

In addition to address recognition, all S- and I-frames are processed independently by the ISAC-S in the auto mode with window size 1. The control field format may be either for basic (modulo 8) or extended (modulo 128) operation. The following functions are performed:

- Update of transmit and receive counter
- Evaluation of transmit and receive counter
- Processing of S commands
- Flow control with RR/RNR
- Response generation
- Recognition of protocol errors
- Transmitting of S commands, if an acknowledgment is not received
- Continuous status query of remote station after RNR has been received
- Programmable timer/repeater functions

Terminal-Specific Functions

In addition to the ISAC-S standard functions supporting the ISDN basic access, the ISAC-S contains optional functions, useful in various terminal configurations:

- Subscriber Awake (using SIP/SAW line)
- Watchdog Timer

The terminal-specific functions, STCR, SPCR, and CIXR registers (see Detailed Register Description), make it possible to generate a Reset signal as well as the appropriate interrupt in the following cases:

- Power Down
  - —Subscriber Awake—initiated by a falling edge on the SAW line (SAW interrupt)
  - —Exchange Awake—initiated by a message from Layer 1 (CIC interrupt)
- Power Up
  - —Watchdog Timer—after expiration of the internal watchdog timer (WOV interrupt)

The reset pulse generated by the ISAC-S has a pulse width of 5 ms and is an active High signal. During one time period of 128 ms, the WTC1- and WTC2-bit of the ADFR register must be set consecutively in the following manner:

Steps	WTC1	WTC2
1	1	0
2	0	1

As a result, the watchdog timer is reset and restarted.

Test Functions

The ISAC-S provides several test and diagnostic functions which can be grouped as follows:

- Closing loops in the transmission path; that is;
  - —loop internally inside the B-channel switching circuit (B-channel IOM loop) see B-Channel Switching)
  - —loop internally at the IOM interface, with reduced timer resolution (IOM interface loop), SPCR: TLP-bit, (see Detailed Register Description)
  - —loop at the analog end of the S interface (see Interrupt List)
- Using the IOM interface as an HDLC port without IOM frame structure (D-channel splitting) and no Layer 1, MODE and ADFR: TEM-bit, (see Detailed Register Description)
- Sending of special test signals on the S-bus, according to the modified AMI code; that is;
  - —single zeros (SSZ, 2 kHz repetition rate)
  - —continuous zeros (SCZ, 96 kHz repetition rate) (see Layer 1 Functions for the ISDN Basic Access; also see Processing)

DETAILED REGISTER DESCRIPTION

The parameterization of the ISAC-S as well as transfer of data and control information between the microprocessor and ISAC-S is performed with the R- and XFIFO and two register sets (Figure 27). The two FIFOs are accessed with the addresses 00—1FH which are of equal value and are referenced to the respective actual byte in the FIFO.

The special purpose registers of the address range 20-2FH pertain to the HDLC transceiver and LAPD controller as well as to higher-ranking functions in the ISAC-S.

The serial interfaces are controlled and monitored with the register record 30–3FH.

Register Address [1]	Read			Write	
	Name	Description	Name	Description	
00 ... 1F	RFIFO	Receive FIFO		Transmit FIFO	FIFO Buffer
20	ISTA	Interrupt Status Register	MASK	Mask Register	HDLC-Transceiver LADP-Controller, ...
21	STAR	Status Register	CMDR	Command Register	
22	MODE	MODE Register			
23	TIMR	Timer Register			
24	EXIR	Extended Interrupts	XAD1	Transmit Address 1	
25	RFBC	Receive Frame Byte Counter	XAD2	Transmit Address 2	
26	[2]		SAP1	SAPI Address 1	
27	RSTA Receive Status Register		SAP2	SAPI Address 2	
28	TEI1	TEI Address 1		[3]	
29	RHCR Receive HDLC Control		TEI2	TEI Address 1	
2A ... 2F	[4]				
30	SPCR	Serial Port Control			Serial Interface ...
31	CIRR	Command/Indicate Receive	CIXR	Command/Indicate Transmit	
32	MDNR	Monitor Register			
33	SSGR	SLD Signaling Receive	SSGX	SLD Signaling Transmit	
34	SFCR	SLD Feature Control			
35	BCX1	B1-Channel Transmit			
36	BCX2	B2-Channel Transmit			
37	BCR1	B1-Channel Receive	STCR	Synchr. Transfer Control	
38	BCR2	B2-Channel Receive	ADFR	Additional Features	
39 ... 3F					

Notes: 1) Hexadecimal representation of AD0_7
2) Inverted contents of SAP1 when reading
3) Inverted contents of TEI1 when reading auto
4) Invalid address range (data value "00" when reading)

11136-027B

Figure 27. Register Address Arrangement

For quick reference, the page numbers for all registers are listed below:

Internal Events and Conditions

LAPD/(HDLC) Operation Control

LAPD/(HDLC) Address Variables

LAPD/(HDLC) Control Data

LAPD/(HDLC) Information Data

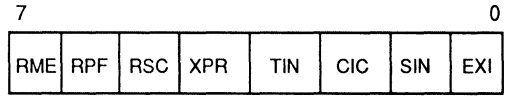
Serial Interface Control

Special Functions

ISTA—Interrupt Status Register—(Read)

Value after Reset: 00H
Address: 20H

Format:



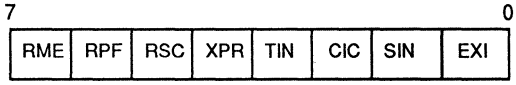
Bit Name	Description
RME	Receive Message End The receive message is now complete. Either one complete message, shorter than 32 bytes, or the remaining part of a longer message has been received. The contents are now available in the RFIFO. The actual message length and additional information may be obtained from the RFBC and the RSTA register.
RPF	Receive Pool Full A 32 byte block of a message, greater than 32 bytes has been received and is now available in the RFIFO. The message is not yet complete.
RSC	Receive Status Change (used in auto mode only) A status change in the receiver of the remote station (receiver ready/receiver not ready) has been detected, due to a received "RR"/"RNR" S-Frame (according to HDLC and LAPD). The current status can be read from the STAR register (RRNR-bit).
XPR	Transmit Pool Ready A data block of up to 32 bytes can be written to the XFIFO. An XPR interrupt will be generated in the following cases: after an XTF or XIF command, when one transmit pool is emptied and the frame is not yet complete after an XTF together with an XME command, when the whole transparent frame has been transmitted after an XIF together with an XME command, when the whole I-frame has been transmitted and a positive acknowledge from the remote station has been received.
TIN	Timer Interrupt The internal timer and repeat counter has expired (see also TIMR register).
CIC	C/I Code Change A change in C/I code has been recognized. The same C/I code, which differs from the prior subsequent received C/I code, has been received at least twice. The actual C/I code can be read from the CIRR register.
SIN	Synchronous Transfer Register The beginning or the center of an IOM-frame has been indicated, as programmed in the STCR register.
EXI	Extended Interrupt This interrupt signifies that one of eight non-critical interrupts, indicated in the extended interrupt register EXIR, has been generated. The exact cause must be read from EXIR.

Note: Reading of the EXIR register by the processor clears the EXI-bit, and reading of the ISTA register clears all other bits of ISTA.

MASK—Mask Register—(Write)

Value after Reset: 00H (all interrupts enabled)
Address: 20H

Format:



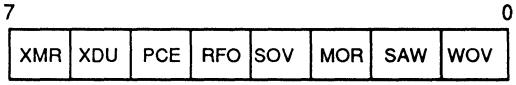
Bit Name	Description
RME	Each bit of the MASK register relates to the corresponding bits of the ISTA register. Each interrupt can be selectively masked by setting the respective bit in MASK.
EXI	Masked interrupts are not indicated when reading ISTA. Instead, they remain internally stored and will be indicated after the respective MASK is reset.

Note: In the event of an extended interrupt, no interrupt request (INT = Low) will be generated with a masked EXI-bit, although this bit is set in ISTA.

EXIR—Extended Interrupt Register—(Read)

Value after Reset: 00H
Address: 24H

Format:



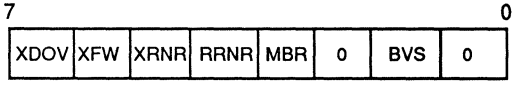
Bit Name	Description
XMR	Transmit Message Repeat The transmission of the last message has to be repeated because: the ISAC-S has received a negative acknowledgment in auto mode (according to HDLC/LAPD) a collision on the S-bus has been detected after the 32nd data byte
XDU	Transmit Data Underrun The actual transmission of a frame has been aborted with "IDLE" because the XFIFO holds no further data. This interrupt occurs whenever the processor has failed to respond to an XPR interrupt (ISTA register) quickly enough, after having initiated a frame transmission, and the message to be transmitted is not yet complete.
PCE	Protocol Error (significant during auto mode only) A protocol error has been detected during auto mode due to a received S- or I-frame with an incorrect sequence number N(R), or to an S-frame containing an I-field.
RFO	Receive Frame Overflow The received data of a message could not be stored entirely, because the internal message buffer is occupied (the whole message has been lost). This interrupt can be used for statistical purposes and indicates that the processor does not respond quickly enough to an incoming RPF or RME interrupt (ISTA).
SOV	Synchronous Transfer Overflow The synchronous transfer programmed in STCR (ST0/ST1) was not confirmed in time (setting SC0/SC1 by the processor).
MOR	Monitor Byte Received (not used in the ISAC-S) A valid monitor byte has been identified in an IOM frame and has been stored in the MONR register.
SAW	Subscriber Awake Indicates that a falling edge on the SAW line has been detected, in case the terminal-specific functions are adjusted (TFS-bit in STCR).
WOV	Watchdog Timer Overflow Signals the expiration of the watchdog timer, which means that the processor has failed to set the watchdog timer control bits WTC1 and WTC2 in the correct manner. A reset pulse has been generated by the ISAC-S.

Note: When an XMR or XDU interrupt is generated, it is not possible to send transparent frames or I-frames until the interrupt has been acknowledged by the processor.

STAR—Status Register—(Read)

Value after Reset: 48H
Address: 21H

Format:



Bit Name	Description
XDOV	Transmit Data Overflow More than 32 bytes have been written to one pool of the XFIFO, and data have been overwritten in this pool.
XFW	Transmit FIFO Write Enable Data can be written to the XFIFO. The polling of the XFW-bit can be used instead of or in addition to an XPR interrupt handling when few messages have to be transmitted (refer to D-channel capacity).
XRNR	Transmit RNR (significant during auto mode only) During auto mode, this bit indicates whether the ISAC-S receiver status is ready or not ready (if not ready, the ISAC-S sends an "RNR" S-frame autonomously to the remote station.) 0 —> receiver ready 1 —> receiver not ready
RRNR	Received RNR (significant in auto mode only) During auto mode, this bit indicates the receiver status of the remote station 0 —> (remote) receiver ready 1 —> (remote) receiver not ready (If not ready, the ISAC-S has received an "RNR" S-frame from the remote station.)
MBR	Message Buffer Ready This bit signifies that temporary storage is available in the RFIFO to receive at least the first 16 bytes of a new message.
BVS	B-channel Valid at SLD port BVS indicates the beginning of an SLD frame, especially the validity of the two B-channels, and represents a 16-kHz signal, synchronous to the SLD frame.

LAPD/(HDLC) Operation Control

CMDR—Command Register—(Write)

Value after Reset: 00H
Address: 21H

Format:

7							0
RMC	RHR	RNR	STI	XTF	XIF	XME	XRES

Bit Name	Description
RMC	Receive Message Complete Reaction to RPF (Receive Pool Full) or RME (Receive Message End) interrupt. By setting this bit, the processor confirms to the ISAC-S, that it has fetched the current frame or data block, and the ISAC-S can release the space occupied in the RFIFO.
RHR	Reset HDLC Receiver All data in the RFIFO and the contents of the message buffer is deleted. In auto mode, the Transmit and Receive counters (V)[S], V[R] according to HDLC), also are reset.
RNR	Receiver Not Ready (used in auto mode only) The status of the ISAC-S receiver is set. Determines, whether a received frame is acknowledged via an "RR" or "RNR" supervisory frame (S-frame according to HDLC) in auto mode. 0 —> receiver ready ("RR") 1 —> receiver not ready ("RNR")
STI	Start Timer The internal timer is started in external timer mode (TMD-bit in MODE register). The timer is stopped by rewriting the TIMR register after start.
XTF	Transmit Transparent Frame After having written up to 32 bytes in the XFIFO, this command bit initiates the transmission of a transparent frame. An opening flag sequence is automatically added to the data by the ISAC-S.
XIF	Transmit I-Frame (used in auto mode only) Initiates the transmission of an I-frame in auto mode. In addition to the opening flag sequence, the address and control field of the frame is automatically added by the ISAC-S.
XME	Transmit Message End Indicates that the data block written last to the XFIFO completes the current frame. The ISAC-S terminates the transmission operation properly by appending the CRC and the closing flag sequence to the data.
XRES	Transmit Reset The contents of the XFIFO is deleted and an "IDLE" is transmitted. This command can be used by the processor to abort a frame currently in transmission.

Note: After an XPR interrupt, further data have to be written in the XFIFO and the appropriate Transmit Command (XTF or XIF) has to be written in the CMDR register again to continue transmission, when the current frame is not yet complete (see also XPR in ISTA).

During frame transmission, the 0-bit insertion according to the HDLC bit-stuffing mechanism is done automatically.

MODE—Mode Register—(Read/Write)

Value after Reset: 00H
Address: 22H

Format:

7							0
MDS1	MDS0	ADM	TMD	RAC	HMD2	HMD1	HMD0

Bit Name — **Description**

MDS1, MDS0 — Mode Select
The message transfer mode of the HDLC controller is selected.

MDS1	MDS0	Mode
0	0	auto
0	1	non-auto
1	0	transparent
1	1	extended transparent

ADM — Address Mode
Defines the length of the address field in an HDLC frame.
0 —> 1-byte address field, LAPB
1 —> 2-byte address field, LAPD
If the extended transparent mode is selected by setting MDS1 = MDS0 = 1, this bit differentiates between the two modes:
0 —> extended transparent mode 0 (no address recognition)
1 —> extended transparent mode 1 (SAPI address recognition)

TMD — Timer Mode
The operating mode of the internal timer is set.
0 —> external mode
The Timer is controlled by the processor and can be started at any time by setting the STI-bit in CMDR and can be stopped by rewriting the TIMR register (see also TIMR register).
1 —> internal mode
The timer is used internally by the ISAC-S for time-out and retry conditions in auto mode (refer to TIMR register description).

RAC — Receiver Active
Switches the receiver to operational or inoperational state.
0 —> receiver inactive
1 —> receiver active

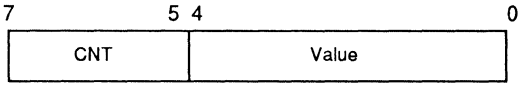
HMD2, HMD1, HMD0 — HDLC Port Mode
The operating mode of the IOM/(HDLC) interface is set.

HMD2	HMD1	HMD0	Mode	Interface Monitor Channel
0	0	0	IOM	Monitor channel is not used. In point-to-point or for primary use (LT-S/NT mode) in point-multipoint configurations (S-bus).
0	0	1		Monitor channel is used. For secondary use (TE mode) in point-multipoint configurations (D-channel access control).
0	1	0	/	Reserved
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1	HDLC	For diagnostic or test configurations, it is possible to program a pure HDLC frame without the complex 2-D-bit splitting of the IOM frame. The data clock and the component clock are identical.

TIMR—Timer Register—(Read/Write)

Value after Reset: Undefined (Previous Value)
Address: 23H

Format:



Field Name	Description

CNT is a 3-bit field; value is a 5-bit field.

Value — Sets the time period T1 as follows:
T1 = (Value + 1) • 64 µs

CNT — Interpreted differently, depending on the selected timer mode (TMD-bit in the MODE register).
Internal timer mode (TMD = 1)

CNT indicates the maximum number of S-commands "N1" which are transmitted autonomously by the ISAC-S after expiration of time period T1 (retry, according to HDLC). The internal timer procedure will be started in auto mode:

after start of I-frame transmission; or,
after an "RNR" S-frame has been received

After the last retry, a timer interrupt (TIN-bit in ISTA) is generated.
The maximum time between the start of I-frame transmission or reception of an "RNR" S-frame and the generation of a TIN interrupt would be: (CNT + 1 • T1.

The timer procedure will be stopped when:

a TIN interrupt is generated; or,
the TIMR is written to; or,
a positive or negative acknowledgment has been received.

Note: The maximum value of CNT can be 6. If CNT is set to 7, the number of retries is unlimited.
External timer mode (TMD = 0)
CNT together with Value determine the time period T2 after which a TIN interrupt will be generated:
T2 = CNT • 2.048 sec + T1
The timer can be started by setting the STI-bit in CMDR and will be stopped when :

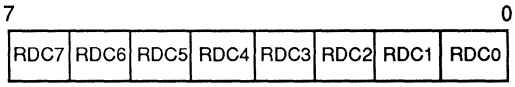
a TIN interrupt is generated; or,
the TIMR register is written to

If CNT is set to 7, a TIN interrupt is periodically generated after every expiration of T1.

RFBC—Receive Frame Byte Counter—(Read)

Value after Reset: 00H
Address: 25H

Format:



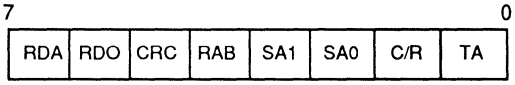
Bit Name	Description
RDC7–0	Receive Data Count Represents the total number of actual received data bytes of a message (limited to 223).
RCRDC4–0	Indicates always the length of the data block currently available in the 32-byte RFIFO. For message lengths greater than 223 data bytes, the bits RDC7–5 remain to the value "111." In this case only the bits RDC4–0 are significant.

Note: Normally this register should be read by the processor after an RME interrupt in order to determine the number of bytes to be read from the RFIFO, and the total message length.

RSTA—Receive Status Register—(Read)

Value after Reset: Undefined
Address: 27H

Format:



Bit Name	Description
RDA	Data Received Data is available in the RFIFO. (RDA = "1"). After an RME interrupt, an RDA = "0" means that data is available in the internal registers RHCR or TEI1 only (that is, S-frame).
RDO	Receive Data Overflow A data overflow has occurred with the current frame. At least one byte of the frame has been lost (that is, the last message has to be deleted).
CRC	CRC compare/check 0 —> CRC check failed (received frame contains errors) 1 —> CRC check passed (received frame contains no errors) Receive Message Aborted The received message was aborted from the remote station. According to HDLC, this frame must be discarded by the processor.
SA1, SA0	SAPI Address Identification Indicates which of the three possible SAPI addresses were recognized.

SA1	SA0		
0	0	SAPI2	(programmable)
0	1	Group SAPI	(fixed value, ISDN/LAPD management functions)
1	0	SAPI1	(programmable)

C/R — Command/Response
The C/R bit identifies a frame as either a command or a response, according to LAPD.

Command	Response	Direction
0	1	user to network side
1	0	network to user side

TA — TEI Address Identification
The comparison of the TEI address in the received frame with the two programmable addresses TEI1 and TEI2 resulted in:

1 —> TEI1 was recognized
0 —> TEI2 was recognized

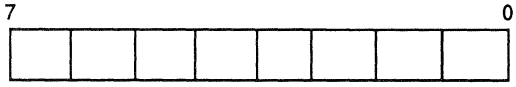
Note: In transparent and extended transparent mode with ADM-bit set to 1 in the MODE register:
SAPI address recognition—the TA-bit is irrelevant.
In extended transparent mode with ADM-bit set to 0
No address recognition—bits 0–3 are irrelevant.

If the programmable address registers SAP1 and SAP2 contain the same address value, the bit combination "00" will be omitted.

XAD1—Transmit Address 1—(Write)

Address: 24H

Format:



Bit Name	Description
Bit 0–7	Used in auto mode only XAD1 contains an individual programmable address byte which is appended automatically to the frame by the ISAC-S in auto mode. Depending on the selected address mode (ADM-bit in MODE,) XAD1 is interpreted as follows: 2-byte address field (AMD = 0) XAD1 builds up the high byte (SAPI in the ISDN) of the 2-byte address field. According to the ISDN LAPD protocol, bit 1 is interpreted as the command/response bit "C/R" and will be automatically inserted by the ISAC-S depending on the CRI bit in the SAP1 register. Bit 1 must be set to 0. In the ISDN LAPD, the address field extension bit "EA," that is, bit 0 of XAD1 must be set to 0.

C/R Bit			
Command	Response	Transmission Side	CRI Bit
0	1	Network	0
1	0	User	1

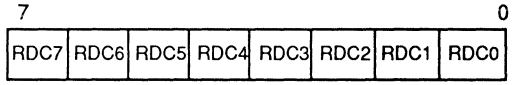
1-byte address field (ADM = 1)
According to the X.25 LAPB protocol, XAD1 indicates a command.

Note: In standard ISDN/LAPD applications, only 2-byte address fields are used.

XAD2—Transmit Address 2—(Write)

Address: 25H

Format:



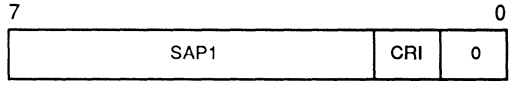
Bit Name	Description
Bit 0–7	Used in auto mode only XAD2 represents the second individual programmable address byte, whose function depends on the selected address mode (ADM-bit in MODE) 2-byte address (ADM = 0) XAD2 builds up the low byte (TEI in the ISDN) of the 2-byte address field. 1-byte address (ADM = 1) According to the X.25 LAPB protocol, XAD2 indicates a response.

Note: See note to XAD1 register description.

SAP1—SAPI Register 1—(Write)

Address: 26H

Format:



Bit Name	Description
SAPI1 Bit 7–2	SAPI 1 value Value of the first individual programmable Service Access Point Identifier (SAPI) according to the ISDN LAPD protocol.
CRI	Command/Response Interpretation CRI defines the side of the ISDN user-network interface for identification of command/response frames. Depending on CR,I the C/R-bit will be interpreted autonomously by the ISAC-S, when receiving frames in auto mode, as follows:

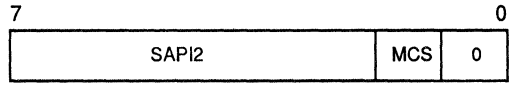
		C/R Bit	
CRI Bit	Receiver End	Command	Response
0	Network	1	0
1	User	0	1

For transmitting frames in auto mode, the C/R-bit manipulation will also be done automatically, depending on the setting of the CRI-bit (refer to XAD1 register description). In message transfer modes with SAPI address recognition (all except extended transparent mode 0) the high byte of the received address is compared with the individual programmable values in SAP1, SAP2, and the fixed group SAPI. (In 1-byte address mode, the CRI-bit would have to be set to 0.)

SAP2—SAPI Register 2—(Write)

Address: 27H

Format:



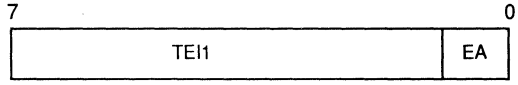
Bit Name	Description
SAPI2 Bit 7–2	SAPI 2 value Value of the second individual programmable Service Access Point Identifier (SAPI) according to the ISDN LAPD protocol.
MCS	Module Count Select (valid in auto mode only) The MCS bit adjusts the control field format according to the ISDN LAPD protocol. 0 —> basic operation (modulo 8) 1 —> extended operation (modulo 128)

Note: When modulo 128 is selected in auto mode, the RHCR register contains compressed information of the extended control field (see RHCR register description).

TEI1—TEI Register 1—(Read/Write)

Address: 28H

Format:



Bit Name	Description
TEI(1) Bit 7–1	TEI value 1
EA	Address field Extension bit Has to be set to "1" according to ISDN LAPD. Write: Value of the first individual programmable Terminal Endpoint Identifier (TEI) according to the ISDN LAPD protocol (2-byte address field). In auto and non-auto mode this value is used by the ISAC-S for the address recognition. Read: Depending on the message transfer mode, the reading of the TEI1 register contains the following information:

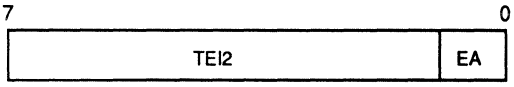
Message Transfer Mode	Meaning/Contents
Auto	Inverted value of the programmed TEI1
Non-auto	Inverted value of the programmed TEI1
Transparent	Received TEI value
Extended Transparent 0	First byte of the frame after the opening flag
Extended Transparent 1	Received TEI value

Note: In auto and non-auto mode with 1-byte address field, the whole contents of TEI1 would be recognized as a command according to X.25 LAPB.

TEI2—TEI Register 2—(Write)

Address: 29H

Format:



Bit Name	Description
TEI(2) Bit 7–2	TEI value 2 Value of the second individual programmable Terminal Endpoint Identifier (TEI) according to the ISDN LAPD protocol (2-byte address field). In auto and non-auto mode, this value is used by the ISAC-S for the address recognition.
EA	Address field Extension bit Has to be set to "1" according to ISDN LAPD.

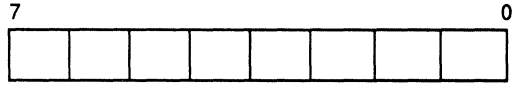
Note: In auto and non-auto mode with 1-byte address field, the whole contents of TEI2 would be interpreted as a response according to X.25 LAPB.

LAPD/(HDLC) Control Data

RHCR—Receive HDLC Control Register—(Read)

Address: 29H

Format:



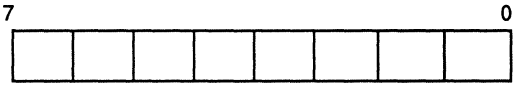
Bit Name	Description
Bit 0–7	Value of the received HDLC control field In extended transparent mode 0 (no address recognition), RHCR contains the second byte of a received frame after the opening flag. When modulo 128 is selected in auto mode, the RHCR register contains compressed informa tion of the extended control field. In this case the bit 0 of the RHCR register has the following meaning: 0 —> an I-frame has been received 1 —> a U-frame has been received (S-frames will be handled autonomously by the ISAC-S.) When message transfer modes other than the auto mode are used and "modulo 128" is agreed upon, then the first octet of the extended control field is available in the RHCR register. The second octet is always available in the RFIFO corresponding to the message transfer mode.

LAPD/(HDLC) Information Data

RFIFO—Receive FIFO—(Read)

Address: 00–1FH

Format:

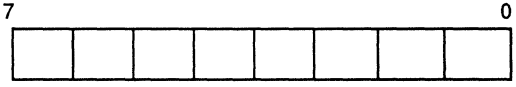


Bit Name	Description
Bit 0–7	Up to 32 bytes of received data can be read from the RFIFO. After an RPF interrupt (ISTA), exactly 32 bytes are available. After an RME interrupt (ISTA), the number of data bytes to be read may be obtained reading the RFBC register.

XFIFO—Transmit FIFO—(Write)

Address: 00–1FH

Format:



Bit Name	Description
Bit 0–7	Up to 32 bytes of transmit data can be written into the XFIFO following an XPR interrupt (ISTA).

Note: Addresses within the address space of the FIFOs (00–1FH) are interpreted equally, that is, the current data byte can be accessed with any address within the valid space.

Serial Interface Control

SPCR—Serial Port Control Register—(Read/Write)

Address: 30H

Format:

7							0
SPU	SAC	SPM	TLP	B1C1	B1C0	B2C1	B2C0

Bit Name	Description
SPU	Software Power Up (in TE mode only) For activating the ISDN S interface in TE mode, the SPU-bit has to be set to '1' and then cleared again. After a subsequent CIC interrupt (C/I code change; ISTA) and reception (the C/I code "PU" Power Up indication in TE mode), the reaction of the processor would be: to write an Activate Request command as C/I code in the CIXR register to reset the SPU-bit and wait for the following CIC interrupt.
SAC	SIP Activated With SAC, the state of the SLD port can be programmed as follows: 0 —> inactive in transmit direction (SIP: always high impedance) 1 —> active in transmit direction
SPM	Serial Port Timing Mode SPM selects the timing mode: 0—> timing mode 0 — terminal: Timing signals are derived from the S-bus. DCL, FSC1 and FSC2 are outputs. FSD/SCA is a 128-kHz clock signal for SSI. 1 —> timing mode 1 — exchange: Timing signals are provided by the digital exchange system. DCL, FSC1 and FSC2 are inputs. FSD/SCA is the delayed frame signal of FSC1.
TPL	Test Loop Instructs the ISAC-S to connect internally the IOM interface lines SDI and SDO. Also, the times T1 and T2 programmed in the TIMR register—are reduced by a factor of 64.
B1C1, B1C0	Switching of B1 channel — Both B-channels can be switched independently of each other.
B2C1, B2C0	Switching of B2 channel

BC1	BC0	B-Channel Switching	µP Access
0	0	SLD loop	monitoring, loop
0	1	SLD-IOM (S-bus) connection	monitoring
1	0	SSI-IOM (S-bus) connection	monitoring
1	1	IOM loop	monitoring, loop

1

STCR—Synchron Transfer Control Register—(Write)

Address: 37H

Format:

7					0
TSF	T-ADR	ST1	ST0	SC1	SC0

Bit Name	Description
TSF	Terminal Specific Functions 0 —> The serial interface port for the SLD interface is selected (SIP). 1 —> The terminal-specific, such as Subscriber/Exchange Awake (SIP/SAW)functions, are activated, Watchdog Timer In this case, the SIP/SAW line is always an input signal which can serve as a request signal from the subscriber to initiate the awake function in a terminal. A falling edge on the SAW line generates an SAW interrupt (EXIR). When the RSS-bit in the CIXR register is zero, a falling edge on the SAW line (Subscriber Awake) or a C/I code change (Exchange Awake) initiates a reset pulse. When the RSS-bit is set to one on the other hand, a reset pulse is triggered only by the expiration of the watchdog timer (see also CIXR register).
T-ADR	TIC bus Address T-ADR defines an individual address for the point-to-multipoint bus configuration of the IO -interface (TIC bus: Telecom IC bus). The bus configuration makes it possible to connect further ICCs (up to seven) to the Layer 1 component of the ISAC-S.
ST1	Synchronous Transfer 1 When set, causes the ISAC-S to generate an SIN interrupt (ISTA register) at the beginning of the 8-kHz frame signal. timing mode 0 FSC1 and FSC2 (IOM, SLD, SSI) timing mode 1 FSC1 (System)
ST0	Synchronous Transfer 0 When set, causes the ISAC-S to generate an SIN interrupt at the center of the 8-kHz frame signal. timing mode 0 FSC1 and FSC2 (IOM, SLD, SSI) timing mode 1 FSC1 (System)
SC1	Synchronous Transfer 1 Completed After an SIN interrupt, the processor has to access and subsequently acknowledge the interrupt by setting the SC1-bit before the center of the frame signal is reached, if the interrupt originated from a Synchronous Transfer 1 (ST1). Otherwise, an SOV interrupt (EXIR register) will be generated.
SC0	Synchronous Transfer 0 Completed After an SIN interrupt, the processor has to access and subsequently acknowledge the interrupt by setting the SC0-bit before the end of the frame signal is reached, if the interrupt originated from a Synchronous Transfer 0 (ST0). Otherwise, an SOV interrupt (EXIR register) will be generated.

Notes: ST0/1 and SC0/1 are useful for synchronizing microprocessor accesses and receive/transmit operations in the ISAC-S.

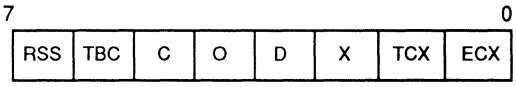
In a TIC bus configuration, that is, if additional Layer-2 controllers (ICCs) are connected to the IOM interface, it has to be ensured that one Layer-2 component has been assigned the TIC bus address "7." For applications without additional Layer-2 components it is generally recommended to set ADR to "7" (after reset T-ADR = "0").

The TSF-bit will be cleared only by Hardware reset.

CIXR—Control/Indicate Transmit Register—(Write)

Value after Reset: BFH
Address: 31H

Format:

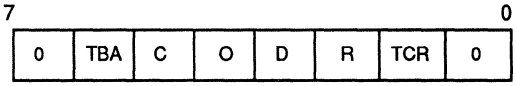


Bit Name	Description
RSS	Reset Source Select 0 —> Subscriber or Exchange Awake (Power Down) As reset source serves: a falling edge on the SAW line (Subscriber Awake) a C/I code change (Exchange Awake) A logical zero on the SAW line also activates the IOM interface clock and frame signal, just as the SPU-bit (SPCR) does. 1 —> Watchdog Timer (Power Up) The expiration of the watchdog timer generates a reset pulse. The watchdog timer will be reset and restarted when two special bit combinations are written in the ADFR register within the time period of 128 ms (see also ADFR register description). After a reset pulse generated by the ISAC-S and the appropriate interrupt (WOV, SAW or CIC), the actual reset source can be read from the ISTA and EXIR.
TBC	TIC bus Control For transmitting the C/I code written in the CIXR register, an access to the TIC bus can be forced by setting the TBC-bit. When no further Layer 2 controllers are connected to the TIC bus (IOM interface) and the TIC bus address is programmed to "7" (T-ADR in the STCR register,) the TIC bus is always accessed and it is not necessary to set the TBC-bit.
CODX	C/I-Code Transmit These four bits are transmitted continuously (beginning with the left-most one) as a Command/Indicate code in the C/I channel of the IOM frame, if the TIC bus is accessed.
TCX	T-Channel Transmit The TCX-bit will be transmitted continuously in the T-channel of the IOM frame if the TIC bus is accessed.
ECX	E-Channel Transmit The ECX-bit will be transmitted continuously in the E-channel of the IOM frame if the TIC bus is accessed.

CIRR—Control/Indicate Receive Register—(Read)

Value after Reset: 7EH
Address: 31H

Format:



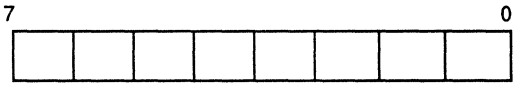
Bit Name	Description
TBA	TIC Bus Access Indicates the state of the TIC bus (transmit direction: Layer 2 to Layer 1; SDI) when a C/I code change has been recognized. TBA = 0 means the TIC bus is occupied by the ISAC-S.
CODR	C/I Code Receive The receipt of a new C/I code in two successive IOM frames, which differs from the previously received value, will release a CIC interrupt (ISTA register). After detection of such a C/I code change the new code can be read from CIRR.
TCR	T-Channel Receive TCR represents the current value received in the T-channel.

Note: The TBA- and CODR-bits of the CIRR will be updated every time CIRR has been read previously by the processor and a C/I code change is recognized. If several C/I code changes were recognized before reading the CIRR register, only the very first and very last changes in the C/I code (also TBA) will be available for the processor to read.

MONR— Monitor Register—(Read/Write)

Address: 32H

Format:

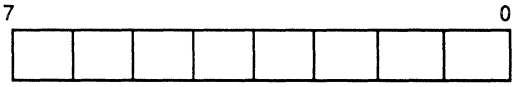


Bit Name	Description
Bit 0–7	The MONR register normally contains control information transferred in the monitor channel of an IOM frame between Layer 2 and Layer 1. In the ISAC-S however, MONR is not required since the monitor channel is used only for the TIC bus access mechanism, which will be automatically realized by the ISAC-S.

BCX1—B1-channel Transmit Register—(Read/Write)

Address: 35H

Format:



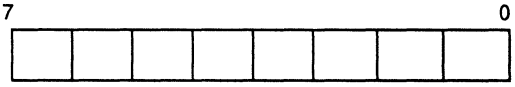
Bit Name	Description
Bit 0–7	BCX1 can be used in microprocessor-controlled interactions with the serial transmission process, for instance in loop or monitor applications. Depending on the selected B-channel switching (SPCR register) and synchronized to the transmission process via SIN interrupt (ISTA register) and BVS-bit (STAR register), the serial data can be read/written from/to BCX1 as follows:

B-channel Switching	Receive Data from (Read)	Transmit Data to (Write)
SLD loop	SLD	SLD
SLD-IOM connection	SLD	/
SSI-IOM connection	SSI	/
IOM loop	IOM	IOM

BCX2—B2-channel Transmit Register —(Read/Write)

Address: 36H

Format:



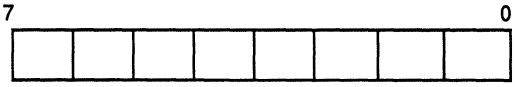
Bit Name	Description
Bit 0–7	BCX2 can be used in microcprocessor-controlled interactions with the serial transmission process, for instance in loop or monitor applications. Depending on the selected B-channel switching (SPCR register) and synchronized to the transmission process via SIN interrupt (ISTA register) and BVS-bit (STAR register), the serial data can be read/written from/to BCX2 as follows:

B-channel Switching	Receive Data from (Read)	Transmit Data to (Write)
SLD loop	SLD	SLD
SLD-IOM connection	SLD	/
SSI-IOM connection	SSI	/
IOM loop	IOM	IOM

BCR1—B1-channel Receive Register—(Read)

Address: 37H

Format:



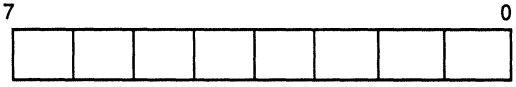
Bit Name	Description
Bit 0–7	BCR1 can be used in microprocessor-controlled monitoring of the serial transmission process by reading data from BCR1 as listed below:

B-channel Switching	Receive Data from (Read)
SLD-IOM connection	IOM
SSI-IOM connection	IOM

BCR2—B2-channel Receive Register—(Read)

Address: 38H

Format:



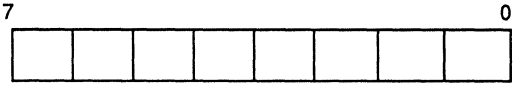
Bit Name	Description
Bit 0–7	BCR2 can be used in microprocessor-controlled monitoring of the serial transmission process by reading data from BCR2 as listed below:

B-channel Switching	Receive Data from (Read)
SLD-IOM connection	IOM
SSI-IOM connection	IOM

SSGX—SLD Signaling Register Transmit—(Write)

Address: 33H

Format:

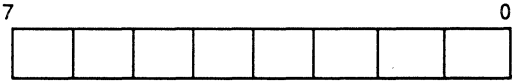


Bit Name	Description
Bit 0–7	The contents of SSGX represent directly the signaling byte (SIG) which will be transmitted continuously in an SLD frame.

SSGR—SLD Signaling Register Receive—(Read)

Address: 33H

Format:

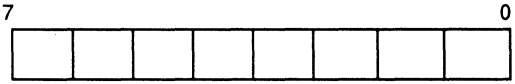


Bit Name	Description
Bit 0–7	The signaling byte of a received SLD-frame can be read from SSGR.

SFCR—SLD Feature Control Register—(Read/Write)

Address: 34H

Format:



Bit Name	Description
Bit 0–7	The Feature Control (FC) byte of an SLD frame (receive) can be read from SFCR. Also, an FC byte to be transmitted in an SLD frame (transmit) has to be written into SFCR. The microprocessor accesses are synchronized to SIN interrupts (ISTA, refer to STCR).

ADFR—Additional Feature Register—(Write)

Value after reset: 00H
Address: 38H

Format:

7							0
WTC1	WTC2	TEM	PFS	CFS	FC2	FC1	ITF

Bit Name	Description

WTC1, WTC2 — Watchdog Timer Control 1, 2

After the watchdog timer mode has been selected, the watchdog timer is started. During every time period of 128 ms the processor has to program the WTC1- and WTC2-bit in the following consecutive sequence:

	WTC1	WTC2
1.	1	0
2.	0	1

to reset and restart the watchdog timer. If not, the timer expires and a WOV interrupt together with a reset pulse, are initiated.

TEM — Test Mode

Setting this bit makes it possible to test the Layer 2 functions of the ISAC-S using the IOM interface. The Layer 1 functions are disabled, and the ISAC-S is fully compatible to the "ICC" at the IOM interface with the following references to the "ICC" pin designation:

ICC (IOM) =	ISAC-S	Meaning
FSC	FSC1	frame synchronization
DC	DCL	interface clock
SDBR	SDO	receive data
SDBX	SDI	transmit data

PFS — Prefilter Select

This bit has to be set, if an external prefilter is connected to S-bus port receive, respectively pins SR1, SR2, and UFI. PFS initiates an internal delay time compensation.

CFS — Configuration Select

Depending on the operating mode, CFS determines clock relations and recovery on S and IOM interfaces.
TE mode:

0 —> The IOM interface clock and frame signal is always active (standby). With the C/I command Timing (TIM), the processor can initialize the "Power Up" state. With C/I command Deactivation Indication (DIU) the "Stand By" state will be reached again. It should be mentioned, however, that it is also possible to activate the S interface directly with the C/I command Activate Request (AR 8/10/L) without the TIM command.

1 —> The IOM interface clock and frame signal is normally inactive ("Power Down"). For activating the S inter face, the "Power Up" state can be initialized via software (SPU-bit in SPCR register). After that, the S interface can be activated with a C/I command Activate Request (AR8/10/L). The "Power Down" state will be reached again with the C/I command Deactivation Indication (DIU).

Note: After reset the IOM interface is always active. To achieve "Power Down" state properly the CFS- bit has to be set previously.

LT-S mode:

0 —> Bit stream on S-interface with the internal PLL. This is to tolerate a variable bit shift from 2 to 8 bit-times (greater distances possible: max. ≤ 1.0 – 1.5 km).

1 —> In bus configurations only a fixed bit shift of 2 bit-times will be accepted according to CCITT (max. round trip delay time; max. ≤ 150 m).

LT-T mode:

0 —> CFS always has to be set to "0."

ADFR (continued)

Bit Name	Description
FC2, FC1	Frame Synchronization Control 2,1 (significant in TE mode only) Adjusts the polarity of the symmetrical 8-kHz-frame output signal (IOM, SLD, SSI). 0 —> normal: High during the first half of the 125-µs frame, Low during the second half. 1 —> inverted: Low during the first half, High during the second half of the 125-µs frame. Note: If the FSC1, FSC2 outputs (in TE mode) supply the data strobe signal for B-channel sources/destinations connected to the Serial Synchron Interface (SSI), it is possible to select individual switching to the B1- or B2-channel.
ITF	Interframe Time Fill ITF selects the interframe time-fill signal which will be sent between HDLC frames when the HDLC port mode is selected in MODE. 0 —> idle (sequence of "1"s) 1 —> flags (sequence of patterns: "0111 1110") When the IOM interface mode is selected, the interframe time fill signal is always idle, according to LAPD.

ABSOLUTE MAXIMUM RATINGS

Storage temperature . . . . . –65 to +125°C
Ambient temperature under bias . . . . . 0 to +70°C
Voltage on any pin with respect to ground . . . . . –0.4 to V_{DD} + 0.4 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) . . . . . 0 to +70°C
Supply Voltage (V_{CC}) . . . . . +5 V ±5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges

T_A = 0 to 70°C, V_{DD} = 5 V ±5%, V_{SSD} = 0 V, V_{SSA} = 0 V

Parameter Symbol	Parameter Descriptions	Test Conditions	Limit Values Min.	Max.	Unit
All pins except SX1, SX2, SR1, SR2, RREF					
V_{IL}	Input Low voltage	—	–0.4	0.8	V
V_{IH}	Input High voltage	—	2.0	V_{CC} + 0.4	V
V_{OL}	Output Low voltage	I_{OL} = 2 mA	—	0.45	V
V_{OH}	Output High voltage	I_{OH} = –400 µA	2.4	—	V
V_{OH}	Output High voltage	I_{OH} = –100 µA	—	V_{CC} –.5	V
I_{CC}	Power supply current operational	V_{DD} = 5V, Inputs at 0 V/V_{DD}	—	13	mA
	Power supply current power down	No output loads	—	1.3	mA
I_{LI}	Input leakage current	0 V < V_{IN} < V_{DD} to 0 V	—	+10	µA
I_{LO}	Output leakage current	0 V < V_{OUT} < V_{DD} to 0 V	—	+10	µA
SX1, SX2					
VX	Absolute value of output	R_L = 25 ohms**	—	0.3	V
	Pulse amplitude	R_L = 200 ohms	1.35	1.65	V
	(VSX2–VSX1)*	R_L = 1600 ohms	1.35	2.4	V
IX	Transmitter output current	R_L = 200 ohms	—	8.25	mA
RX	Transmitter output	Inactive or during binary one	10	—	kohm
	Impedance	During binary zero	80	—	ohm
		R_L = 200 ohms			
SR1, SR2					
VSR1	Receiver output voltage	I_o < 100 µA	2.4	2.6	V
VTR	Receiver threshold voltage VSR2–VSR1	Dependent on peak level	+225	+375	mV
RREF					
VO	Voltage at RREF	R_{REF} = 2.2 kohm ±1%	1.0	1.2	V
IO	Output Current	R_{REF} = 2.2 kohm ±1%	450	550	µA

Notes: *Due to the transformer, the pulse amplitude zero to peak on S interface line will be halved.
**Load resistance on S interface line will be divided by four.

CAPACITANCES

T_A = 25°C, V_{DD} = 5 V ±5%, V_{SSD} = 0 V, V_{SSA} = 0 V

Parameter Symbol	Parameter Descriptions	Test Conditions	Limit Values Min.	Max.	Unit
All pins except SR1,SR2, XTAL1, XTAL2					
C_{IN}	Input capacitance	—	—	7	pF
C_{IO}	I/O	—	—	7	pF
SX1, SX2					
C_{OUT}	Output capacitance against V_{SSA}	—	—	10	pF
SR1, SR2					
C_{IN}	Input capacitance	—	—	7	pF
XTAL1, XTAL2					
C_{LD}	Load capacitance	—	—	50	pF

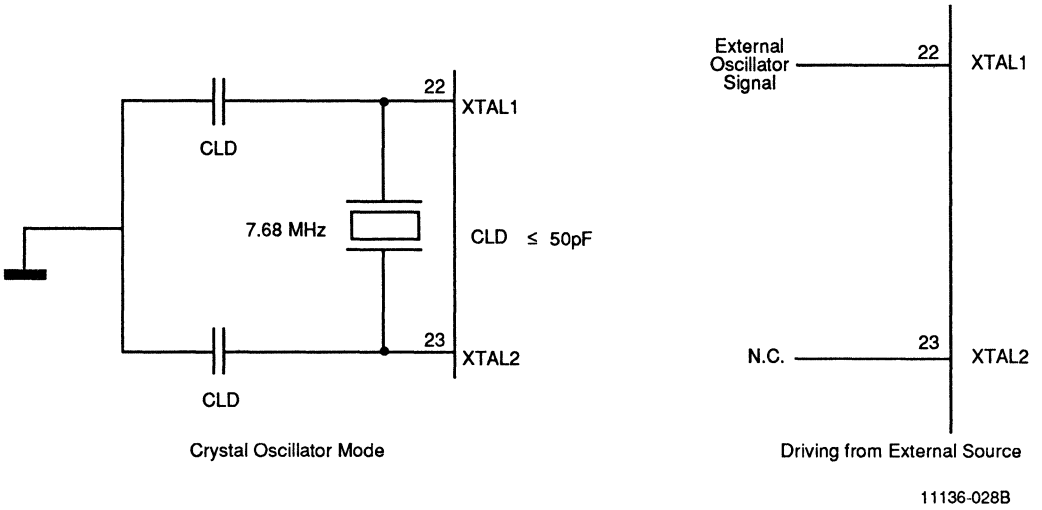


Figure 28. Recommended Oscillator Circuits

SWITCHING CHARACTERISTICS

T_A = 0 to 70°C, V_{DD} = 5 V ±5%, V_{SSD} = 0 V, V_{SSA} = 0 V

Inputs are driven at 2.4 V for a logic "1" and at 0.4 V for a logic "0." Timing measurements are made at 2.0 V for a logic "1" and at 0.8 V for a logic "0." The AC testing input/output waveforms are shown below.

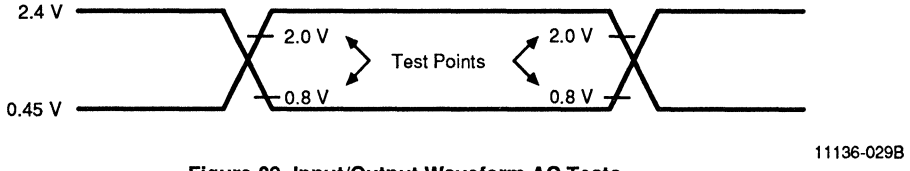
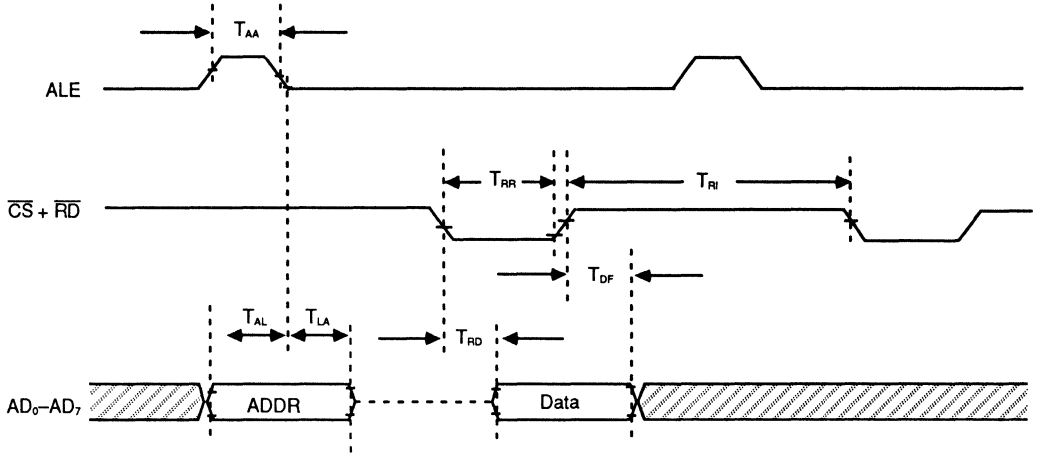


Figure 29. Input/Output Waveform AC Tests

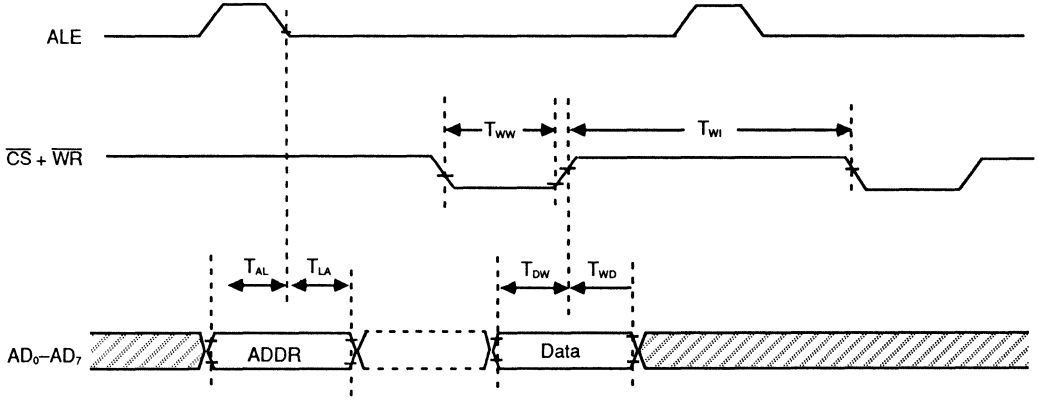
Table 11. Microprocessor Interface Timing

Parameter Symbol	Parameter Description	Test Conditions	Limit Values Min.	Max.	Unit
T_{AA}	ALE pulse width	—	50	—	ns
T_{AL}	Address setup time at ALE	—	20	—	ns
T_{LA}	Address hold time from ALE	—	10	—	ns
T_{RR}	$\overline{RD}$ pulse width	—	110	—	ns
T_{RD}	Data output delay from $\overline{RD}$	—	—	25	ns
T_{DF}	Data float delay from $\overline{RD}$	—	—	110	ns
T_{RI}	$\overline{RD}$ control interval	—	70	—	ns
T_{WW}	$\overline{WR}$ pulse width	—	60	—	ns
T_{DW}	Data setup time to $\overline{WR}$ + $\overline{CS}$	—	35	—	ns
T_{WD}	Data hold time from $\overline{WR}$ + $\overline{CS}$	—	10	—	ns
T_{WI}	$\overline{WR}$ control interval	—	70	—	ns



11136-030B

Figure 30. Microprocessor Read Cycle



11136-031B

Figure 31. Microprocessor Write Cycle

Table 12. SLD Interface Timing

Parameter Symbol	Parameter Description	Test Conditions	Limit Values Min.	Max.	Unit
T_{SLD}	SLD data delay	—	20	100	ns
T_{SLS}	SLD data setup	—	30	—	ns
T_{SLH}	SLD data hold	—	30	—	ns
T_{FSH}	Frame sync. hold	—	30	—	ns
T_{FSS}	Frame sync. setup	—	50	—	ns
T_{FSW}	Frame sync. width	—	40	—	ns
T_{FSD}	Frame sync. delay	—	–20	20	ns

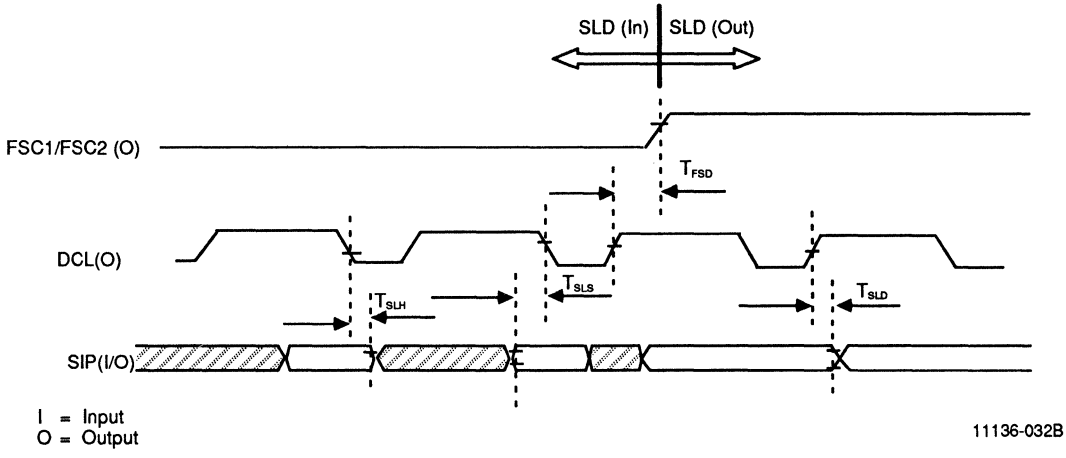


Figure 32. SLD (TE Mode)

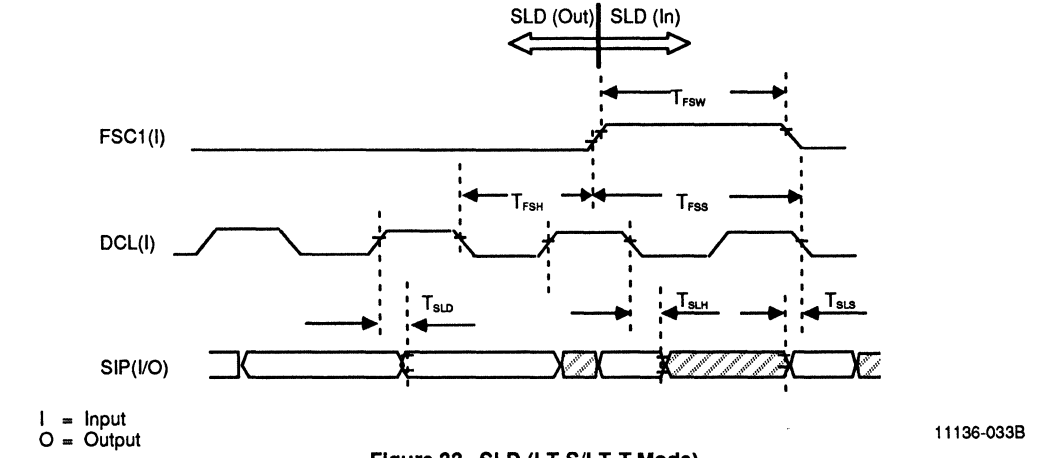
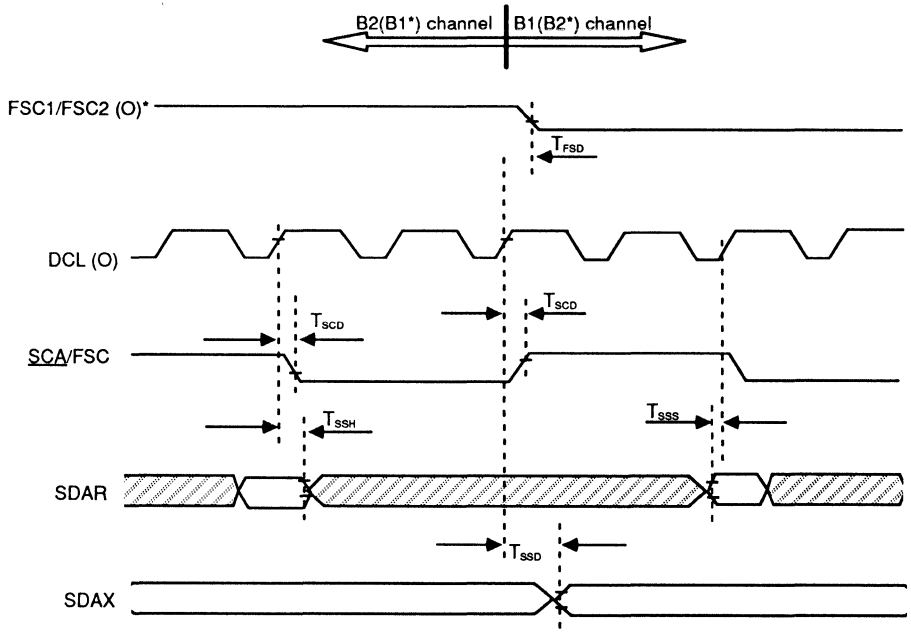


Figure 33. SLD (LT-S/LT-T Mode)

Table 13. SSI Timing

Parameter Symbol	Parameter Description	Test Conditions	Limit Values Min.	Max.	Unit
T_{SCD}	SCA clock delay	—	20	100	ns
T_{SSD}	SSI data delay	—	20	100	ns
T_{SSS}	SSI data setup	—	30	—	ns
T_{SSH}	SSI data hold	—	30	—	ns



*Default polarity
Individual B-channel switching to the B1 and B2 channel can be selected by programming the output polarity of PSC1 and FSC2 in the ADFR register.

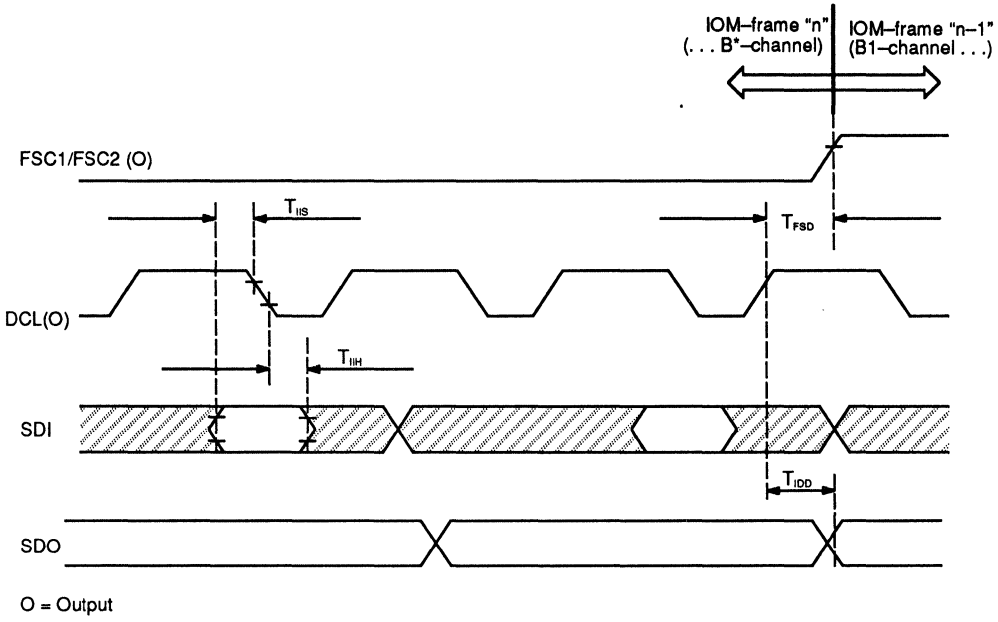
11136-034B

O = Output

Figure 34. SSI (TE Mode)

Table 14. IOM Interface/(HDLC Port) Timing

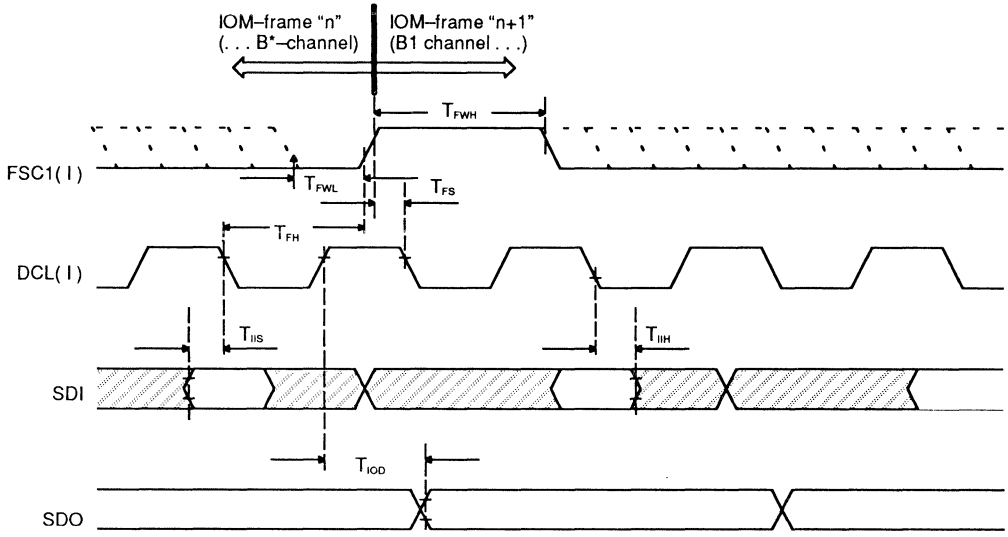
Parameter Symbol	Parameter Description	Test Conditions	Limit Values Min.	Max.	Unit
T_{DH}	Frame sync. hold	—	30		ns
T_{FS}	Frame sync. setup	—	50		ns
T_{FWH}	Frame sync. high	—	40		ns
T_{FWL}	Frame sync. low	—	2150		ns
T_{FSD}	Frame sync. delay	—	–20	20	ns
T_{IOD}	IOM output data delay	—	—	200	ns
T_{IIS}	IOM input data setup	—	20		ns
T_{IIH}	IOM input data	—	50		ns



O = Output

Figure 35. IOM (TE Mode)

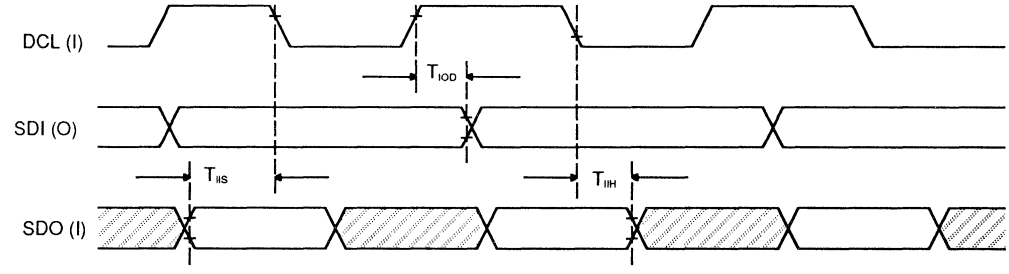
11136-035B



Note: At the SDI pin an internal Pull-Up Resistor is integrated for the TIC bus configuration with wired-OR connections (open drain outputs).

11136-036B

Figure 36. IOM (NT, LT-S, LT-T Mode)



HDLC port mode: – FSC1 = "High"
– HMD0,1,2 bits in MODE are set
– TEM—bit in ADFR is set

11136-37B

Figure 37. HDLC Port Mode

Clock Timing

The clocks in the different operating modes are summarized in Table 15, with the respective duty ratios.

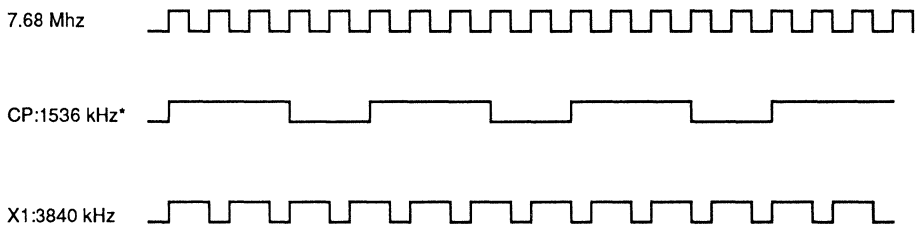
Table 15. ISAC-S Clock Signals

Application	M1	M0	DCL	FSC1/FSC2	CP	X1	X0
TE	0	0	O:512 kHz* 2:1	O:8 kHz* 1:1	O:1536 kHz* 3:2	O:3840 kHz 1:1	
LT-T	0	1	I:512 kHz	I:8 kHz	O:512 kHz* 2:1	—	—
LT-S	1	0	I:512 kHz	I:8 kHz	— —	O:7680 kHz 1:1	I:fixed at 0
NT	1	1	I:512 kHz	I:8 kHz	—	—	—

*Synchronous to receive "S" line P.

Clock CP is phase-locked to the receive S signal and is derived using the internal DPLL and the 7.68-MHz ±100 ppm crystal (TE and LT-T). A phase tracking of CP with respect to "S" is performed once in 250 µs. As a consequence of this DPLL tracking, the high state of CP may be either reduced or extended by one 7.68-MHz period (CP duty ratio 2:2 or 4:2 instead of 3:2) once every 250 µs. Since DCL and FSC1/FSC2 are derived from CP (TE mode), the high or low states of FSC1/FSC2 and DCL may likewise be reduced or extended by the same amount once every 250 µs.

Note: The phase adjustment may take place either in the sixth, seventh, or eighth CP cycle counting from the beginning of an IOM frame in TE.



*Synchronous to receive "S," Duty ratio 3:2 normally

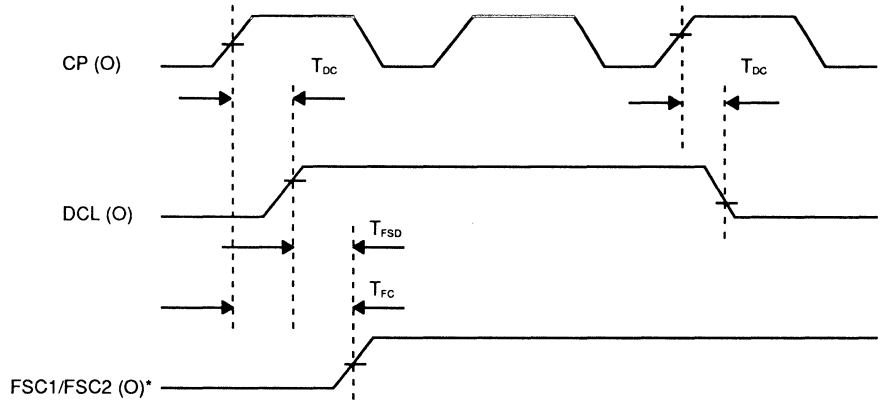
11136-038B

Figure 38. Phase Relationships of Auxiliary Clocks

The timing relationships between CP, DCL and FSC1/FSC2 are specified in Table 16 and Figure 39.

Table 16. CP, DCL, and FSC1/FSC2 Timing Specifications

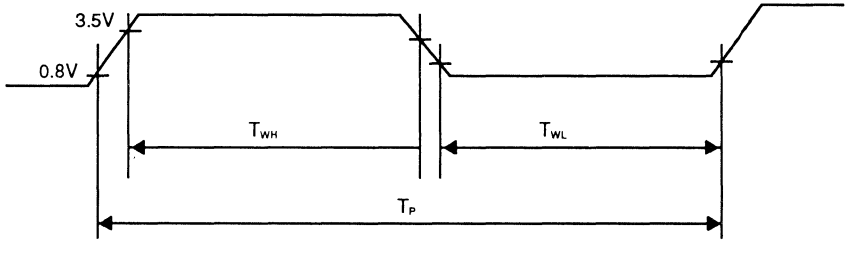
Symbol	Description	Test Conditions	Min.	Max.	Unit
T_{DC}	Clock delay CP-DCL	—	0	50	ns
T_{FC}	Clock delay CP-SC	—	0	50	ns
T_{FSD}	Frame sync. delay	—	–20	20	ns



*Default polarity (can be programmed in ADFR).

11136-039B

Figure 39. CP, DCL, FSC1/FSC2 Timing Relationships



11136-040B

Figure 40. Definition of Clock Period and Width

Tables 17 through 19 give the timing characteristics of the clocks.

Table 17. Clocks Timing

Symbol	DCL Description	Test Conditions	Min.	Typ.	Max.	Unit
T_P Output	(TE) 512 kHz	Osc ±100 ppm	1822	1953	2084	ns
T_{WH} Output	(TE) 512 kHz 2:1	Osc ±100 ppm	1121	1302	1483	ns
T_{WL} Output	(TE) 512 kHz 2:1	Osc ±100 ppm	470	651	832	ns
T_{WH} Output	(NT, LT-S, LT-T)	—	200	—	—	ns
T_{WL} Output	(NT, LT-S, LT-T)	—	200	—	—	ns

Table 18. Clocks Timing

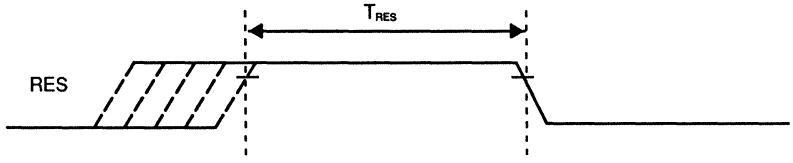
Symbol	CP Description	Test Conditions	Min.	Typ.	Max.	Unit
T_P Output	(TE) 1536 kHz	Osc ±100 ppm	520	651	782	ns
T_{WH} Output	(TE) 1536 kHz	Osc ±100 ppm	240	391	541	ns
T_{WL} Output	(TE) 1536 kHz	Osc ±100 ppm	24	260	281	ns
T_R, T_F	(TE, LT-T)	CL = 100 pF	—	—	20	ns
		CL = 50 pF	—	—	10	ns
T_P Output	(LT-T) 512 kHz	Osc ±100 ppm	1822	1953	2084	ns
T_{WH} Output	(LT-T) 512 kHz	Osc ±100 ppm	1121	1302	1483	ns
T_{WL} Output	(LT-T) 512 kHz	Osc ±100 ppm	470	651	832	ns

Table 19. Clocks Timing

Symbol	X1 Description	Test Conditions	Min.	Typ.	Max.	Unit
T_P Output	(TE) 3840 kHz	Osc ±100 ppm	−100 ppm	260	+100 ppm	ns
T_{WH} Output	(TE) 3840 kHz	Osc ±100 ppm	120	130	140	ns
T_{WL} Output	(TE) 3840 kHz	Osc ±100 ppm	120	130	140	ns

Table 20. Reset

Symbol	Parameter	Test Conditions	Min.	Unit
T_{RES}	Length of active high-state cycles	Power on/power down to Power Up (standby) During Power Up (standby)	4 2 • DCL clock	ms



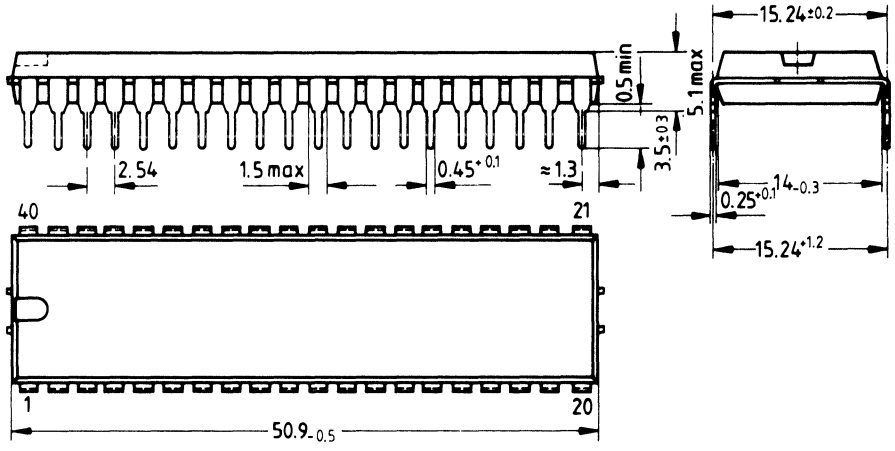
11136-041B

Figure 41. Reset Width

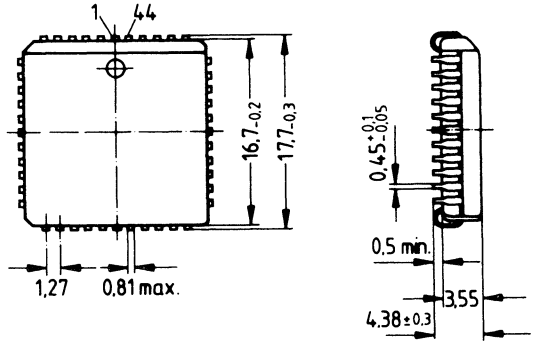
PHYSICAL DIMENSIONS

Note: All dimensions in metric.

PD040



PL044



Am2085

ISAC-S A2 ISDN Subscriber Access Controller

Advanced Micro Devices

According to CCITT I.430 the electrical characteristics of the S/T interface transmitter are to fulfill the following requirements:[1]

- The output impedance, when the transmitter is inactive or transmitting a binary '1,' should exceed 2500 ohms. Note that this also applies to TEs with local power sources when the local power is switched off, although the TE is connected to an activated S-bus.
- The output impedance, when the transmitter is transmitting a binary '0,' should be $\geq$ 20 ohms. Note that this also applies in the case of a 400 ohm load when the transmitter reaches a current- or voltage-limiting state.
- Pulse shape and amplitude shall be in accordance with the given pulse masks. Note that in TE applications the effective test load for the transmitter not only consists of twice the terminating resistance (50 ohms) but also of the series resistances of other external components such as the transformer and the cord (maximum length 7 m).

The transmitter circuitry of the Am2085 ISAC-S A2 meets these requirements in full.

The transmitter essentially is a current-limited voltage source, delivering nominally 1.5 V to the 2:1 transformer. The high output impedance when transmitting a binary '1,' including when the power supply is switched off, is guaranteed by the circuit design.

When transmitting a binary '0,' the output impedance (Z_{OUT}) has to be in the range 20–25 ohms in order that the pulse mask be fulfilled. Since the internal output impedance of the transmitter is negligible for a binary '0,' the impedance is realized by the total sum of external resistances (Figure 1); for example:

Resistance of cord (TE only) (R_{CORD})	4-7 Ω
Copper resistance of transformer (R_{COPPER})	1-3 Ω
Resistances for overvoltage protection of the transmitter, transformed to the primary (line) side with 4:1 ($R_{OVERVOLT}$)	15 Ω

[1]ISDN User-Network Interfaces: Layer 1 Recommendations.
CCITT Recommendations of the Series I—Volume III Fascicle III.5 Integrated Services Digital Network (ISDN). VIIIth Plenary Assembly Malaga - Torremolinos, 8–10 October 1984.

Publication #	Rev.	Amendment
11136	C	/1

Issue Date: July 1989

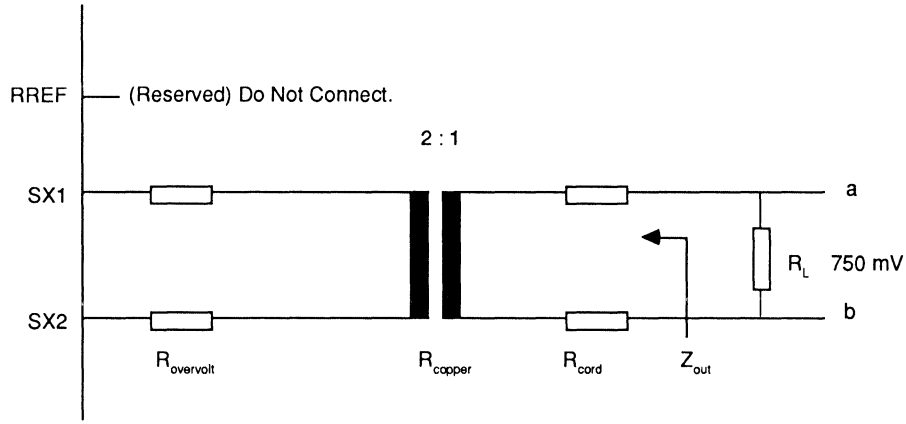


Figure 1. Transmitter Environment Model of the ISAC-S A2

DC CHARACTERISTICS

TA = 0 to 70° C; V_{DD} = 5 V ± 5%; V_{SS} = 0 V

Symbol	Parameter	Limit Values Min.	Max.	Units	Test Condition
VX	Absolute value of output pulse amplitude | (VSX1 - VSX2) |	2.03 2.10	2.31 2.39	V V	RL = 50 Ω * RL = 400 Ω *
IX	Transmitter output current	7.5	13.4	mA	RL = 5.6 Ω *

* Due to the transformer, the load resistance seen by the Am2085 is four times R_L.

Preliminary

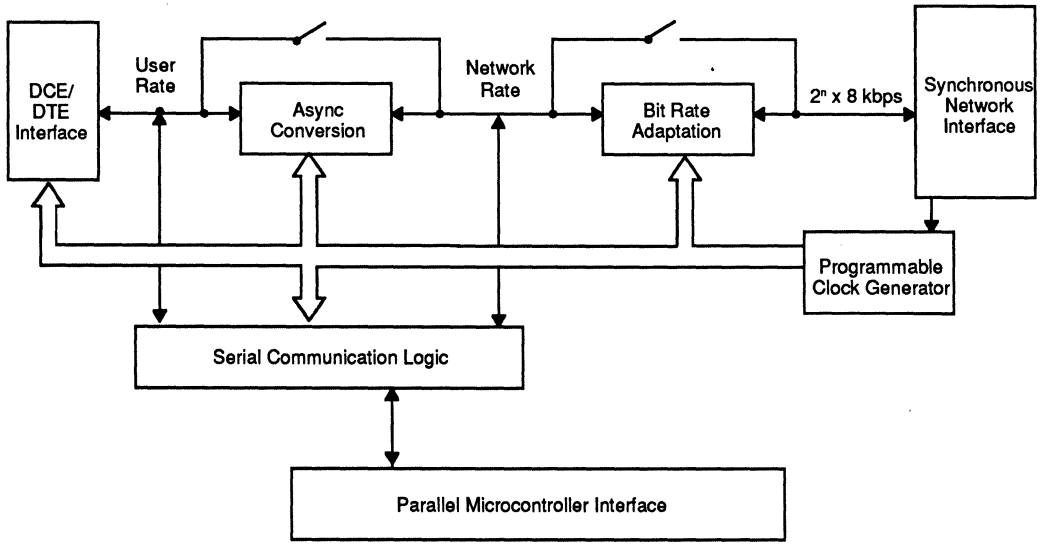
Am2110
ISDN Terminal Adapter Circuit (ITAC)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Universal adapter for ISDN R reference point
- Support of async and sync interfaces: X.21, X.21 bis, V.24, RS232C
- Programmable speeds from 300 bps to 64 kbps
- Bit rate adaptation according to X.30, V.110, ECMA.102, V.120 and DMI
- Programmable time slots and subchannels for intermediate rates
- Automatic calling/answering with on-chip controllers
- In-band parameter exchange support
- Parallel 8-bit microcontroller interface
- DMA support
- Single +5 V supply, low power CMOS technology

BLOCK DIAGRAM



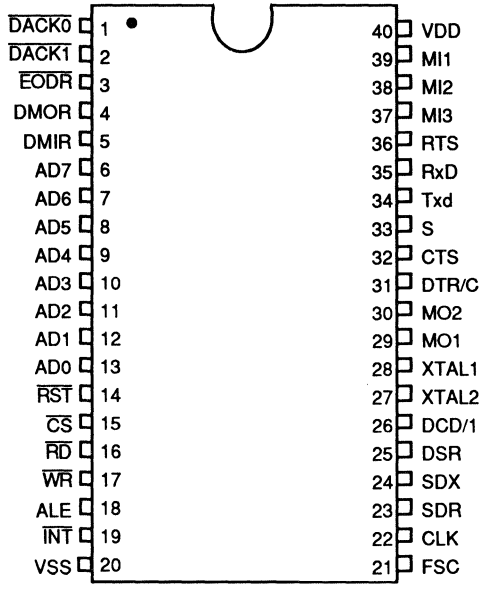
Publication # 11154 Rev. B Amendment /0
Issue Date: May 1989

GENERAL DESCRIPTION

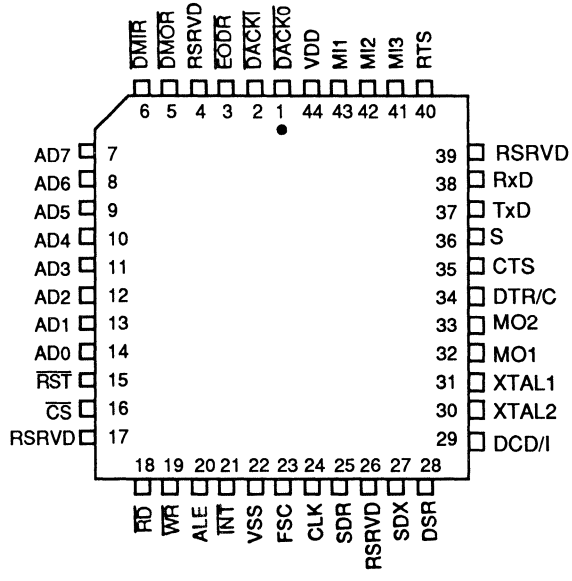
The ISDN Terminal Adapter Circuit (ITAC) is a monolithic, full custom circuit for interfacing standard terminals and PCs to a circuit switched data network or an ISDN. It may be programmed to perform bit rate adaptation for 64 kbps clear channels according to the newest rate adaptation protocols. The on-chip communication controllers handle signaling between data equipment and the network, effectively replacing the "smart modem" of the PSTN. The features of the ITAC make it suitable for use in advanced networking applications that require flow control, in-band parameter exchange and interworking.

CONNECTION DIAGRAMS
Top View

40-Pin DIP

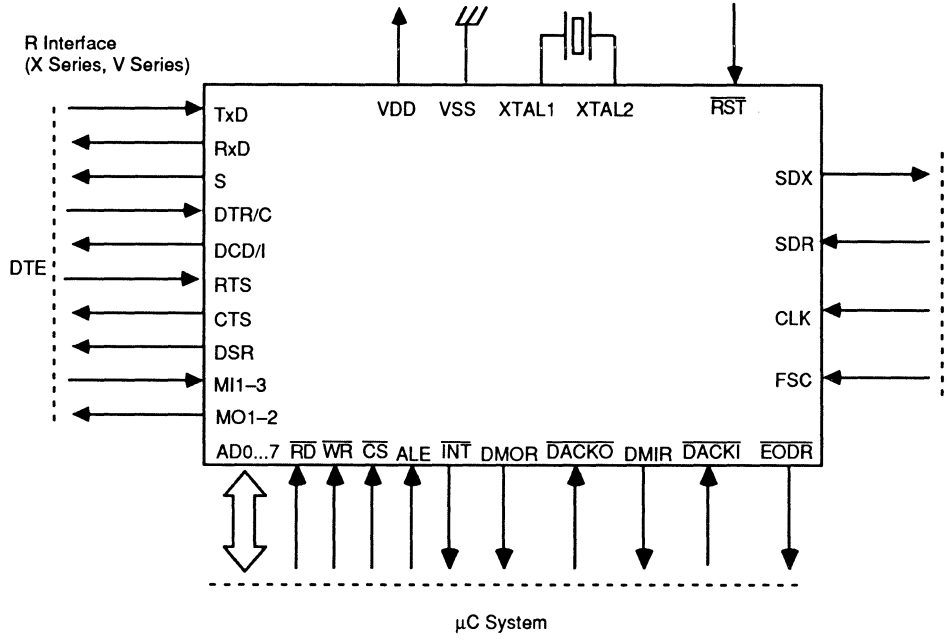


44-Pin PLCC



Note: Pin 1 is marked for orientation.

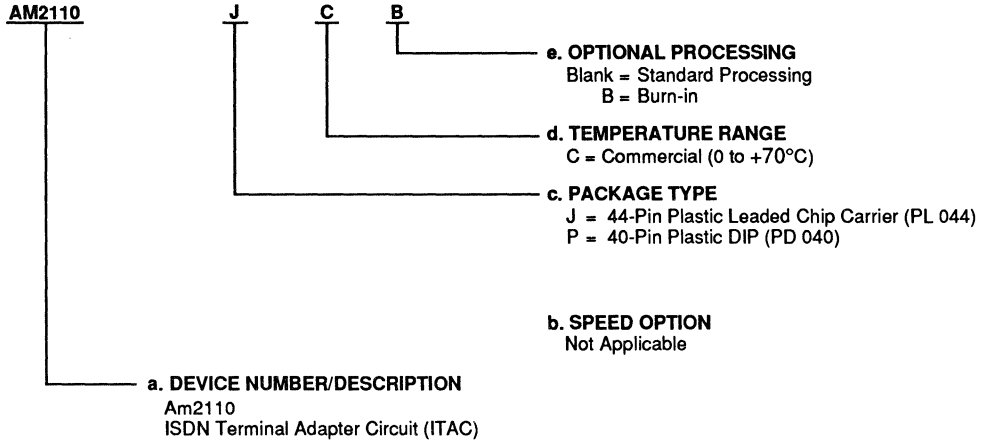
LOGIC SYMBOL



ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM2110	JC, JCB, PC, PCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

DCE Mode

AD0–AD7
Address/Data Bus (Input/Output)

ALE
Address Latch Enable (Input)

CLK
Clock (Input)
Data clock for the synchronous network interface.

$\overline{CS}$
Chip Select (Input)

CTS
Clear to Send (Output)
(106) V.24 interchange circuit.

$\overline{DACKI}$
DMA Input Acknowledge (Input)

$\overline{DACKO}$
DMA Output Acknowledge (Input)

DCD/I
Data Channel Received Line Signal Detector (Output)
Carrier Detect (109) V.24/Indicate X.21 interchange circuit.

DMIR
DMA Input Request (Output)

DMOR
DMA Output Request (Output)

DSR
Data Set Ready (Output)
(107) V.24 interchange circuit.

DTR/C
Data Terminal Ready (Input)
(108) V.24/control X.21 interchange circuit.

$\overline{EODR}$
End of DMA Output Request (Output)

FSC
Frame Sync (Input)
8 kHz.

$\overline{INT}$
Interrupt (Open Drain)
Open-drain interrupt request.

MI1–3
Multifunctional (Input)
V.24 interchange circuit.

MO1–2
Multifunctional (Output)
V.24 interchange circuit.

$\overline{RD}$
Read Enable (Input)

$\overline{RST}$
Reset (Input)

RTS
Request to Send

RxD
Receive Data to DTE (Output)
Data is clocked off by the ITAC on the falling edge of "S" on synchronous DT interfaces.

S
(Output)
Bit element timing for synchronous DTE.

SDR
Synchronous Data Receive (Input)
Data are input on the falling edge of CLK.

SDX
Synchronous Data Transmit (Output)
Data are on the rising edge of CLK.

TxD
Transmit Data from DTE (Input)
Data is latched by the ITAC on the rising edge of "S" on synchronous DTE interfaces.

$\overline{WR}$
Write Enable (Input)

XTAL1
Connection for External Crystal (Input)
Input for external clock generator.

XTAL2
Connection for External Crystal (Output)
N.C. when clock generator is used.

VDD
(Input)
Power supply, ±5 V ±5%.

VSS
(Input)
Power supply, ground.

DTE Mode

DCD

DSR
Data Set Ready (Input)
(107) V.24 interchange circuit.

DTR
Data Terminal Ready (Output)
(108) V.24 interchange circuit.

MO3
Multifunctional (Output)
V.24 interchange circuit.

RTS
Request to Send (Output)
(105) V.24 interchange circuit.

RxD
Receive Data from DCE (Input)

TxD
Transmit Data to DCE (Output)

PIN NAMES

Note: Pin names refer to DCE mode.

AD0–7	I/O	Address/Data bus
ALE	I	Address Latch Enable
CLK	I	Network Clock
$\overline{CS}$	I	Chip Select
CTS	O	Clear to Send
$\overline{DACKI}$	I	DMA Input Acknowledge
$\overline{DACKO}$	I	DMA Output Acknowledge
DCD/I	O	Carrier Detect/Indicate
DMIR	O	DMA Input Request
DMOR	O	DMA Output Request
DSR	O	Data Set Ready
DTR/C	I	Data Terminal Ready/Control
$\overline{EODR}$	O	End of DMA Output Request
FSC	I	8 kHz Frame Sync Clock
$\overline{INT}$	O	Interrupt request
MI1–3	I	Multifunctional Inputs
MO1–2	O	Multifunctional Outputs
$\overline{RD}$	I	Read Enable
$\overline{RST}$	I	Reset
RTS	I	Request to Send
RxD	O	Receive Data
S	O	Bit Clock
SDR	I	Synchronous Data Receive
SDX	O	Synchronous Data Transmit
TxD	I	Transmit Data
$\overline{WR}$	I	Write Enable
XTAL1,2	I/O	Connections for 10.752 MHz crystal or external oscillator

System Integration

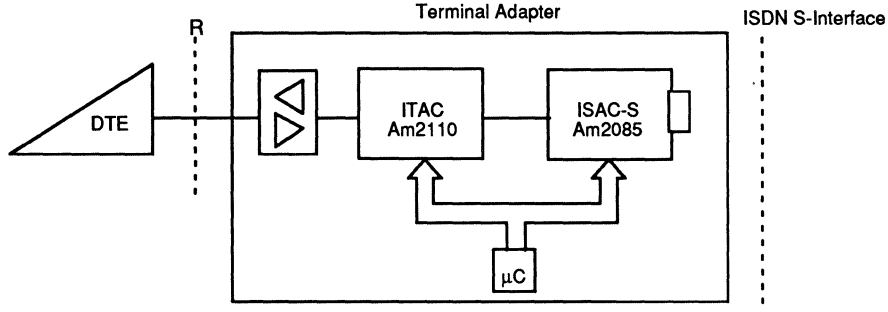
Terminal Adapter for ISDN Basic Access

A typical implementation of an ISDN basic access for a conventional X- or V-series terminal using the ITAC is shown in Figure 1.

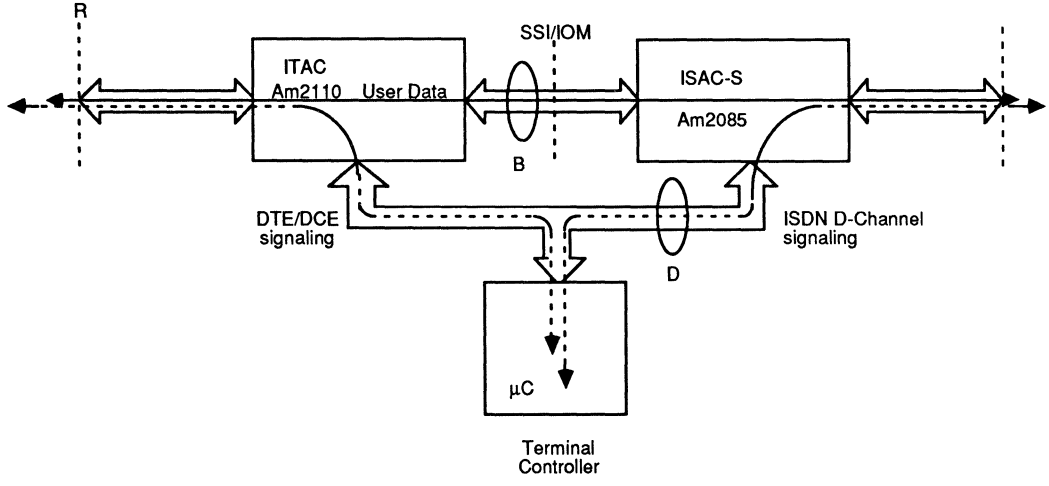
The ITAC can be connected via a serial synchronous interface to an ISDN basic access transceiver/LAPD controller (in this case, the ISDN Subscriber Access Controller for the S interface, ISAC-S). These two devices, together with the terminal controller, convert V- and X-series interface characteristics to the functional and procedural interface characteristics required by an ISDN at reference point S.

The ITAC subchannel multiplexing feature allows sharing a single 64 kbps bearer channel by up to eight independent terminals. This is illustrated in Figure 2.

Figure 2 also illustrates how the ISDN Subscriber Access Controller for the U interface (ISAC-P) can be used, instead of the ISAC-S, in U interface applications.



(a)



(b)

Figure 1. (a) Universal TA for the ISDN basic access R reference point (bit stuffing or flag stuffing bit rate adaptation)

(b) Data paths for bit rate adapation according to V.110/X.30

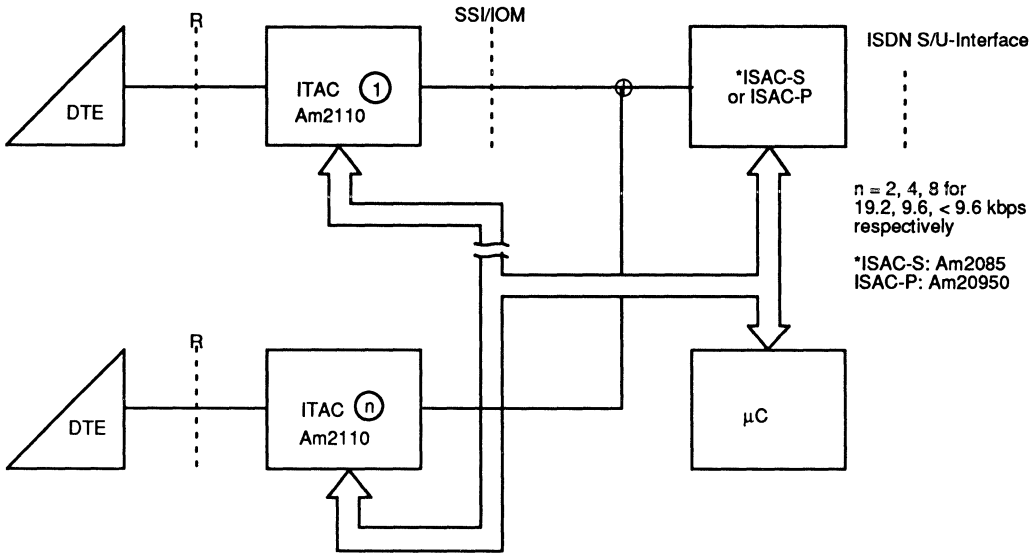


Figure 2. Sharing of bearer channels among several independent DTEs (up to 16 DTEs for two B channels).

PC Adapter Board

The ITAC may be used to enable communication between a personal computer and conventional asynchronous or synchronous terminals. In this case, the data is sent and received over the parallel microcontroller interface, via the integrated communication FIFOs, or using DMA. Depending on the bit rate adaptation scheme used, the ITAC performs the appropriate formatting functions (V.110, V.120 or DMI, or V.110 embedded HDLC).

Other Applications

The synchronous network interface of the ITAC is compatible to most PCM systems using programmable time slots. Consequently, the circuit in association with all IOM-compatible circuits is ideally suited for applications on PABX line cards and concentrators/multiplexers.

Other applications of the ITAC include: host computer multiple line communication couplers, primary access/DMI peripheral boards, and Interworking Units (IWUs) between ISDN and analog PSTN.

FUNCTIONAL DESCRIPTION

General Functions and Device Architecture

A simplified block diagram of the ITAC is on page 1.

Data conversion is implemented by two main blocks. First, user data is transformed to a data rate synchronous to the ISDN, called the Network rate, by an Async/Sync converter.

The Bit Rate Adaption block converts the Network rate into a rate which can be transmitted to the ISDN, 2^n x 8 kbps, n = 0,1,2,3.

For synchronous switched-through DTE data, the Asynch/Synch Converter is not used, and the Network rate is identical to the User rate. Otherwise the Network rate is defined independent of the User rate.

Through the Serial Communication Logic, the microcontroller can access the receive and transmit data.

An architectural overview of the ITAC is shown in Figure 3.

The Programmable Clock Generator provides timing synchronized by the network master clock, to the different functional blocks.

The Async/Sync Converter (ASC) block contains the stop bit and break signal manipulations necessary for asynchronous DCE/DTE interfaces according to V.22. The Intermediate Rate Conversion (IRC) and the Bearer Rate Conversion (BRC) blocks correspond to the RA1 and RA2 stages of X.30 (I.461) and V.110 (I.463) CCITT recommendations.

The Data Multiplexer switches the data between the DCE/DTE interface, the Intermediate Rate Conversion block and the Serial Communication Logic. The latter consists of a Universal Synchronous/Asynchronous-receiver and transmitter section and of an HDLC receiver and transmitter section, each with an integrated FIFO. The flexible FIFOs with DMA capability are optimized for fast parallel access to the control and user data streams.

The ITAC performs recognition of local and remote DTE/DCE states via the Status Detect Logic.

The microcontroller interface consists of registers necessary to configure the circuit and to monitor state changes.

Finally, testing capabilities are provided, including test loops for data from local and from remote data terminals.

Operating Modes

The operating mode of the ITAC depends on the user interface type and on the bit rate adaptation protocol.

In the case of a synchronous DTE, either an X.21 or an X.21 bis (synchronous V.24) may be selected. Interworking between X.21 and X.21 bis DTEs is also provided for.

In the case of an asynchronous DCE/DTE interface, the ITAC may be used as a DCE or a DTE.

Transparent Mode

This mode is applied in the data transfer phase of a data call where only layer 1 conversion is required.

In this mode, the microcontroller need only monitor the changes of call status through the Status Detect Logic and/or through generated interrupts. All conversion functions are implemented by the ITAC and are transparent to the microcontroller (Figure 4).

The selectable User rates are summarized in Table 1, for both synchronous and asynchronous DCE/DTE interfaces.

For each of the async rates, the user is able to select the character length, the number of Stop elements, as well as the tolerance range for the handling of DTE overspeed.

Table 1. User Data Rates

User rate, bps	Async data	Sync data
300	X	
600	X	X
1200	X	X
2400	X	X
4800	X	X
9600	X	X
19200	X	X
38400	X	X
48000		X
56000		X
64000		X

As the name of this operation mode implies, the user data is handled transparently; in particular, no parity checking or generation is performed by the ITAC.

Non-transparent Mode

In the non-transparent mode, the Serial Communication Logic is involved in the reception or the transmission of data or control information. This mode of operation is used in particular in the setup phase of a data call, in the simulation of a Hayes Smartmodem command state, for In-band Parameter Exchange and in Host or Personal Computer applications (Figure 5).

It is realized in practice by switching the USART receiver or the HDLC receiver and/or the USART transmitter or the HDLC transmitter into the data path between the DCE/DTE interface and the network interface.

In this mode, higher-level functions are optionally performed by the ITAC on the user data, for example, parity generation and checking.

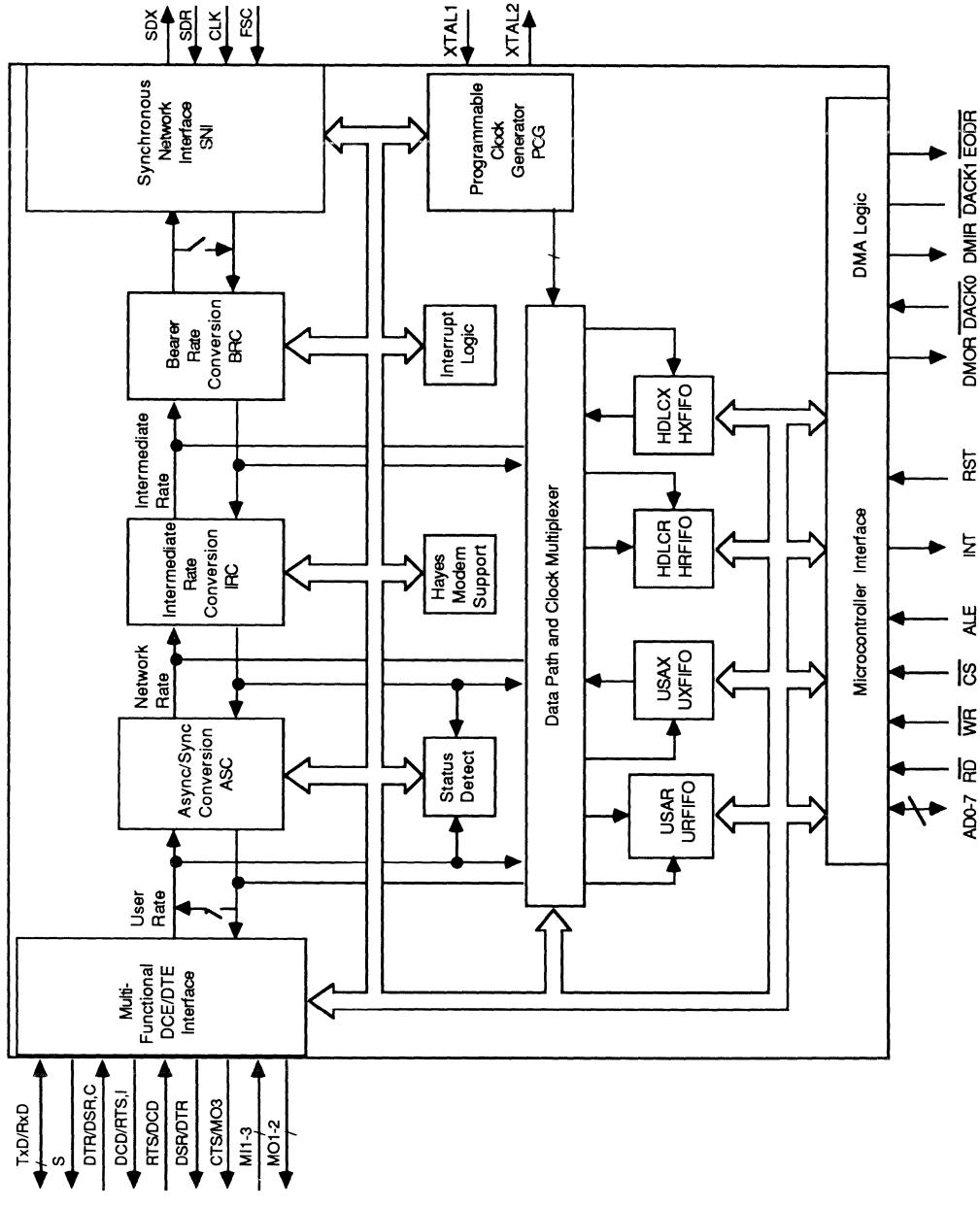


Figure 3. General Architecture of ITAC

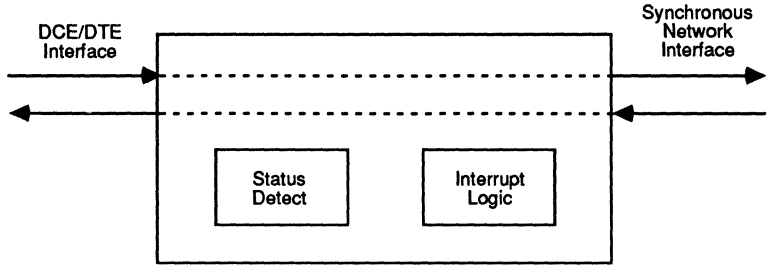


Figure 4. Transparent Mode

Interfaces

The ITAC supports interfaces:

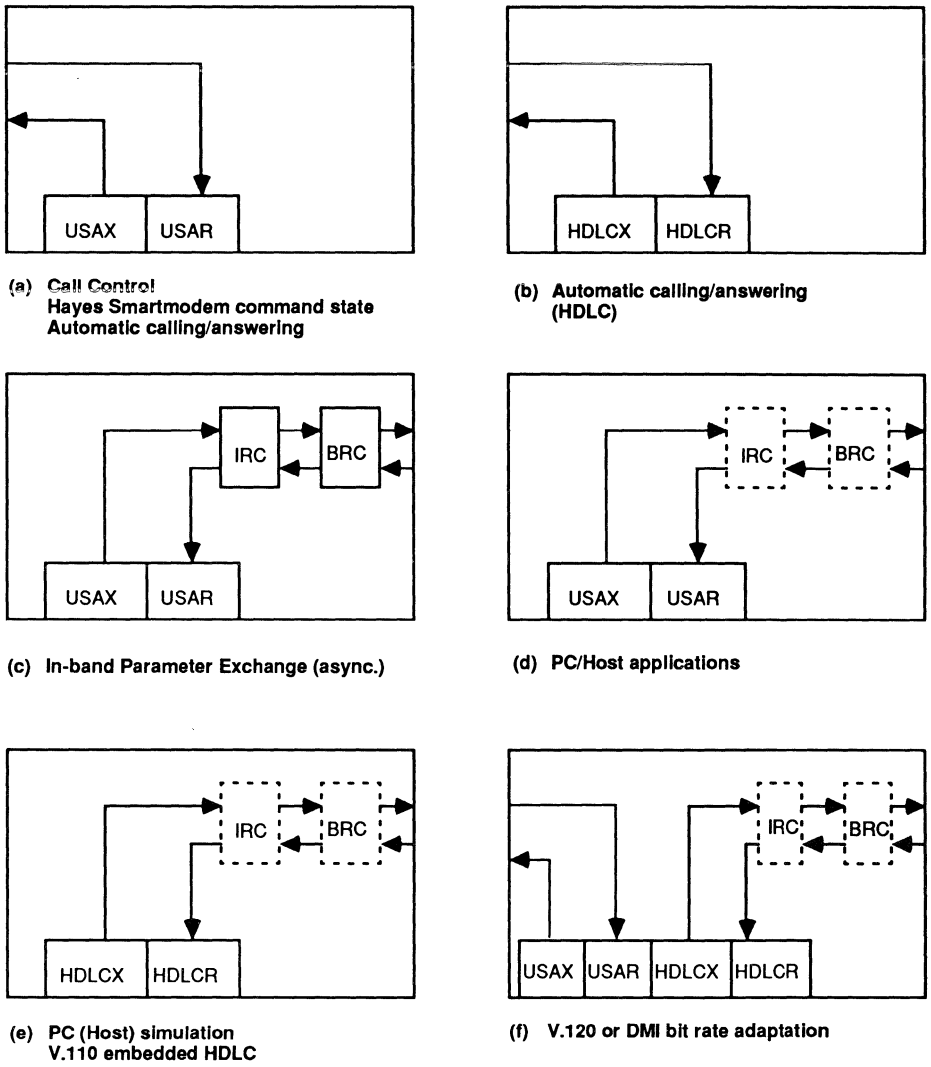
- to a microcontroller system
- to synchronous/asynchronous DTEs
- to the network

Parallel Microcontroller Interface

The ITAC is programmable over an 8-bit parallel microcontroller interface. Easy and fast microprocessor access is provided by 8-bit address decoding on chip. The interface consists of 18 lines and is directly compatible with processors of the multiplexed address/data bus type (Table 2).

Table 2. Microcontroller Interface Signals for ITA

Symbol	Type	Name and Function
AD0–AD7	I/O	Address-Data bus. The multiplexed address/data bus transfers data and commands between the µC system and the ITAC.
$\overline{CS}$	I	Chip Select. A low on this signal selects the ITAC for a read/write operation.
$\overline{WR}$	I	Write. This signal indicates a write operation.
$\overline{RD}$	I	Read. This signal indicates a read operation.
$\overline{INT}$	OD	Interrupt Request. The signal is activated when the ITAC requests an interrupt. It is an open drain output.
ALE	I	Address Latch Enable. A high on this line indicates an address on the external address/data bus.
DMOR	O	DMA Output Request. The signal is activated when the ITAC wishes to output a byte of data via DMA.
DMIR	O	DMA Input Request. The signal is activated when the ITAC wishes to receive a byte of data via DMA.
$\overline{DACK0}$	I	DMA Output Acknowledge. DMA Controller's response to DMOR.
$\overline{DACK1}$	I	DMA Input Acknowledge. DMA Controller's response to DMIR.
$\overline{EODR}$	O	End-of-DMA output request for HDLC receiver.



(a) Call Control
Hayes Smartmodem command state
Automatic calling/answering

(b) Automatic calling/answering
(HDLC)

(c) In-band Parameter Exchange (async.)

(d) PC/Host applications

(e) PC (Host) simulation
V.110 embedded HDLC

(f) V.120 or DMI bit rate adaptation

Figure 5. Non-transparent Modes of the ITAC

DCE/DTE Interface

The DCE/DTE interface of the ITAC consists of six input and seven output lines. In addition to common X.21 and V.24 interface interchange circuits, the circuit supports three programmable, multifunctional inputs and two programmable, multifunctional outputs.The assignment of the DCE/DTE interchange circuits are as shown in Table 3.

The bits received and transmitted on RxD and TxD are such that logical "1" = high voltage and logical "0" = low voltage.

The control signals are such that state OFF = high voltage and state ON = low voltage.

For synchronous DTE interfaces, data on TxD are latched by the ITAC on the rising edges of S, and data on RxD are clocked off by the ITAC on the falling edges of S.

Table 3. ITAC DCE/DTE Interface Signals

Pin DCE mode	Pin DTE mode	Mnemonic	Description	X.21	V.24 / X.21 bis
I	O	TxD	Transmit Data	T	103
O	I	RxD	Receive Data	R	104
O		S	Signal Element Timing	S	114/115 (sync only)
I	O	DTR/C	Data Terminal Ready/Control	C	108
O	I	DCD/I	Data Carrier Detect/Indicate	I	109
I	O	RTS	Request to Send	–	105
O	I	CTS	Clear to Send	–	106
O	I	DSR	Data Set Ready	–	107
I	I	MI1–3	Multifunctional Inputs		programmable
O	O	MO1–2	Multifunctional Outputs		programmable

Synchronous Network Interface

The Synchronous Network Interface (SNI) consists of four physical connections (Figure 7). Two lines are for the transmission of data, one for each direction. The other two lines are for the bit and frame clocks.

The bit rate on the Serial Data Receive SDR and Serial Data Transmit SDX lines is equal to the bit clock frequency (CLK). Data is output on SDX on the rising edges of CLK and latched from SDR on the falling edges. The particular channel where data is to be received and transmitted by the ITAC is programmable. The position of the time channel is relative to the Frame Sync signal (FSC) which marks the beginning of the frame. The repetition rate of FSC is 8 kHz. The maximum bit rate is 4.096 Mbps, corresponding to a maximum of 512 bits per frame.

Programmable Clock Generator

The Programmable Clock Generator (PCG) consists of DPLL circuitry synchronized by the network reference clock (FSC) and driven by an external crystal of frequency 10.752 MHz ±100 ppm or by an external oscillator. It delivers the clocks used by the different functional blocks:

- Bit clock for the Intermediate Rate Converter: 8, 16, 32 or 64 kHz
- Frame clock for the Intermediate Rate Converter
- Network rate clock: 600 Hz to 64 kHz
- User rate clock: 600 Hz to 64 kHz
- Transmit async clock: 300 x 2^n, (n = 0,1,2,3,4,5,6,7) synchronized by the leading edge of a start bit.
- Receive async clock: 300 x 2^n Hz.

The clocks are synchronized upon the timing received from the Synchronous Network Interface, when provided. When the SNI timing is not provided, the clocks are running free if not disabled by the microcontroller. Upon application of network clocking, synchronization of the clocks is automatically performed in such a way that any phase jump does not exceed 1 of the nominal synchronous clock period.

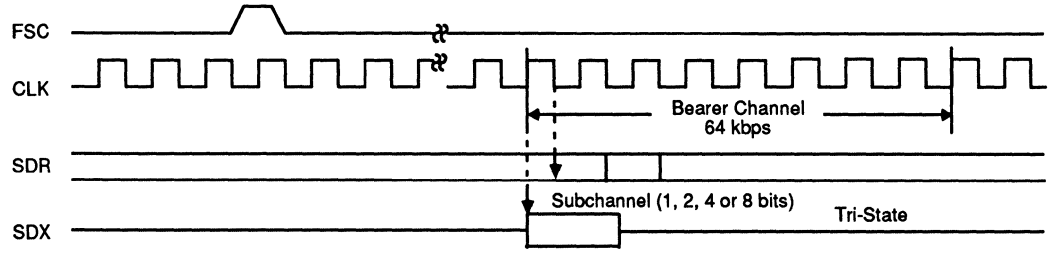


Figure 6. Timing of the Synchronous Network Interface

Table 4. Network Rate-to-Intermediate Rate Conversion Summary

Network rate bps	Intermediate rate (kbps)	Frame length (bit)	User data bit repetition factor
600	8	4 x 80	8
1200	8	2 x 80	4
2400	8	80	2
4800	8	80	1
9600	16	80	1
19200	32	80	1
38400	64	80	1
48000	64	32	1
56000	64	8/64	1
64000	64	transparent	1

Async/Sync Converter

The ASC transforms the stream of Start/Stop formatted characters into a synchronous bit stream defined by 2^n times 600 bps, and vice versa.

The possible asynchronous rates are: 300, 600, 1200, 2400, 4800, 9600, 19200 and 38400 bps.

Functions In the Transmit Direction

The async-to-sync converter transforms the asynchronous characters (User rate) into a selected synchronous Network rate. The Network rate is equal to 2^n x nominal User rate (n = 0 to 7).

- Framing upon the incoming characters from local DTE/DCE. (Baud rate factor 16)
- Handling of underspeed and overspeed:

  Transmitted characters are padded with Stop elements as often as necessary to match the Network rate. In case of overspeed, Stop elements are deleted within the limits of the selected tolerance range.
- Recognition and generation of break signals:

  A break signal is generated in accordance with V.110 when at least M-bits of Start polarity at User rate are detected.

Note: M denotes the number of bits per character in the selected format including Start and Stop bits.

Functions In the Receive Direction

The sync-to-async converter transforms the synchronous bit stream (Network rate) into an asynchronous User rate. The latter is equal to 2^n x Network rate, (n = 0 to 6) for Network rate greater than 600 bps. It is either 300 or 600 bps for Network rate equal to 600 bps.

- Framing upon the incoming characters from the Intermediate Rate Converter
- Missing Stop elements:

  Missing Stop elements are restored. If necessary, the length of Stop elements are reduced according to the selected tolerance range, to allow for overspeed on the transmitting side.
- Break signal:

  The 2M + 3 or more bits of Start polarity received from the transmitting side are output to the DCE/DTE interface.

Intermediate Rate Converter

The IRC converts

- The synchronous net data originating from the DCE/DTE interface, the Async-to-Sync Converter or the Serial Communication Logic into an intermediate rate equal to 2^n x 8 kbps.
- The intermediate rate into synchronous net data sent to the DCE/DTE interface, the Sync-to-Async Converter or the Serial Communication Logic.

  The intermediate rate and the frame are implicitly determined by the Network rate, as shown in Table 4.

  For rates up to 38400 bps, the frame is based on the 80-bit frame structure shown in Table 5.

Table 5. 80-bit Frame Structure (X.30/V.110)

Bit number	Bit position 1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
1	1	D1	D2	D3	D4	D5	D6	Si
2	1	D7	D8	D9	D10	D11	D12	X
3	1	D13	D14	D15	D16	D17	D18	Sj
4	1	D19	D20	D21	D22	D23	D24	Sk
5	1	E1	E2	E3	E4	E5	E6	E7
6	1	D25	D26	D27	D28	D29	D30	Sl
7	1	D31	D32	D33	D34	D35	D36	X
8	1	D37	D38	D39	D40	D41	D42	Sm
9	1	D43	D44	D45	D46	D47	D48	Sn

The adaptation of 48 and 56 kbps rates is based on a 32- and a 64-bit frame, respectively, according to V.110.

Functions In the Receive Direction

- Frame recovery

  The frame alignment pattern is composed of eight zeros and nine ones. In addition, bit E7 is used for superframe alignment in the 600 bps case.

  The receiver indicates its state of synchronism to the microcontroller via a maskable interrupt status.
- Extraction of D bits
- Extraction of status bits

  These bits are accessible to the microcontroller. Additionally, they can be directly mapped on to the DCE/DTE interface in synchronism with the D-bits according to X.30 or V.110. If an X.21 interface is selected, the SP, SQ and SR are mapped on I interchange circuit. Otherwise, SA (S1,S3, S6, S8) and SB (S4, S9) are mapped as shown in Figure 7.
- Extraction of E and X bits

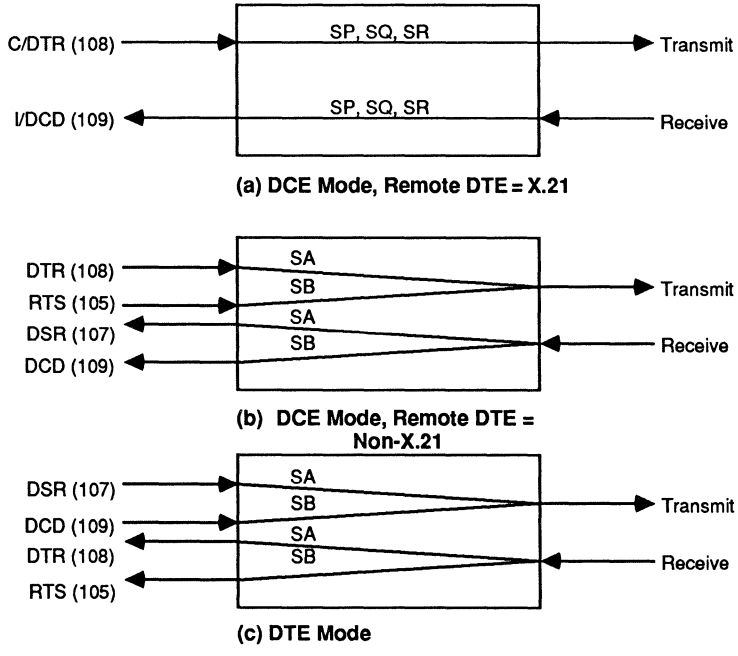


Figure 7. Status Bit Mapping Summary

These bits are accessible to the microcontroller. A change is reported via a maskable interrupt status.

Functions in the Transmit Direction

- Frame setup
- Insertion of D-bits
- Insertion of status bits

  These bits either originate from the microcontroller or they convey the state of DTE interchange circuits at specific sampling points according to recommendation X.30 or V.110. In the latter case, if an X.21 interface is programmed, the SP, SQ and SR bits reflect the state of C interchange circuit. Otherwise, SA (S1, S3, S6, S8) and SB (S4, S9) are mapped as shown in Figure NO TAG.
- Insertion of E and X bits.

  These bits originate from the microcontroller.

Bearer Rate Converter

The BRC takes the intermediate rate data to groups of 1, 2 or 4 bits, to be sent as a subchannel in a 64 kbps bearer channel, and vice versa. The BRC is not used in the case of programmed network rate greater than or equal to 38400 bps.

Status Detect Logic

The ITAC performs detection of local DCE/DTE interchange circuit states for V.24 interfaces. The local DCE/DTE states monitored by the Status Detect Logic are DTR (108), RTS (105) and MI1–3 (DSR, DCD and MI1–3 in DTE mode). Changes of state are reported to the Microcontroller Interface Logic by a maskable interrupt status.

In the case of a local X.21 DTE, combined local (TxD, C) status recognition is performed by the Status Detect Logic. Valid X.21 states are reported to the Microcontroller Interface Logic after 16 bit times over a status bit. The valid states are shown in Table 6.

When the local DTE is synchronous and the local DTE or the remote DTE is of the X.21 type, combined remote (TxD, C) status recognition is performed as also shown in Table 6.

Table 6. Valid X.21 States

Local DTE status (TxD, C) local	Remote DTE status (TxD, C) remote
(1, ON)	(1, ON)
(0, ON)	
(1, OFF)	(1, OFF)
(0, OFF)	(0, OFF)
(0011, OFF)	(0011, OFF)
(01, OFF)	
(00001111, OFF)	

Hayes Protocol Support

The Hayes (Smartmodem) Protocol is supported by the integrated USART plus additional hardware.

In the "on-line mode," the ITAC behaves as a conventional modem, treating the data transparently as explained in the *Transparent Mode* section on page 10. To implement the Hayes modem command mode, the USART with its associated FIFOs is switched into the data path for the exchange of control character sequences with the local DTE.

In the Hayes Smartmodem protocol, the return to the command mode from the on-line mode is possible without tearing down the data call. The return is effected after the reception of escape characters.

The ITAC implements this procedure by recognizing a local character(s) from the local DTE. In addition, the "guard time" specified in the Hayes protocol can be supervised via a status bit.

Flow Control Support

A flow control option, for use with TAs supporting asynchronous DTEs, is offered by the ITAC. Flow control allows the connection of asynchronous DTEs operating at different user data rates by reducing the character output of the faster to that of the slower DTE.

The connection of two DTEs operating at unequal rates requires that the selected Network rate is different from the User rate for one of the ITACs.

If the Network rate is lower than the User rate, external buffering of the characters received from the local DTE is implemented by switching the USART in the transmit data path. The receive data path requires no buffering, and the speed conversion is automatically handled by the ITAC (Figure 8a).

If, on the other hand, the User rate is lower than the Network rate, external buffering of the characters received from the network is implemented by switching the USART in the receive data path. The transmit data path is transparent (Figure 8b).

For use in a terminal adapter connected to the faster DTE (Figure 9a and b), the ITAC supports in-band and out-band local flow control procedures. In-band flow control is implemented by outputting a special XON/XOFF character to the local DTE. The character is inserted in the input data stream upon a microcontroller command without loss of receive characters. Out-band flow control is implemented by the microcontroller directly controlling an output line, CTS.

Remote flow control is required in the case where the Network rate of one of the terminal adapters is higher than the selected User rate (Figure 9b). The X-bit is used to carry flow control information. The microcontroller is able to directly set the X-bit according to the

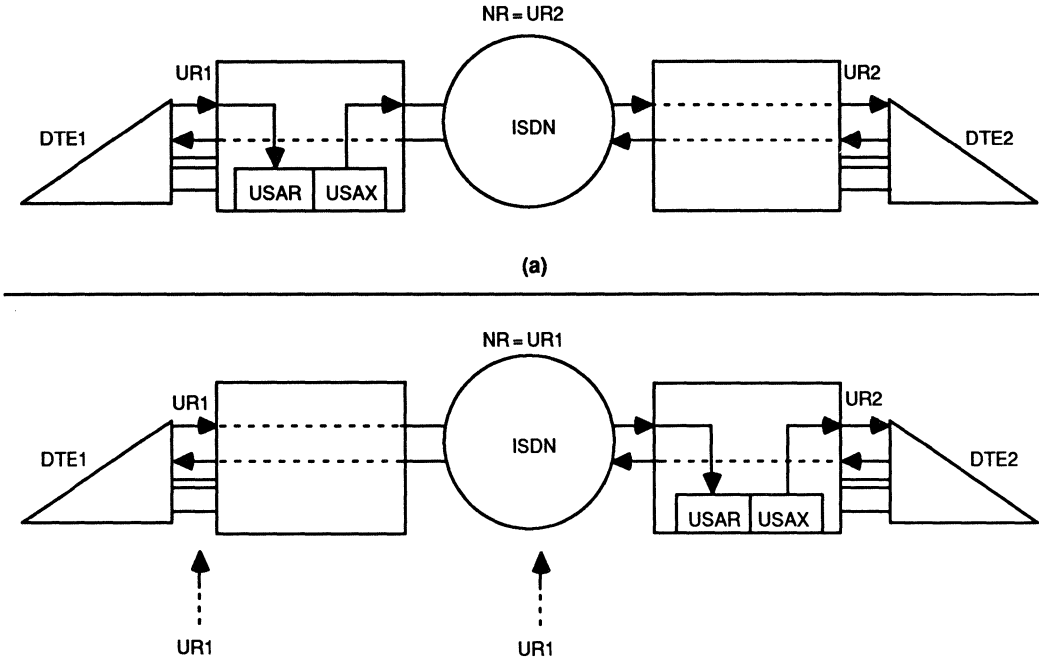


Figure 8. Local and Remote Flow Control Situations. It is assumed that DTE1 is always the faster DTE.

state of the character buffer on the transmitting side. On the receiving side, the ITAC reports any changes in the X-bit by a maskable interrupt.

For the case where a DTE itself transmits flow control characters to slow down the remote DTE, the ITAC provides for recognition and inhibiting of two programmable characters from the DCE/DTE interface. Finally, to support the connection of terminal adapters employing in-band remote flow control, the ITAC is also able to recognize two programmable characters from the network.

Serial Communication Logic

The Serial Communication Logic consists of two independent controllers, the USART and the HDLC controller, with their associated FIFOs.

FIFO Operation

The length of each of the Receive FIFOs (RFIFO) and Transmit FIFOs (XFIFO) is nine characters/bytes.

The first bit of a character/byte received or transmitted is the rightmost stored bit or LSB. Characters are right-justified.

Interrupt status bits indicate the state of RFIFO and XFIFO. Data can be loaded and fetched by the microcontroller, or using DMA. However, even in the DMA case the FIFOs are used. This gives a worst case reaction time of 1 ms in both non-DMA and DMA operation (worst case of 64 kbps), to maintain a constant data throughput.

In the HDLC case several frames can be simultaneously stored in the RFIFO and the XFIFO.

USART

In its asynchronous mode, the USART has the following programmable settings (Table 7).

Table 7. Programmable Settings of the USART

Parameter	Values
Character length excluding possible parity bit	5, 6, 7, 8
Parity check/generation	yes, no
Parity type	even, odd, 0,1
Number of stop bits	1, 2

When enabled, the receiver searches for valid character frames (Start + character + Stop) and stores the charac-

ters in the RFIFO as they arrive, stripping off the start and stop bit(s).

A possible parity bit is stored in the RFIFO as the bit immediately on the left of the MSB. Thus the parity is transferred to the microprocessor, an exception being: number of character data bits = 8 plus parity.

In the case where a parity error is detected (and parity check/generation is programmed), this fact is indicated with a status bit at the time the microcontroller reads the corresponding character from the RFIFO. Similarly, a framing error status is generated at the time when the illegal character is read from the RFIFO.

A break signal is received if a row of null bits of length > M − 1 is detected from the DCE/DTE interface, or of length > 2M − 2 is detected from the network. A break signal is reported by an interrupt status.

When the asynchronous transmitter is enabled, the transmission of any characters entered in the XFIFO is immediately started. Characters are separated by the selected number of Stop bits. Characters may be entered in the XFIFO even if the async transmitter is not enabled. When the XFIFO runs empty, continuous "1" is transmitted until a new character is available.

Break signals of any length may be transmitted.

In the synchronous mode of the USART, the characters received and transmitted have a length of eight bits, including possible parity.

The sync receiver has two modes of operation: Transparent mode and Hunt mode. In the Transparent mode the receiver, when enabled, immediately starts storing received characters without regard for octet synchronization. In the Hunt mode, the receiver searches for one (monosync) or two (bisync) programmable characters before starting to store the received characters. The Hunt mode and the Transparent mode may be entered at any time by issuing the corresponding command.

If desired, a programmable parity check is performed on the receive characters, as in the asynchronous case.

The operation of the transmitter is similar to the operation in the async mode. However, no parity is generated. Further, in the synchronous mode, the last character is repeatedly transmitted when the XFIFO runs empty, as opposed to constant "1" in the async mode.

HDLC Controller

The functions performed by the HDLC receiver are:

- flag detection
- zero deletion
- CRC checking (generator polynomial: $x16 + x12 + x5 + 1$)
- check for abort
- check for idle

When enabled, the receiver enters a hunt phase and remains in the hunt phase until it detects a valid opening flag.

Once a flag is recognized by the receiver, not followed by another flag or by an abort or idle sequence, all the subsequent eight-bit bytes are stored in the RFIFO, up to and excluding the Frame Check Sequence (FCS) that immediately precedes the closing flag. When the closing flag is detected, a "receive status byte" is appended to the stored frame. This status byte contains status information pertaining to the received frame: frame aborted yes/no, CRC error yes/no data overflow yes/no, and number of significant bits in the last receive byte.

As the receive status byte is read, an interrupt status is generated to indicate the end of the receive frame.

When the beginning of a receive frame is detected and the RFIFO is full, the frame will be completely lost and an interrupt status is generated.

The functions performed by the HDLC transmitter are:

- flag generation
- zero insertion
- CRC generation
- abort sequence generation
- interframe time fill generation

When the transmitter is enabled, the transmission of any bytes entered in the XFIFO is immediately started, preceded by an opening flag.

To indicate the end of frame, the microcontroller sets a control bit after having entered the last data byte of that frame in the XFIFO. Every write operation of the bit serves as a frame delimiter.

As long as frames are entered in the XFIFO, they are transmitted in the order in which they were entered.

When no data is available in the XFIFO and the transmitter is enabled, interframe time fill (either non-shared flags or continuous ones), is transmitted.

When the XFIFO becomes empty and a frame end indication has not been issued, an error interrupt status is generated. An abort sequence (instead of FCS plus flag) is automatically transmitted following the last byte entered.

Maintenance Functions

Two test loops are provided in the ITAC. The "remote" test loop is located close to the DCE/DTE interface. The "local" test loop is located close to the Synchronous Network Interface (Figure 9).

In the remote loop mode, communication between the local DCE/DTE and Serial Communication Logic is possible.

In the local loop mode "all ones" or tri-state is transmitted to the Synchronous Network Interface.

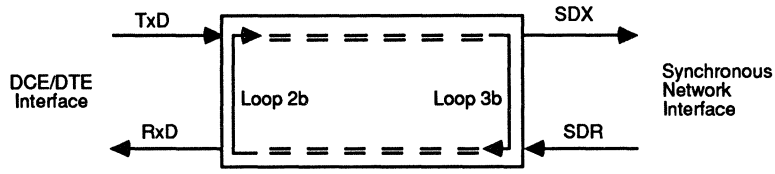


Figure 9. ITAC Test Loops

OPERATIONAL DESCRIPTION

Register Set

The communication between the microcontroller and the ITAC is done via a set of directly accessible 8-bit registers.

The register set can be divided into:

- interrupt status registers and status registers for supervising the operation of the circuit, and registers for data transfer to the microcontroller;
- control registers for configuring the ITAC according to the different phases of a data call, registers for parameter selection, command registers and registers for data transfer from the microcontroller.

Interrupt Structure

Since the ITAC provides only one interrupt request output, the cause of an interrupt is determined by the microcontroller by reading the Interrupt Status Register IST. The bits in IST point to status registers, as shown in Figure 10.

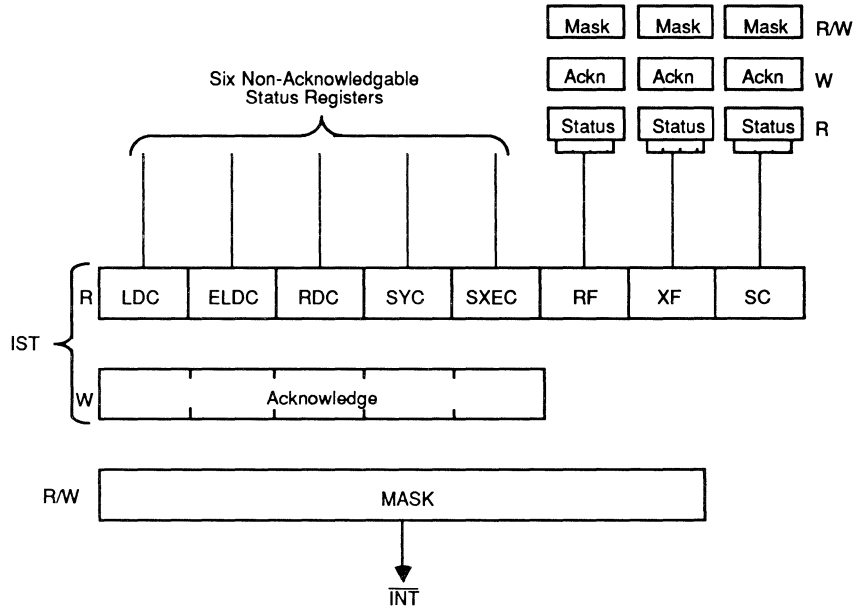


Figure 10. Interrupt/Status Register Structure

A "1" in any of the IST register bits causes the $\overline{INT}$ line to be activated.

$\overline{INT}$ is deactivated when all the bits set in IST are acknowledged by the microcontroller, or reset internally by the ITAC.

The IST register bits can be selectively masked (MASK). This has no effect on the IST register bits, but a masked interrupt status bit in IST does not cause the $\overline{INT}$ line to be activated.

Status Registers

The status registers are listed in Table 8.

Six of the registers contain information concerning the state of the communicating DTEs, the state of synchronization of the ITAC, and S, X and E bit status. A change in any of the registers causes an interrupt status bit in IST to be set.

The other three registers include the state of the Serial Communication Logic and information about special events. A "1" in one of the bits in these registers causes the interrupt status bit in IST to be set. An active status is reset when the condition that caused it is no longer true, or when acknowledged by the microcontroller. The registers have an individual mask register. An active mask bit prevents the bit in IST from being set, but the status can be polled in the originating register.

Acknowledgment is performed by writing a "1" in the position of the status bit.

In addition to the eight status registers which are part of the interrupt logic, one register is provided which is polled by the microcontroller but which may not cause interrupts (STR).

Table 8. Interrupts and Status Registers

Interrupt status	Source of Interrupt Status	Maskable status	Interrupt/Status acknowledged through
LDC	Change in Local DTE Status (LDS	No	IST
ELDC	Change in Local DTE Status (ELDS)	No	IST
RDC	Change in Remote DTE Status (RDS)	No	IST
SYC	Change in Synchronization Status (SYS)	No	IST
SXEC	Change in S, X, or E bits (SXS, ES)	No	ST
RF	Receiver and RFIFO Status (RFS):	Yes	RFS
	HDLC Receive Data USART Receive Data HDLC Receive FIFO Full USART Receive FIFO Full Receive Message End Reception Receive Message End Framing Error Parity Error		
XF	Transmitter and XFIFO Status (XFS):	Yes	XFS
	HDLC Transmit FIFO Write Enable USART Transmit FIFO Write Enable HDLC Transmit FIFO Empty USART Transmit FIFO Empty HDLC Transmit Data Underrun		
SC	Special Condition Status (SCS):	Yes	SCS
	HDLC Receive FIFO Overflow USART Receive FIFO Overflow Break Begin Break End Local Character 1 Recognized Local Character 2 Recognized Remote Character 1 Recognized Remote Character 2 Recognized		
	Additional status information:		
	Status Register (STR)		STR

Write Registers

Table 9. Write Registers

Mask Registers		Write/Read
MASK	Interrupt Status Mask Register	W/R
RFIM	Receiver & RFIFO Interrupt Status Mask Register	W/R
XFIM	Transmitter & XFIFO Interrupt Status Mask Register	W/R
SCIM	Special Condition Interrupt Status Mask Register	W/R
Configuration Registers		
GCR	General Configuration Register	W/R
SCR	Special Configuration Register	W/R
AICR	Asynchronous Interface Configuration Register	W/R
DPCR	Data Path Configuration Register	W/R
HMR	HDLC Mode Register	W/R
UMR	USART Mode Register	W/R
Operational Parameter Registers		
BRS	Bit Rate Select	W/R
TSR	Time Slot Register	W/R
NRF	Number of Retry Frames	W/R
Control Registers		
LDR	Local DTE Register	W/R
RDR	Remote DTE Register	W/R
XER	Transmitted E-bits	W/R
Command Registers		
HCC	HDLC Controller Commands	W
UCC	USART Controller Commands	W
INSC	Character Insert	W/R
Constants & Special Character Registers		
SYN	Synchronization character	W/R
LCAR1	Recognizable Character 1 from Local DTE/DCE	W/R
LCAR2	Recognizable Character 2 from Local DTE/DCE	W/R
RCAR1	Recognizable Character 1 from Remote TA/DTE	W/R
RCAR2	Recognizable Character 2 from Remote TA/DTE	W/R
Other Registers		
TEST	Test Register	W

Reset State

The ITAC is in the reset state after power-on or after the application of a reset pulse on $\overline{RST}$.

In the reset state, all control registers are zeroed and all interrupts are masked. Synchronous Network Interface and DCE/DTE interface outputs are tri-state.

Power-Down State

The ITAC may be set in a standby state to save unnecessary power consumption when idle. For switching the ITAC between standby and operational states two cases are distinguished, according to whether the DCE/DTE interface is an X.21 interface, or a V.24 interface.

The characteristics of the standby state are shown in Table 10.

Table 10. Characteristics of the Standby State

Output or function	X.21 (V24 = 0)	V.24 (V24 = 1)
Microcontroller Interface	Operational	Operational
Oscillator	Disabled	Enabled
Other logic	Disabled	DTR detection
RxD	Previous state	Previous state
S	Logical 1 or tri-state	Logical 1 or tri-state
Other DCE/DTE outputs	Previous state	Previous state
SDX	Tri-state	Tri-state

For an X.21 interface, switching between "standby" and "operational" is only subject to the state of the control bit PU (Power Up). The oscillator reaches a steady state within 10 ms after the PU bit is set to one.

In the V.24 case, standby is reached when PU is set to zero and DTR(108) interchange circuit is OFF. When DTR is switched ON, the standby state is left, an interrupt is generated, and the PU bit is set to one by internal logic. (Figure 11).

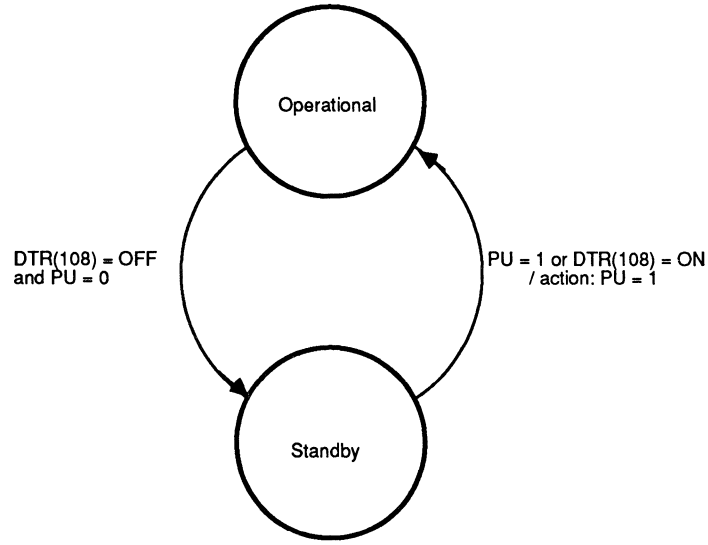


Figure 11. Standby State Diagram (V24 = 1)

Initialization

After reset, the user has to write a minimum number of registers to set the ITAC into an operational state. The most important among these are:

- General Configuration Register (GCR)
- Bit Rate Select Register (BRS)
- Data Path Configuration Register (DPCR)
- Local DCE/DTE Interface Register (LDR).

Interrupts are enabled via the MASK register.

The microcontroller may switch the ITAC into operational state at any time, not necessarily at initialization—unless a clock is required by the local DTE—but, say, upon detecting an incoming data call. Also the automatic wake-up feature can be used in the case of an outgoing call. Note that switching between "standby" and "operational" does not affect the contents of the registers.

Processing

If the ITAC is run in transparent mode, the microcontroller has only a supervisory function. Special events, such as loss of synchronization, or changes in the state of the local or the remote DTE can be monitored via the interrupt logic, or by polling.

In the non-transparent mode, the Serial Communication Logic is activated. Data transfer itself can be interrupt, polling or DMA-driven.

Interrupt Operations

If neither the USART nor the HDLC controller is used in DMA, the non-DMA mode is selected (DMA = 0).

Reception

The reception of data is supervised via the Receiver and RFIFO Status register (RFS).

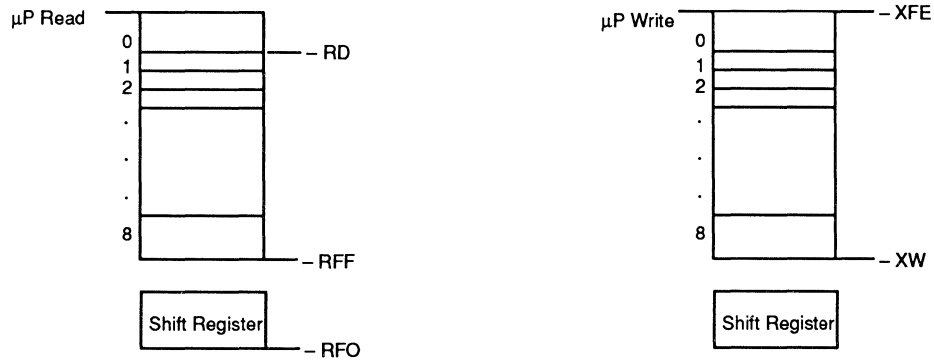
USART

The USART Receive Data (URD) maskable interrupt status indicates that at least one character is stored in the Receive FIFO. The URD status is reset when the RFIFO is empty.

The reaction time of the microcontroller to a URD interrupt status is, in the worst case, 1 ms (64 kbps synchronous data transfer).

The receive FIFO Full (URFF) interrupt status indicates that the Receive FIFO contains nine characters. This interrupt should be serviced with a high priority, otherwise a Receive FIFO Overflow (URFO) interrupt status might occur.

The Framing Error and Parity Error bits (FER, PER) may also be enabled to generate an interrupt, or, else, can be polled in RFS. These bits are updated every time a character is read from RFIFO.



a. Receive FIFO

b. Transmit FIFO

Figure 12. FIFO Status Indications

HDLC

Data transfer over the HDLC receive FIFO is similar to the USART case. The interrupt status HDLC Receive Data (HRD) indicates that at least one byte of data is stored in RFIFO. The HDLC Receive FIFO Full (HRFF) interrupt status indicates that nine bytes are stored. Loss of data is indicated by:

- HDLC Receive FIFO Overflow (HRFO) interrupt status if the first byte of an HDLC frame could not be stored because of a full RFIFO, and consequently one or more HDLC frames are lost;
- Otherwise the loss of data is indicated by the RDO (Receive Data Overflow) bit in the status byte of the corresponding frame.

The end of a frame is indicated by the Receive Message End (RME) bit in RFS. This status is updated every time a byte is read from RFIFO, and may also be enabled as an interrupt status.

Transmission

The transmission of data is supervised via the Transmitter and XFIFO Status register (XFS).

USART

The USART Transmit FIFO Write enable (UXW) maskable interrupt status indicates that one or more characters may be written in the Transmit FIFO. The UXW status is reset when the XFIFO is full.

When all characters in the XFIFO have been read, a Transmit FIFO Empty (UXFE) interrupt status is generated. When the XFIFO becomes empty, "1" (async), or

the last entered character (sync) is continuously transmitted.

HDLC

Data transmission over the HDLC XFIFO is similar to the USART case. In addition to HXW and HXFE a HDLC Transmit Data Underrun (HXDU) interrupt status bit is provided, which is activated when the last byte in XFIFO has been transmitted and no XME command has been issued. The frame is closed with an abort sequence. If the frame is not yet complete, the HXFE interrupt should be serviced with high priority to prevent HXDU from occurring.

Polling Operations

For polling driven operations non-DMA mode is programmed in the General Configuration Register (DMA = 0). Interrupts are disabled by masking in the MASK register. The Interrupt Status Register, however, can be polled and the status bits may be acknowledged as usual. Events can also be masked at the status register level, in which case they do not cause bits to be set in the Interrupt Status Register.

DMA Operations

USART

DMA mode (DMA = 1) with USART (DMH = 0) is programmed in the General Configuration Register.

Receiver

The USART Receive FIFO Full (URFF) interrupt status indicates that a DMA output request (DMOR) is pending and the RFIFO is full. It serves as a request for the microcontroller to program the DMA controller. The URFF status is withdrawn when less than nine characters are present in the ITAC.

Up to nine characters may be stored inside the ITAC before loss of data. The DMOR request is automatically generated until all characters have been read.

Generation of DMOR request is blocked when a PER or an FER interrupt status occurs (if enabled in RFIM). Generation of DMOR can be resumed when the PER/FER interrupt status is acknowledged by the microcontroller.

Loss of data is indicated by the URFO interrupt.

Transmitter

The DMA input request (DMIR) is initially generated by the ITAC when the DMA Start Command (UDMS) is issued. A DMIR request is generated as long as space is left in the XFIFO. If the DMA controller does not respond to the request before the last character is read from the XFIFO, a Transmit FIFO Empty (UXFE) interrupt is generated and DMIR is withdrawn. After a UXFE interrupt, DMA transfer may be started anew by reissuing a UDMS command.

HDLC

DMA mode (DMA = 1) with HDLC (DMH = 1) is programmed in the General Configuration Register.

Receiver

The HDLC Receive FIFO Full (HRFF) interrupt status indicates that a DMA output request (DMOR) is pending and the RFIFO is full. It serves as a request for the microcontroller to program the DMA controller. The HRFF status is withdrawn when less than nine bytes are present in the ITAC.

A Receive Message End Received (RMER) interrupt can be also used as a request to program the DMA controller. It indicates that the end of an HDLC frame has been received.

Up to nine bytes may be stored inside the ITAC before loss of data. The DMOR request is generated until all bytes have been read.

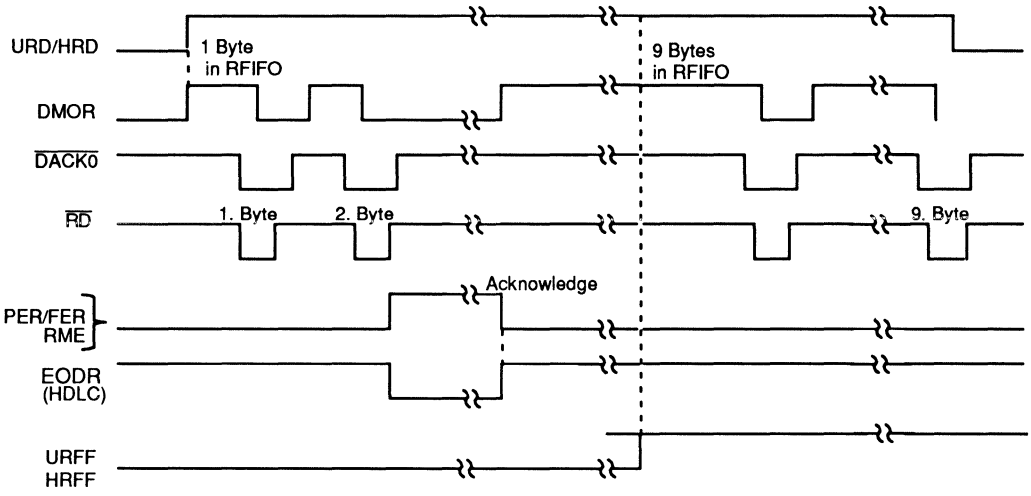
When the status byte pertaining to a received HDLC frame is read, a Receive Message End (RME) interrupt status is generated. Simultaneously, EODR output is activated. The status bit remains set until acknowledged by the microcontroller. Generation of DMOR requests is blocked until RME has been acknowledged, and restarted thereafter. If the microcontroller does not acknowledge the RME status before a RFIFO overflow occurs, then:

- If the first byte of a receive frame could not be stored, a HDLC Receive FIFO Overflow (HRFO) is generated;
- Otherwise, the overflow is indicated in the status byte of the corresponding frame.

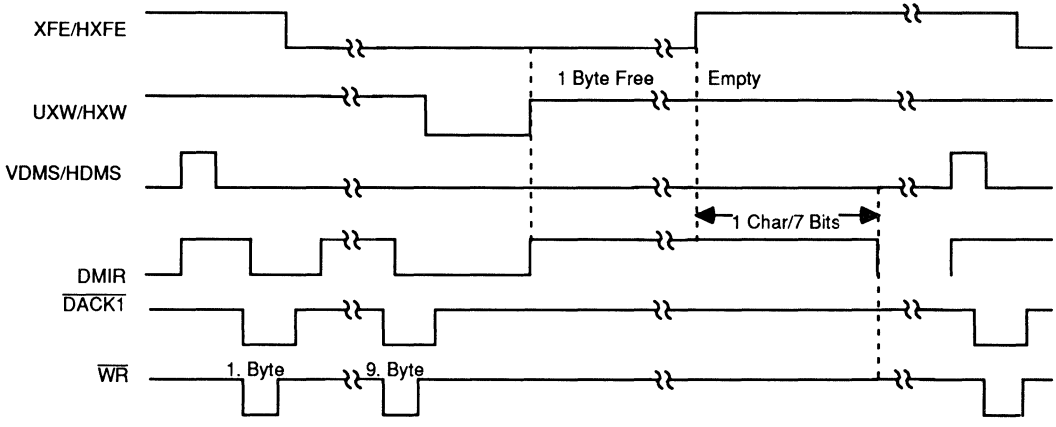
The EODR output is deactivated when RME is acknowledged.

Transmitter

The DMA input request (DMIR) is initially generated by the ITAC when the DMA Start Command (HDMS) is issued. When a byte has been fetched for transmission, a new request is generated. If the DMA controller does not respond to the request before the XFIFO is empty, a Transmit FIFO Empty (HXFE) interrupt is generated, DMIR is withdrawn and the frame is automatically closed by appending a FCS and a closing flag. DMA transfer may be started anew by reissuing a HDMS command.



a. Receive



b. Transmit

Figure 13. DMA Operation

DETAILED REGISTER DESCRIPTION

Register Space

In order to facilitate a direct connection to 16-bit processors, it is possible to access all ITAC registers using either even microprocessor addresses only or odd microprocessor addresses only.

The register address map is shown in Figure 14.

Interrupt/Status Registers

Interrupt Status Register (IST)

LDC	ELDC	RDC	SYC	SXEC	RF	XF	SC

LDC Local DTE Status Change (status register LDS)

ELDC Extended Local DTE Status Change (status register ELDS)

RDC Remote DTE Status Change (status register RDS)

SYC Synchronization Status Change (status register SYS)

SXEC S, X or E bit Status Change (status register SXS, ES)

RF Receiver and RFIFO Status (status register RFS)

XF Transmitter and XFIFO Status (status register XFS)

SC Special Condition Status (status register SCS)

Local DTE Status (LDS)

Non-X.21 Mode (V24 = 1)

DTR	RTS	MI1	MI2	MI3	X	X	X

DTR Data Terminal Ready interchange circuit state

RTS Request To Send interchange circuit state

MI1–3 Multifunctional input states

X.21 Mode (V24=0)

LONO	LOFO1	LL3	X	X	X	X	X

LON0 Local (0, ON) state

LOF01 Local (0101..., OFF) state

LL3 Local Loop 3. Local (00001111..., OFF) state

Extended Local DTE Status (ELDS)

X.21 mode only (V24 = 0).

LONX	LOFX	LON1	LOF1	LOF0	LL2	X	X

Address	Read	Write	
00–0F	RFIFO (HDLC)	XFIFO (HDLC)	
10–1F	RFIFO (USART)	XFIFO (USART)	
20,21	IST	ISTA	Interrupt Status Register
22,23			
24,25			
26,27			
28,29			
2A,2B			
2C,2D			
2E,2F		TEST	
30,31	LDS		
32,33	ELDS		
34,35	RDS		
36,37	SYS		
38,39	SXS		
3A,3B	ES		Status Registers
3C,3D	RFS	RFSA	
3E,3F	XFS	XFSA	
40,41	SCS	SCSA	
42,43			
44,45	STR	STRA	
46,47			
48,49	MASK	MASK	
4A,4B	RFIM	RFIM	Mask Registers
4C,4D	XFIM	XFIM	
4E,4F	SCIM	SCIM	
50,51	GCR	GCR	
52,53	SCR	SCR	
54,55	AICR	AICR	
56,57	DPCR	DPCR	Configuration Registers
58,59	HMR	HMR	
5A,5B	UMR	UMR	
5C,5D			
5E,5F	BRS	BRS	
60,61	TSR	TSR	Operational Parameter Registers
62,63	NRF	NRF	
64,65	LDR	LDR	
66,67	RDR	RDR	Control Registers
68,69	XER	XER	
6A,6B			
6C,6D		HCC	
6E,6F		UCC	Command Registers
70,71	INSC	INSC	
72,73			
74,75			
76,77	SYN	SYN	
78,79	LCAR1	LCAR1	
7A,7B	LCAR2	LCAR2	Constants and
7C,7D	RCAR1	RCAR1	Special Character Registers
7E,7F	RCAR2	RCAR2	

Figure 14. ITAC Register Map

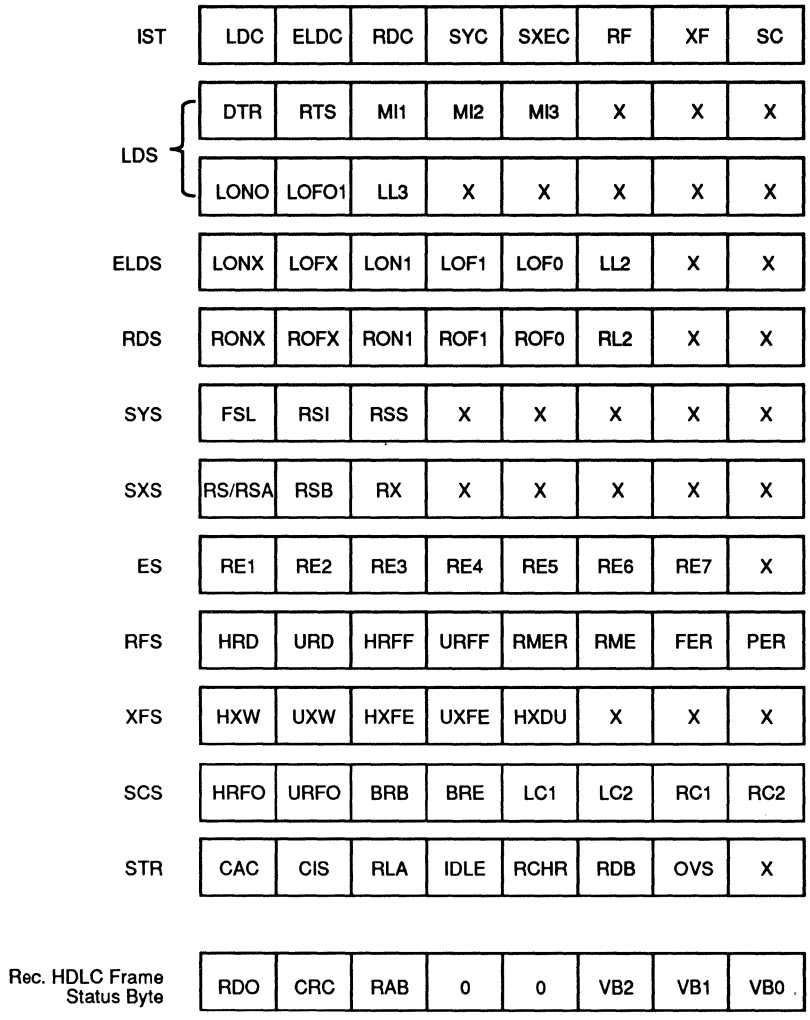


Figure 15. ITAC Status Register Summary

Extended Local DTE Status (ELDS)

X.21 mode only (V24 = 0).

LONX	LOFX	LON1	LOF1	LOF0	LL2	X	X

LONX Local (X, ON) state

LOFX Local (X, OFF) state

LON1 Local (1, ON) state

LOF1 Local (1, OFF) state

LOF0 Local (0, OFF) state

LL2 Local Loop 2. Local (0011. . . , OFF) state

Remote DTE Status (RDS)

RONX	ROFX	RON1	ROF1	ROF0	RL2	X	X

RONX Remote (X, ON) state

ROFX Remote (X, OFF) state

RON1 Remote (1, ON) state

ROF1 Remote (1, OFF) state

ROF0 Remote (0, OFF) state

RL2 Remote Loop 2. Remote (0011. . . , OFF) state

Synchronization Status (SYS)

FSL	RSI	RSS	X	X	X	X	X

FSL Frame Sync Loss of the intermediate rate receiver. At least three consecutive erroneous frames have been received.

RSI Resynchronization Impossible. The intermediate rate receiver has not achieved synchronization.

RSS Resynchronization Successful. The intermediate rate receiver is synchronous.

S- and X-Bit Status (SXS)

RS/RSA	RSB	RX	X	X	X	X	X

RS/RSA Received S (V110 = 0) or SA (V110 = 1) bit

RSB Received SB bit (V110 = 1)

RX Received X bit

E-Bit Status (ES)

RE1	RE2	RE3	RE4	RE5	RE6	RE7	X

RE1–7 Received E-bits

Receiver and RFIFO Status (RFS)

HRD	URD	HRFF	URFF	RMER	RME	FER	PER

HRD HDLC Receive Data in RFIFO

URD USART Receive Data in RFIFO

HRFF HDLC Receive FIFO Full

URFF USART Receive FIFO Full

RMER Receive Message End Reception

RME Receive Message End

FER Framing Error

PER Parity Error

Transmitter and XFIFO Status (XFS)

HXW	UXW	HXFE	UXFE	HXDU	X	X	X

HXW HDLC Transmit FIFO Write enable

UXW USART Transmit FIFO Write enable

HXFE HDLC Transmit FIFO Empty

UXFE USART Transmit FIFO Empty

HXDU HDLC Transmit Data Underrun

Special Condition Status (SCS)

HRFO	URFO	BRB	BRE	LC1	LC2	RC1	RC2

HRFO HDLC Receive FIFO Overflow

URFO USART Receive FIFO Overflow

BRB Break Signal Begin

BRE Break Signal End

LC1 LCAR1 Character recognized from local DTE

LC2 LCAR2 Character recognized from local DTE

RC1 RCAR1 Character recognized from remote DTE/TA

RC2 RCAR2 Character recognized from remote DTE/TA

Status Register (STR)

CAC	CIS	RLA	IDLE	RCHR	RDB	OVS	X

CAC Command Accepted. Polling bit to monitor the execution of HRR, HXR, URR, UXR, SBK, HNT and TRA command

CIS Character Insertion Status. A character can be inserted in the receive character stream using INSC (1)

RLA Receive Line Active. HDLC flags/messages are being received (1) or not (0)

IDLE Idle state on HDLC receive line. A row of at least fifteen ones has been detected since the last acknowledgment (1)

RCHR Receive Characters detected. Only stop bits have been received from the DCE/DTE interface since the last acknowledgment (0)

RDB Receive Data Byte. Set after every eighth bit in X.30/V.110 frame (P8,Q8,R8 or D8,D16,D24. . .)

OVS Overspeed. The local DTE/DCE is transmitting characters at a rate exceeding the tolerance range

Receive HDLC Frame Status Byte

RDO	CRC	RAB	0	0	VB2	VB1	VB0

RDO Receive Data Overflow

CRC CRC Check Correct (1)

RAB Receive Abort

VB2–0 Valid bit count in the last byte received

Write Register

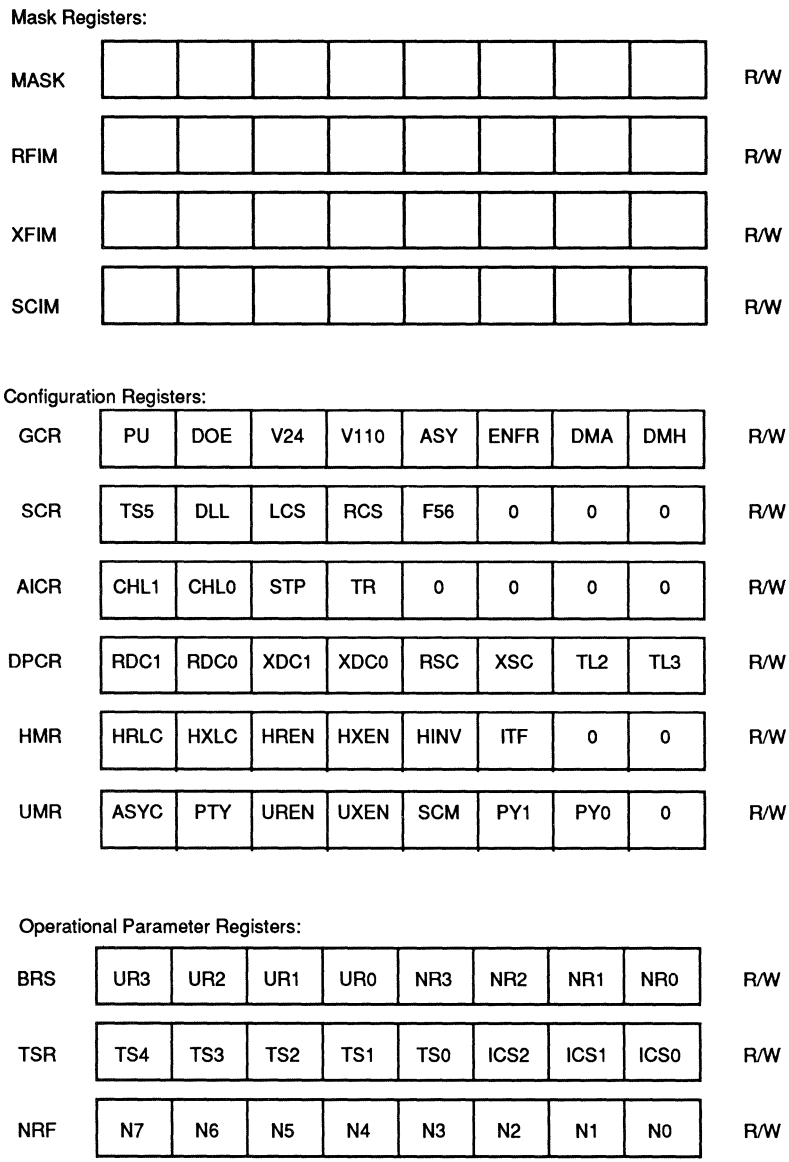


Figure 16. ITAC Control Register Summary

Control Registers:

LDR	DCD/I	DSR	CTS	RD	MO1	MO2	0	0	R/W
RDR	XS/XSA	XSB	XX	XD	0	0	0	0	R/W
XER	XE1	XE2	XE3	XE4	XE5	XE6	XE7	0	R/W

Command Registers

HCC	HRR	HXR	XME	HDMS	0	0	0	0	W
UCC	URR	UXR	0	UDMS	SBI	HNT	TRA	0	W
INSC									R/W

Constants and Special Character Registers

SUM									R/W
LCAR1									R/W
LCAR2									R/W
RCAR1									R/W
RCAR2									R/W

Other Registers

TEST									W

Figure 16. ITAC Control Register Summary (continued)

Mask Registers

Interrupt Status Mask Register (MASK) R/W

A zero in a bit of MASK inhibits the interrupt from the corresponding bit position of the IST register. However the bit in IST will still indicate the status and thus masked interrupts may be polled.

Receiver & RFIFO Interrupt Status Mask Register (RFIM) R/W

A zero in a bit of RFIM inhibits the status in IST being activated by the corresponding bit in RFS. However the bit in RFS will still indicate the status and thus may be polled.

Transmitter & XFIFO Interrupt Status Mask Register (XFIM) R/W

A zero in a bit of XFIM inhibits the status in IST being activated by the corresponding bit in XFS. However the bit in XFS will still indicate the status and thus may be polled.

Special Condition Interrupt Status Mask Register (SCIM) R/W

A zero in a bit of SCIM inhibits the status in IST being activated by the corresponding bit in SCS. However, the bit in SCS will still indicate the status and thus may be polled.

Configuration Registers

General Configuration Register (GCR) R/W

PU	DOE	V24	V110	ASY	ENFR	DMA	DMH

PU Power-up. Forces the ITAC into power-up

DOE DCE/DTE Interface Output Enable

V24 X.21 (0) or non-X.21 (1) interface

V110 Protocol according to X.30 (0) or V.110 (1)

ASY Async (1) or Sync (0) DCE/DTE interface

ENFR Enable Frame. When 0, the Synchronous Network Interface output is inactive and the receiver is idle

DMA DMA mode (1) or not (0)

DMH DMA mode for HDLC controller (1) or for USART (0)

Special Configuration Register (SCR) R/W

TS5	DLL	LCS	RCS	F56	0	0	0

TS5 Time Slot Select MSB. Used only for Synchronous Network Interface data rates greater than 2.048 Mbps

DLL Double Last Look On. Receive status bit changes are mapped on DCE/DTE interface using double last look logic (1)

LCS Local Character Stop. Any character from DCE/DTE interface that matches LCAR1 or LCAR2 is stopped (1). Used for XON/XOFF flow control and Hayes protocol

RCS Remote Character Stop. Any character from remote DTE/TA that matches RCAR1 or RCAR2 is stopped (1).Used for XON/XOFF flow control

F56 Frame for 56 kbps. Bit 8 of a 64 kbps channel is filled with "1" (0), or with the pattern "0X S3 S4 1 1 1 1" (1)

Async Interface Configuration Register (AICR) R/W

CHL1	CHL0	STP	TR	0	0	0	0

CHL0–1 Character Length, excluding possible parity bit:

CHL1	CHL0	Length
0	0	eight
0	1	seven
1	0	six
1	1	five

STP Stop bits. One (0) or two (1) stop bits per character.

TR Tolerance Range. Normal 12.5% (0) or extended 25% (1) tolerance range for local DTE overspeed.

Data Path Configuration Register (DPCR) R/W

RDC1	RDC0	XDC1	XDC0	RSC	XSC	TL2	TL3

RDC0–1 Receive Data Connect

RDC1	RDC0	
0	0	RxD is connected to register bit RD
0	1	RxD is connected to Serial Communication Logic
1	0	RxD is connected to D-bits

XDC0–1 Transmit Data Connect

XDC1	XDC0	
0	0	D-bits are equal to register bit XD
0	1	D-bits originate from Serial Communication Logic
1	0	D-bits originate from TxD

RSC Receive S-bit Connect. Received S/SA, SB bits are mapped on to DCE/DTE interface interchange circuits (1).

XSC Transmit S-bit Connect. Transmitted S/SA, SB bits originate from the microcontroller interface (0) or are mapped from DCE/DTE interface interchange circuits (1).

TL2 Test Loop 2 activation.

TL3 Test Loop 3 activation.

HDLC Mode Register (HMR) R/W

HRLC	HXLC	HREN	HXEN	HINV	ITF	0	0

HRLC HDLC Receiver to Local DTE Connection. The HDLC Receiver is connected to DCE/DTE interface and the USART Receiver to network (1) or vice versa (0).

HXLC HDLC Transmitter to Local DTE Connection. The HDLC Transmitter is connected to DCE/DTE interface and the USART Transmitter to network (1) or vice versa (0).

HREN HDLC Receiver Enable.

HXEN HDLC Transmitter Enable.

HINV HDLC normal (0) or inverted (1).

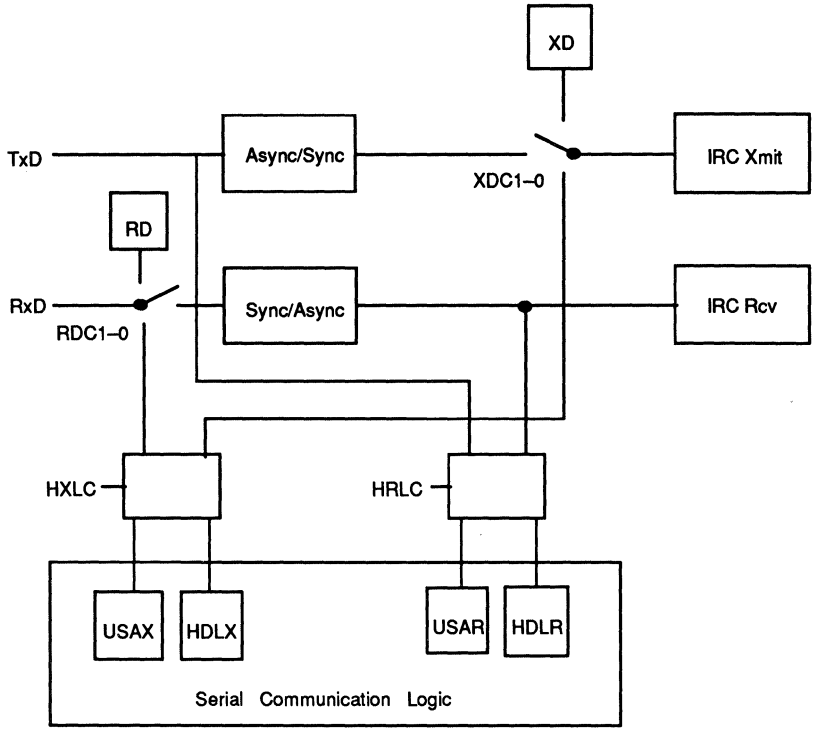


Figure 17. Serial Communication Logic

USART Mode Register (UMR) R/W

ASYC	PTY	UREN	UXEN	SCM	PY1	PY0	0

ASYC Asynchronous (1) or Synchronous (0) mode.

PTY Parity. Hardware parity check/generation (1) or not (0).

UREN USART Receiver Enable.

UXEN USART Transmitter Enable.

SCM Synchronous Communication Mode. Bisync (0) or Monosync (1).

PY0–1 Parity type. Used when PTY = 1.

PY1	PY0	Type
0	0	0
0	1	odd
1	0	even
1	1	1

Operational Parameter Registers

Bit Rate Select (BRS) R/W

UR3	UR2	UR1	UR0	NR3	NR2	NR1	NR0

UR0–3 User rate

NR0–3 Network rate

UR3	UR2	UR1	UR0	Rate, bps	DTE TYPE ASY = 1	DTE TYPE ASY = 0
0	0	0	0	300	x	
0	0	0	1	600	x	x
0	0	1	0	1200	x	x
0	0	1	1	2400	x	x
0	1	0	0	4800	x	x
0	1	0	1	9600	x	x
0	1	1	0	19200	x	x
0	1	1	1	38400	x	x
1	0	0	0	48000		x
1	0	0	1	56000		x
1	0	1	0	64000		x
1	0	1	1	reserved		
1	1	x	x	reserved		

NR3	NR2	NR1	NR0	Rate, bps
0	0	0	x	600
0	0	1	0	1200
0	0	1	1	2400
0	1	0	0	4800
0	1	0	1	9600
0	1	1	0	19200
0	1	1	1	38400
1	0	0	0	48000
1	0	0	1	56000
1	0	1	0	64000
1	0	1	1	reserved
1	1	x	x	reserved

Time-Slot Register (TSR) R/W

TS4	TS3	TS2	TS1	TS0	ICS2	ICS1	ICS0

TS0–4 Time Slot Select. Selects one 8-bit time slot out of a maximum of 32 or 64 (if TS5 is used) on the Synchronous Network Interface.

ICS0–2 Intermediate Rate Channel Select. Position of the first bit of intermediate rate channel inside the 8-bit time slot.

Number of Retry Frames Register (NRF) R/W

Determines the number of intermediate rate frames after initial Frame Sync Loss (FSL) status, during which resynchronization is attempted. If synchronism is not achieved within NRF frames, a RSI status is generated.

Control Registers

Local DTE Control Register (LDR) R/W

DCD/I	DSR	CTS	RD	MO1	MO2	0	0

DCD/I State of DCD/I interchange circuit.

DSR State of DSR interchange circuit.

CTS State of CTS interchange circuit.

RD Value of RxD.

MO1–2 State of Multifunctional Output 1–2.

Remote DTE Control Register (RDR) R/W

XS/XSA	XSB	XX	XD	0	0	0	0

XS/XSA Value of Transmitted S/SA bit.

XSB Value of Transmitted SB-bit.

XX Value of Transmitted X-bit.

XD Value of Transmitted D-bit.

Transmitted E-bit Register (XER) R/W

XE1	XE2	XE3	XE4	XE5	XE6	XE7	0

Value of transmitted E-bits in intermediate rate frame.

Command Registers

HDLC Controller Command Register (HCC) W

HRR	HXR	XME	HDMS	0	0	0	0

HRR HDLC Receiver Reset.

HXR HDLC Transmitter Reset.

XME Transmit Message End.

HDMS DMA Start for HDLC Transmitter.

USART Controller Command Register (UCC) W

URR	UXR	0	UDMS	SBK	HNT	TRA	0

URR USART Receiver Reset.

UXR USART Transmitter Reset.

UDMS DMA Start for USART Transmitter.

SBK Set Break. Forces the USART output to zero (start polarity) regardless of any data being transmitted at the time.

HNT Set Hunt Mode. Forces the USART to search for sync character(s) before starting to store data.

TRA Set Transparent Mode. Enables the USART to store receive data without regard for octet synchronization.

Character Insert Register (INSC) R/W

The character written in this register is inserted in the received character stream at the next opportunity. Received characters are not disturbed. The register may be written when CIS status bit is equal to "1."

Constants and Special Character Registers

Synchronization Character Register (SYN)

In the hunt phase, the receiver searches for one (monosync) or two (bisync) characters before storing data.

Local Characters 1 and 2 (LCAR1, LCAR2)

Characters from the DCE/DTE interface are compared with LCAR1 and LCAR2. Upon a match, an interrupt status may be generated, and the character is optionally stopped. Used for Hayes protocol and for XON/XOFF flow control.

Remote Characters 1 and 2 (RCAR1, RCAR2)

Characters from the remote DTE/TA are compared with RCAR1 and RCAR2. Upon a match, an interrupt status may be generated, and the character is optionally stopped. Used for XON/XOFF flow control.

Other Registers

Test Register (TEST) W

ABSOLUTE MAXIMUM RATINGS

Ambient temperature under bias ..... 0 to +70° C
Storage temperature ..... –65 to +125° C
Voltage on any pin
with respect to ground ..... –0.4 to V_{DD} +0.4 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Ambient Temperature (T_A) ..... 0° to +70°C
Operating V_{DD} ..... 4.75 V to 5.25 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range

T_A = 0 to +70°C; V_{DD} = 5 V ±5%, V_{SS} = 0 V.

Table 11. DC Characteristics

Symbol	Parameter	Limit Values Min.	Max.	Unit	Test Condition
V_{IL}	Input low voltage	–0.4	0.8	V	
V_{IH}	Input high voltage	2.0	V_{DD} + 0.4	V	
V_{OL}	Output low voltage		0.45	V	I_{OL} = 2mA
V_{OH}	Output high voltage	2.4		V	I_{OH} = –400 µA
V_{OH}	Output high voltage	V_{DD} –.5	V		I_{OH} = –200 µA
I_{CC}	Power operational supply			mA	V_{DD} = –5.V, CLK = 4 MHz Inputs at 0 V/V_{DD}
	Current power down			mA	No output loads
I_{LI}	Input leakage current		±10	µA	0 V < V_{IN} < V_{DD} to 0 V
I_{LO}	Output leakage current				0 V < V_{OUT} < V_{DD} to 0 V

Capacitances

T_A = 25°C, V_{DD} = 5 V ± 5%, V_{SS} = 0 V.

Table 12. DC Capacitances

	Symbol	Parameter	Limit Values Min.	Max.	Unit	Test Condition
All pins except SR1,2 XTAL1,2	CIN	Input capacitance			7	pF
	CIO	I/O capacitance			7	pF
XTAL1,2	CLD	Load capacitance			50	pF

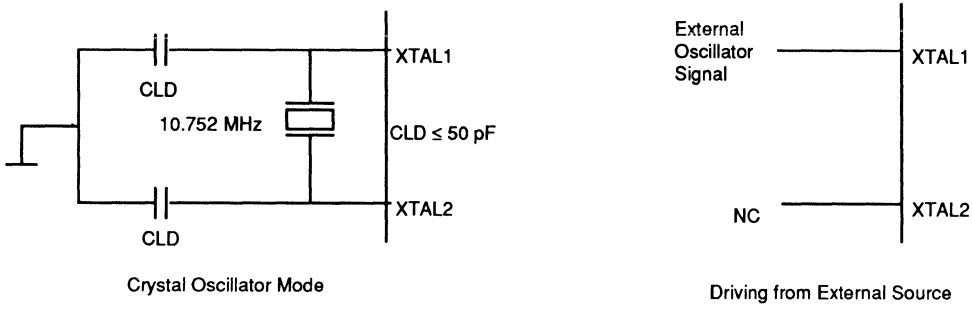


Figure 18. Recommended Oscillator Circuits

SWITCHING CHARACTERISTICS over operating range

T_A = 0 to +70°C, V_{DD} = 5 V ±5%

Inputs are driven to 2.4 V for a logical "1" and to 0.4 V for a logical "0". Timing measurements are made at 2.0 V for logical "1" and at 0.8 V for a logical '0". The AC testing input/output waveforms are shown in Figure 19.

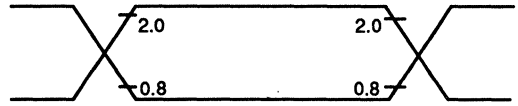


Figure 19. Input/Output Waveform for AC Tests

Clock

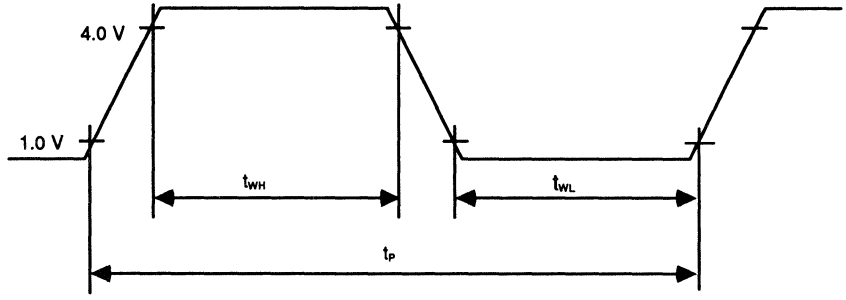


Figure 20. Definition of XTAL1 Period with Width

XTAL1,2

Table 13. XTAL1,2 Characteristics

Symbol	Description	Min.	Typ.	Max.	Unit	Condition
t_P	Clock period	−100 ppm	93.005	+100 ppm	ns	
t_{WH}	Clock high	35			ns	
t_{WL}	Clock low	35			ns	

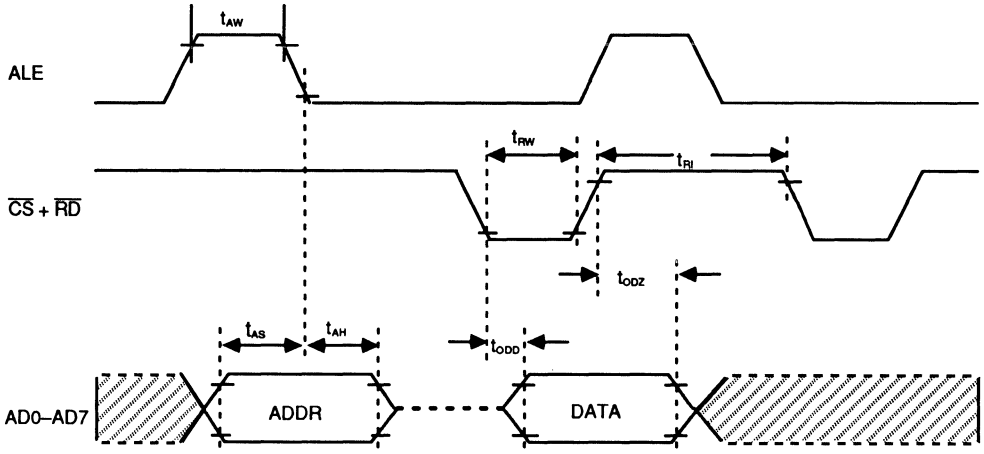


Figure 21. Microcontroller Read Cycle

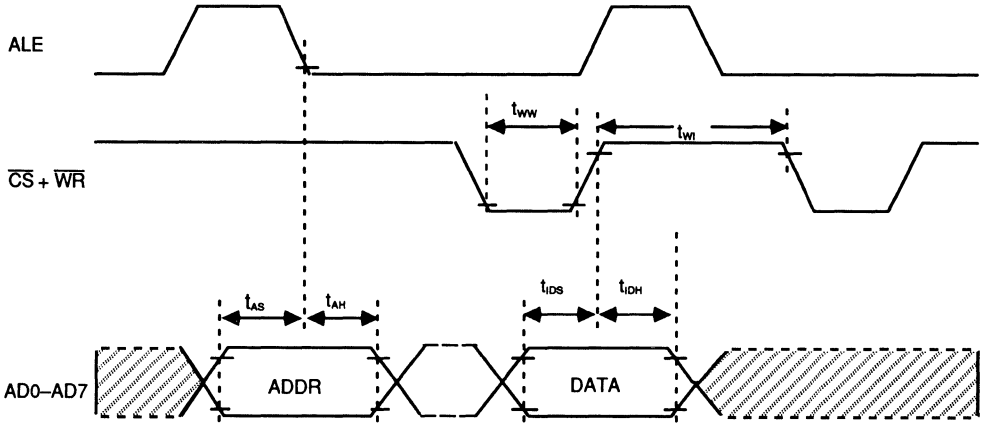
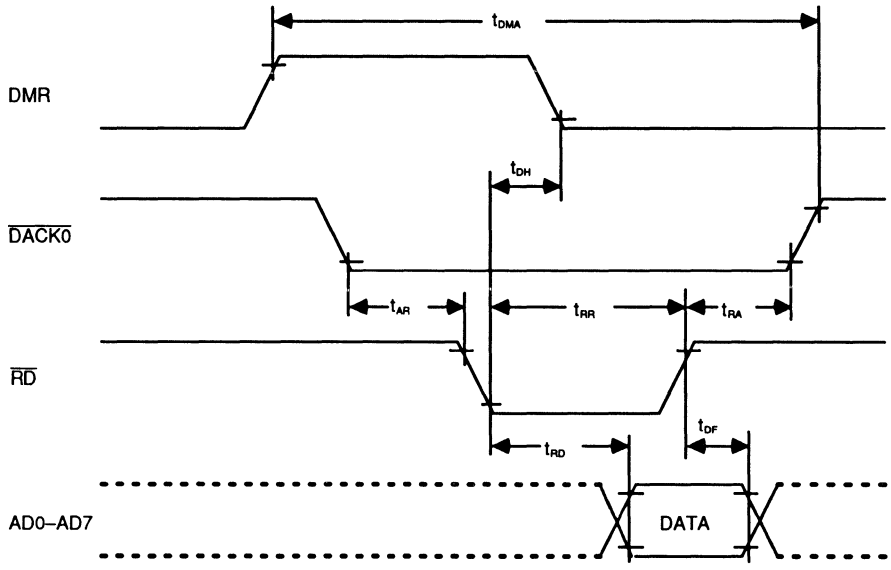


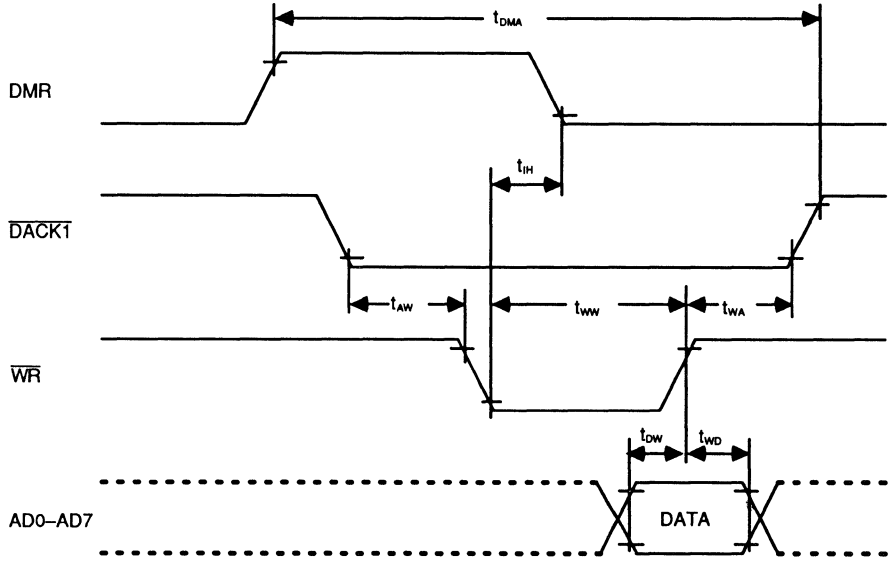
Figure 22. Microcontroller Write Cycle

Table 14. Microcontroller Interface Timing

Symbol	Description	Min.	Max.	Unit	Condition
t_{AW}	ALE pulse width	50		ns	
t_{AS}	Address setup time to ALE	20		ns	
t_{AH}	Address hold time from ALE	10		ns	
t_{RW}	$\overline{RD}$ pulse width	110		ns	
t_{ODD}	Data output delay from $\overline{RD}$		110	ns	
t_{ODZ}	Output data high impedance from $\overline{RD}$		25	ns	
t_{RI}	$\overline{RD}$ control interval	70		ns	
t_{WW}	$\overline{WR}$ pulse width	60		ns	
t_{IDS}	Data setup time to $\overline{WR}+\overline{CS}$	35		ns	
t_{IDH}	Data hold time from $\overline{WR}+\overline{CS}$	10		ns	
t_{WI}	$\overline{WR}$ control interval	70		ns	



a. DMA Read



b. DMA Write

Figure 23. DMA Cycle

Table 15. DMA Characteristics

Symbol	Description	Min.	Max.	Unit	Condition
t_{DH}	DMDR hold time		60	ns	
t_{AR}	Address stable before $\overline{RD}$	0		ns	
t_{RD}	Data delay from $\overline{RD}$		150	ns	
t_{DF}	Output floating delay	20		ns	
t_{RA}	Address hold after $\overline{RD}$	0		ns	
t_{RR}	$\overline{RD}$ pulse width	150		ns	
t_{IH}	DMIR hold time		80	ns	
t_{AW}	Address stable before $\overline{WR}$	0		ns	
t_{WA}	Address hold after $\overline{WR}$	0		ns	
t_{DW}	Data setup to $\overline{WR}$	30		ns	
t_{WD}	Data hold after $\overline{WR}$	25		ns	
t_{WW}	$\overline{WR}$ pulse width	100		ns	

Serial Interface Timing

Table 16. CLK Characteristics

Symbol	Description	Min.	Max.	Unit	Condition
t_P	CLK period	244		ns	
t_{WH}	CLK high	100		ns	
t_{WL}	CLK low	100		ns	

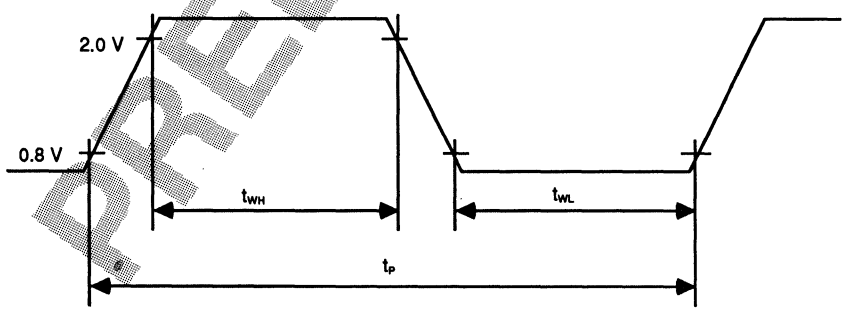


Figure 24. Definition of CLK Period and Width

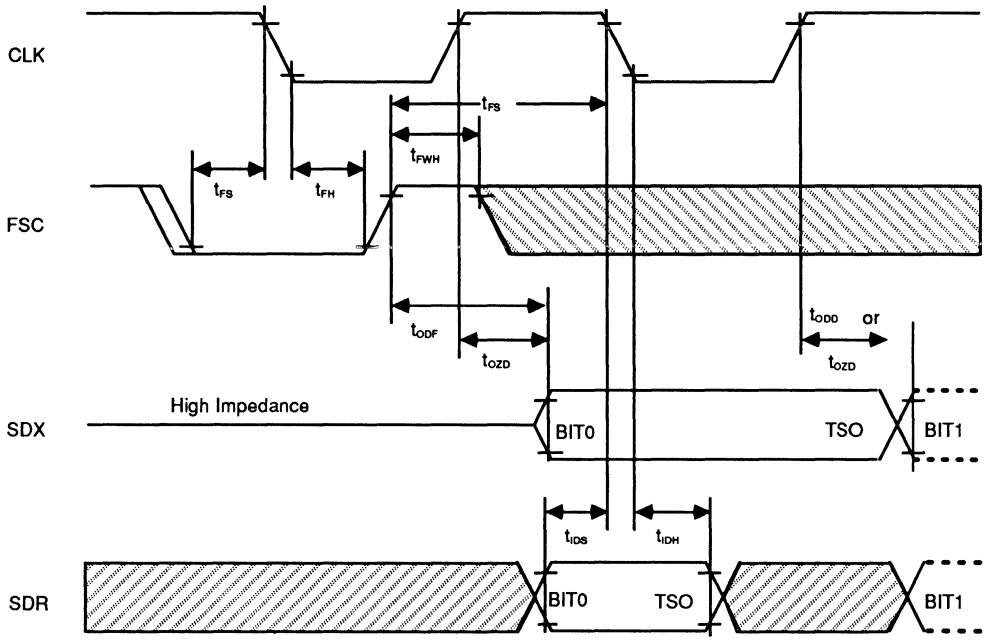


Figure 25. Synchronous Network Interface Timing

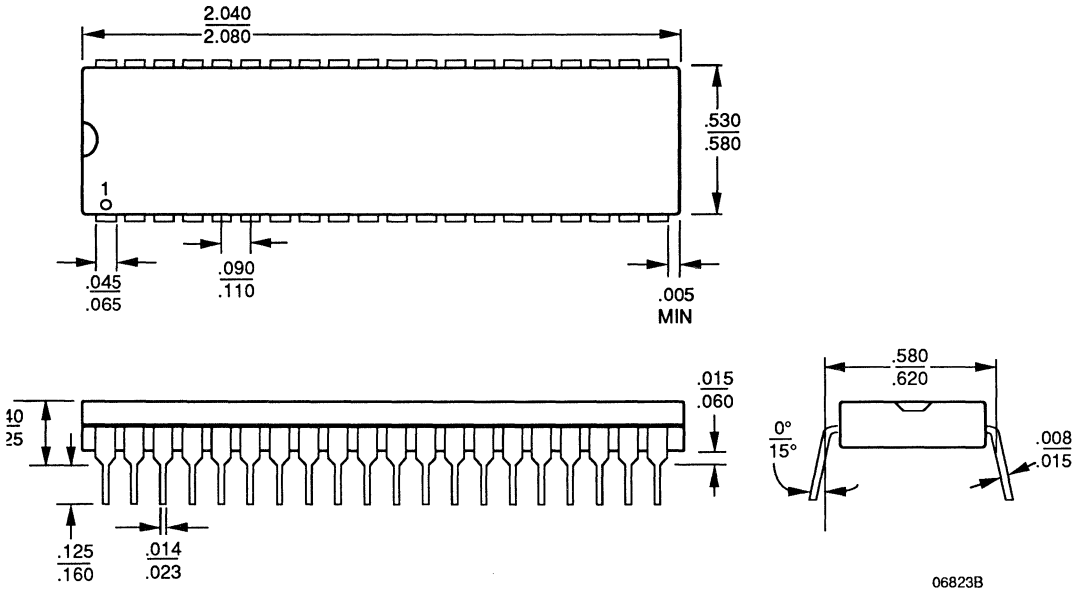
Table 17. Synchronous Network Interface Timing

Symbol	Description	Min.	Max.	Unit	Condition
t_{FS}	FSC setup time	40		ns	
t_{FH}	FSC hold time	40		ns	
t_{FWH}	FSC high width	40		ns	
t_{OZD}	SDX from high impedance to active from CLK		100	ns	
t_{ODD}	SDX from CLK		100	ns	
t_{ODZ}	SDX from active to high impedance from CLK		80	ns	
t_{ODF}	SDX from high impedance to active from FSC (Note)		100	ns	
t_{IDS}	SDR setup time	20		ns	
t_{IDH}	SDR hold time	40		ns	

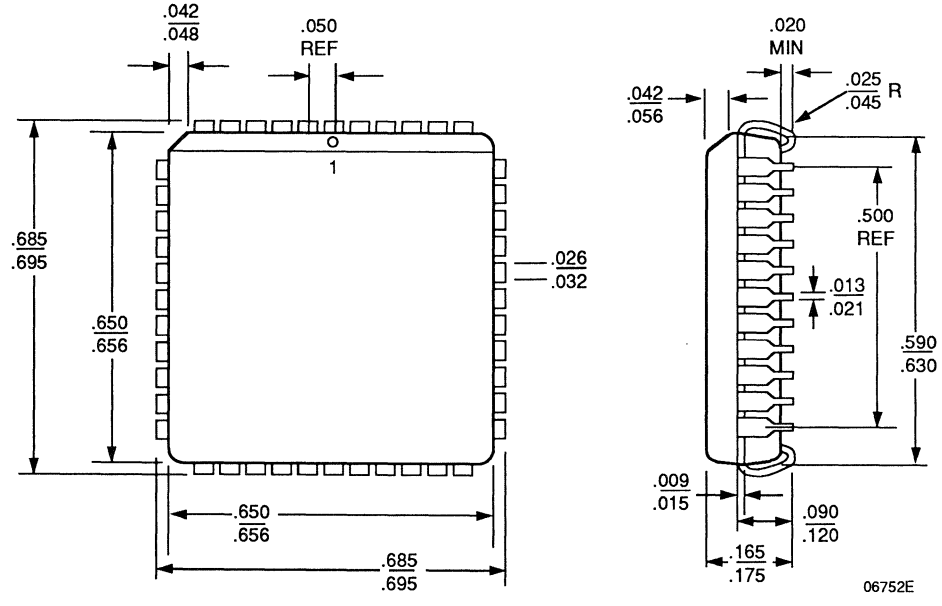
Note: This delay from FSC applies only when FSC repetition period is less than 512 bits, time slot 0 has been programmed, and CLK precedes FSC.

PHYSICAL DIMENSIONS

PD 040



PL 044



Advance Information

Am2160

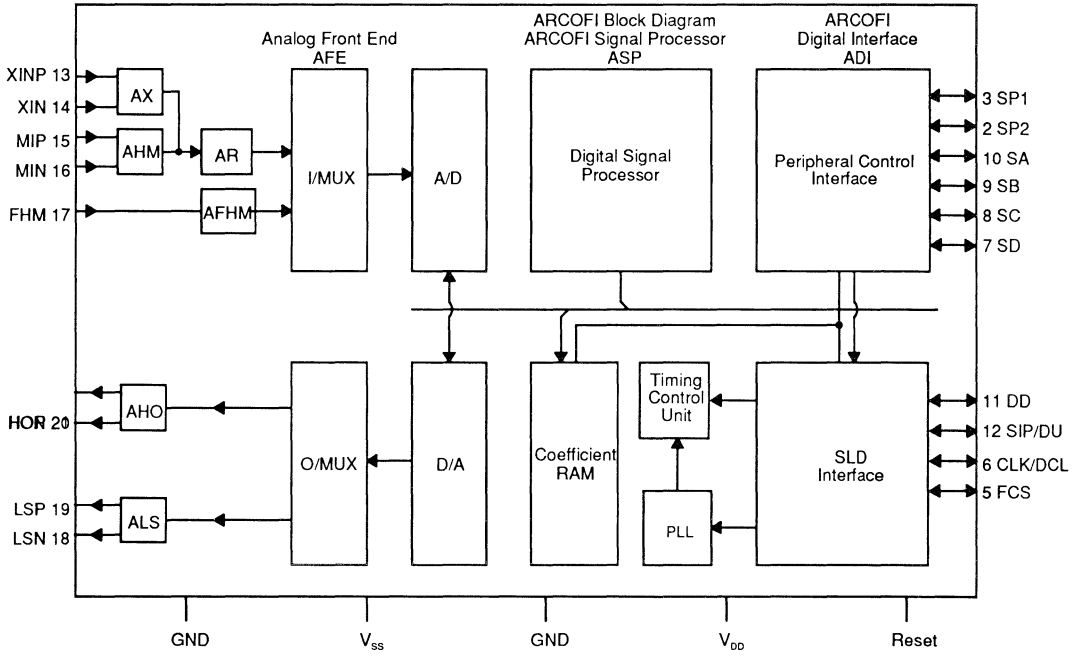
Audio Ringing Codec Filter (ARCOFI)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Applications in digital terminal equipment including a voice path
- Low power CMOS technology
- Test and maintenance loopbacks in the analog front end and the digital processor
- SLD or IOM™ Rev. 2 serial interface bus
- Flexible Peripheral Control Interface (PCI)
- CODEC filter
- DTMF, tone and ringing generators
- Separate output for a piezo ringer
- Dual analog inputs for handset and "handsfree" microphones plus an auxiliary differential analog input
- Two sets of differential outputs for a handset earpiece and loudspeaker
- Power dissipation:
  – active 150 mW
  – standby 10 mW
- Temperature range: –25 to +70°C
- Packages: 24-pin DIP
  28-pin PLCC

BLOCK DIAGRAM



Publication # 11153 Rev. B Amendment /0
Issue Date: March 1989

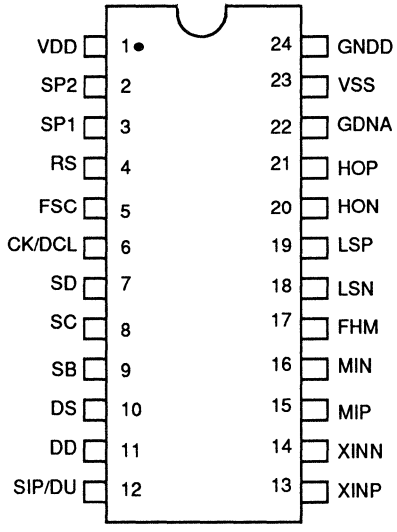
GENERAL DESCRIPTION

The Am2160 Audio Ringing Codec Filter (ARCOFI) provides the subscriber with an optimized audio, ringing, codec filter processor solution for a digital telephone. The ARCOFI fulfills all necessary requirements for the completion of a low cost digital telephone. Full featured applications including handsfree telephone are included by the addition of a voice switched speakerphone circuit. The ARCOFI performs all coding, decoding, and filtering functions according to CCITT and AT&T norms.

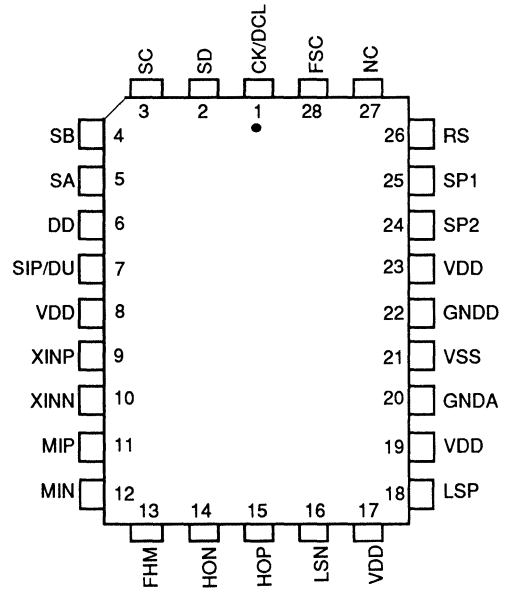
The ARCOFI integrates a DTMF generator in the transmit direction and a tone generator plus a ringing generator in the receive path. The interfacing to a handset mouth and earpiece is facilitated by a flexible analog front end. A loudspeaker output has also been integrated on chip as well as a secondary input for a handsfree microphone. The microphone analog gains is user programmable under microprocessor control.

CONNECTION DIAGRAMS
Top View

24-Pin DIP



28-Pin PLCC

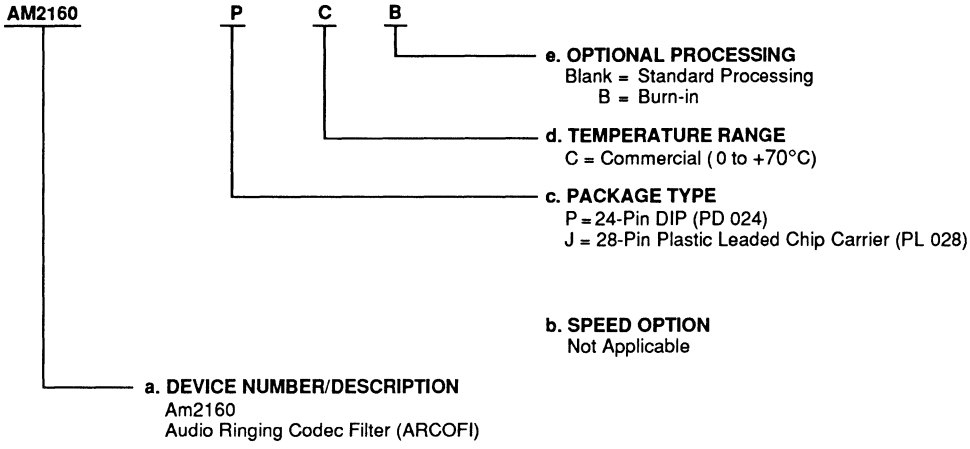


Note: Pin 1 is marked for orientation.

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM2160	PC, PCB, JC, JCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Preliminary

Am82520

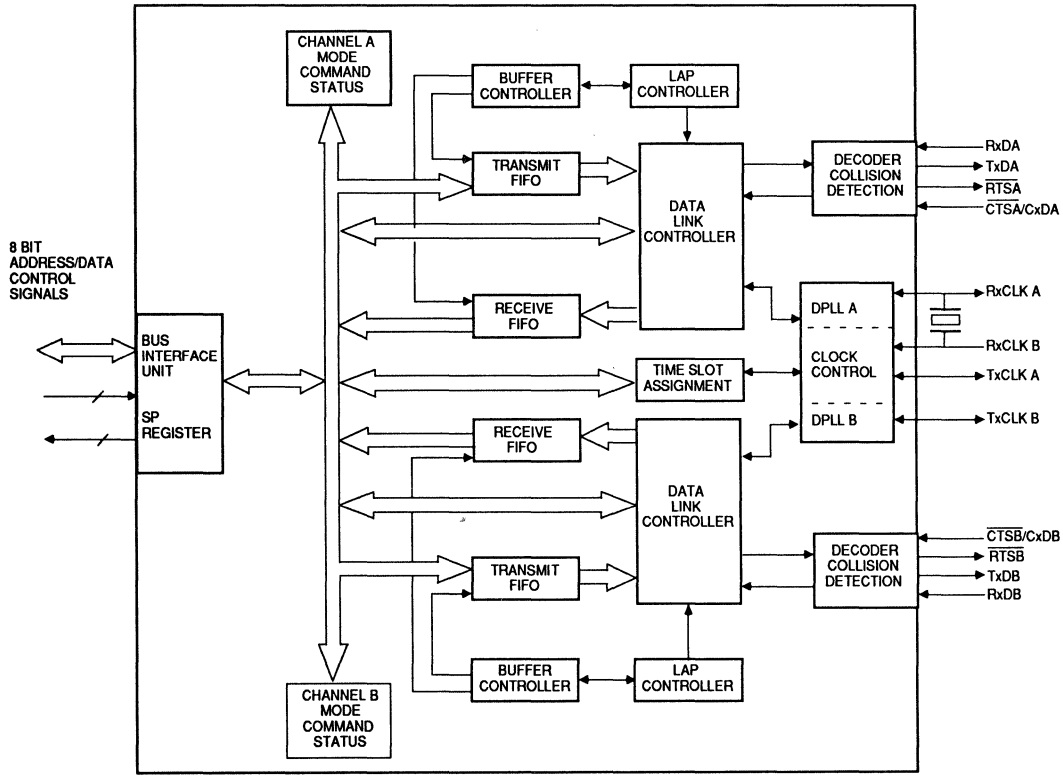
HSCC High-Level Serial Communications Controller

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Two independent HDLC channels
- Implementation of X.25 LAPB/LAPD protocol
- Programmable timeout and retry conditions
- FIFO buffers for efficient transfer of data packets
- Digital phase locked loop for each channel
- Baud rate generator and oscillator
- Different modes for clock recovery and data encoding
- High-speed data rate (up to 4 MHz)
- Supports bus configuration by collision resolution
- Telecom-specific features programmable
- 8-bit parallel microprocessor interface
- Advanced CMOS technology
- Low power consumption;
  active: 25 mW at 4 MHz
  standby: 3 mW
- Package: 28-Pin Plastic or 28-Pin Plastic Leaded Chip Carrier
- Operating temperature 0 to +70°C
- Industrial: operating temperature –40 to +85°C

BLOCK DIAGRAM



Publication #	Rev.	Amendment
11138	B	& /1

Issue Date: May 1989

DISTINCTIVE CHARACTERISTICS (continued)

Support of Layer 2 Functions by HSCC

"Low-level" HDLC devices usually support various protocols. When applying the HDLC protocol, mainly bit-oriented functions such as bit stuffing, CRC check, flag, and address recognition are performed. The Am82520 has been especially designed to support the ISO HDLC protocol. In addition to the bit-oriented functions, the device provides a high degree of procedural support by evaluating the layer 2 control field. In this way the communications procedures are processed between the communications controllers and not between the processors. As a result, procedure handshaking is minimized. The processor, however, is informed of the status of the procedure. The dynamic load of the processor is thus largely reduced. To maintain cost effectiveness and flexibility, not all layer 2 functions have been implemented in hardware. Instead, functions such as connection setup, cleardown, and error recovery errors are performed by the processor software.

Operating Modes

The distribution of functions between the HSCC and CPU applies to the auto mode. As a prerequisite for this operating mode, the window size has been limited to 1. Alternatively, transparent modes can be applied where the data field as well as the layer 2 headers are forwarded directly to the CPU. In these modes of operation the reception and transmission of messages is fully controlled by the CPU. This operating mode is selected when the component is used as a central station (master) or in the case where the window size is larger than 1. Furthermore, there is a possibility of bypassing the receiver and having direct access to the received data.

FIFO Buffers for Efficient Transfer of Data Packets

Another feature of the Am82520 can be seen in the buffers that are used for temporary storage of data packets which are transferred between the serial communication interface and the parallel system bus. Due to the overlapping input/output operation (dual-port behavior), the maximum length of the data packets is not limited by the buffer size. The dynamic load of the processor is reduced by transferring the data packets block by block.

Each channel on the HSCC incorporates a 64-byte FIFO buffer per direction. Each FIFO is divided into two memory pools of 32 bytes each. When a pool is filled (receive mode) or emptied (transmit mode) via the serial interface, the HSCC switches pools and the processor is prompted via an interrupt to read or write this pool. Subsequently, the second pool is filled or emptied. During this time the CPU can transfer the first block, ensuring availability of the pool. With a serial transfer rate of 1 Mbps, the reaction time between the first prompting and data overflow without loss of data is 256 µs. In addition, the transmit FIFO provides the flexibility for temporarily storing blocks of various lengths, which can then be transmitted in rapid succession. The receive FIFO can also store a data packet when a preceding short data packet (32 bytes) stored in the FIFO has not yet been read by the processor.

The HSCC is especially suitable for cost-critical applications with single-chip processors due to its memory organization and on-chip memory control.

Move string commands can be used in high-performance applications where fast data rates at the communication interface and a high level of processor performance is required. The FIFO contents can then by addressed by automatically incrementing the address.

GENERAL DESCRIPTION

The Am82520, High-Level Serial Communications Controller (HSCC), has been designed to free the user from tasks occurring in communication with networks and trunk lines. It is an X.25 LAPB/LAPD controller which to a large degree performs communication procedures independent of CPU support.

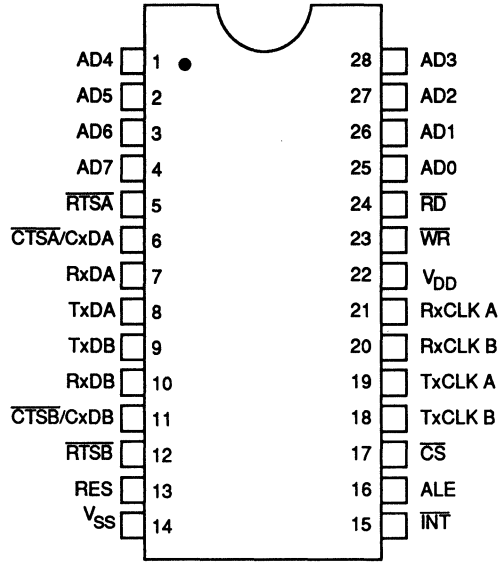
A parallel processor bus constitutes the microprocessor system. The serial communications interface consists of two full-duplex HDLC channels that can be operated independently from one another. The HSCC is connected to the transmission line by additional line drivers or modems. The need for external hardware is reduced because of added functions on-chip. The functions incorporated include an oscillator, DPLL (one per channel), programmable baud rate generator, and time slot assigner.

The chip contains a serial interface for two channels including a DPLL and collision-detection block, a data-link controller, and FIFO buffers. The microprocessor interface, including the status and command registers, is used for both channels. The HSCC is implemented in a 2 micron CMOS technology.

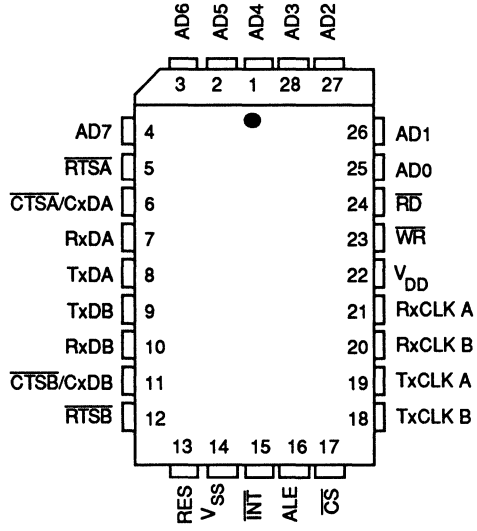
CONNECTION DIAGRAMS

Top View

28-Pin DIP

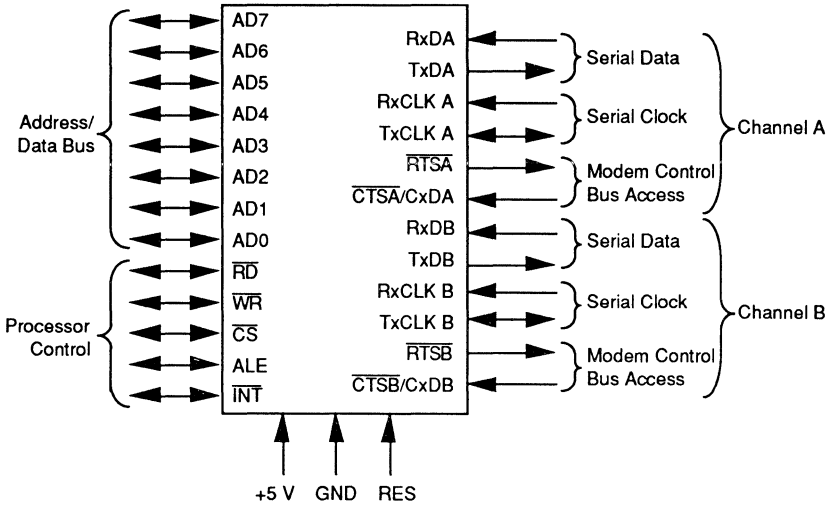


28-Pin PLCC



Note: Pin 1 is marked for orientation

LOGIC SYMBOL

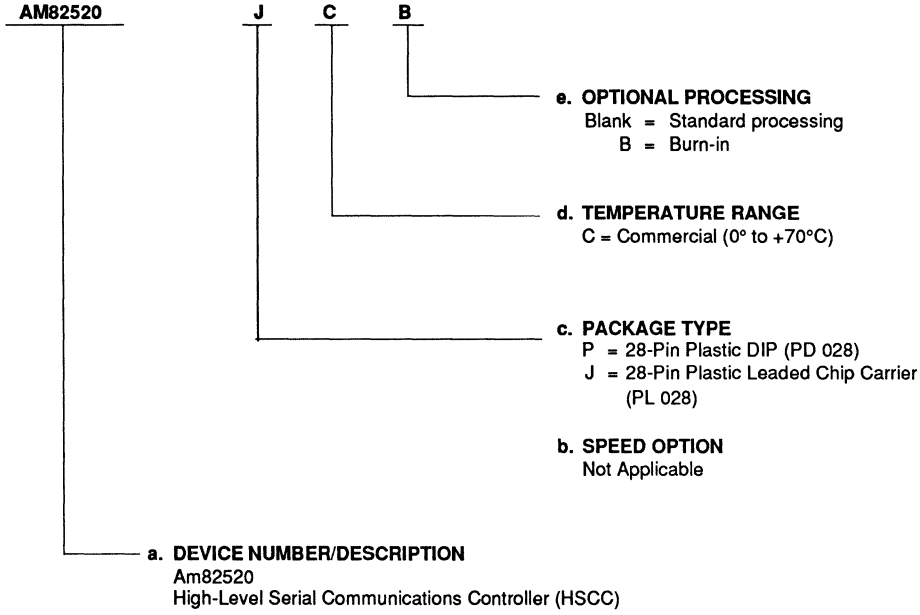


ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM82520	PC, JC PCB, JCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

AD0-AD7
Address Data Bus (Input, Output)
The multiplexed address data bus transfers data and commands between the microprocessor and the HSCC.

ALE
Address Latch Enable (Input)
A High on this line indicates an address on the external address data bus, selecting one of the HSCC's internal sources or destinations.

$\overline{CS}$
Chip Select (Input)
A Low on this signal selects the HSCC for a read/write operation.

$\overline{CTSA}$/CxDA, $\overline{CTSB}$/CxDB
Clear to Send/Collision Data (Input)
A Low on these inputs enables the respective transmitter. If the transmitters are always enabled, $\overline{CTS}$ should be connected to V_{SS}. In a bus configuration the external serial bus must be connected to the respective C x D pin.

$\overline{INT}$
Interrupt Request (Output)
This Signal is activated when the HSCC reqeusts an interrupt. It is an open-drain output.

$\overline{RD}$
Read (Input)
This signal indicates a read operation.

RES
Reset (Input)
A High on this input forces the HSCC into the reset state. The HSCC is in power-down mode during reset and in power-up mode after reset. The minimum pulse length is 1.8 µs.

$\overline{RTSA}$, $\overline{RTSB}$
Request to Send (Output)
When the RTS bit in Mode is set, the $\overline{RTS}$ signal goes Low. When the RTS bit is reset, the signal goes HIGH if the transmitter has finished and there is no further request for a transmission. In a bus configuration, $\overline{RTS}$ goes Low during the actual transmission of a frame shifted by a clock period, excluding collison bits.

RxCLKB, RxCLKA
Receive Clock (Input)
These pins can be programmed in several different modes of operation. In each channel RxCLK may supply the receive clock, the receive and transmit clock, the clock for the baud rate generator, or the clock for the DPLL. They also can be programmed for use as a crystal oscillator.

RxDA, RxDB
Receive Data (Input)
These input lines receive serial data at standard TTL or CMOS levels.

TxDA, TxDB
Transmit Data (Output)
These output lines receive serial data at standard TTL or CMOS levels. They can be programmed as push-pull or open-drain outputs.

TxCLKB, TxCLKA
Transmit Clock (Input/Output)
These pins can be programmed in several different modes of operation. TxCLK may supply the transmit clock for the respective channel, a receive strobe signal (TxCLKA) and a transmit strobe signal (TxCLKB) or a frame synchronization signal (TxCLKA, clock mode 5). Programmed as outputs, TxCLK may be used to supply the transmit clock fo the respective channel or as a tristate signal, indicating the programmed transmit time slot (TxCLKB, clock mode 5).

V_{DD}
Power
+5 V power supply.

V_{SS}
Ground (0 V)

$\overline{WR}$
Write (Input)
This signal indicates a write operation.

APPLICATIONS

In a point-to-multipoint or multimaster configuration, the HSCC can be used as a central station (master) or a peripheral station. As a peripheral station the HSCC can initiate the transmission of data. An internal function block provides for collision avoidance, which may occur if several stations start the transmitting simultaneously.

In a special operating mode the HSCC can transmit or receive data packets in programmable time slots. This makes the Am82520 especially suitable for applications in systems designed for packet switching and digital PABX applications. In the digital PABX application in particular, the integrated collision-resolution mechanism provides an optimal utilization of internal PCM paths.

Serial Interface

The serial interface provides two independent, high-performance communication interfaces. As already mentioned, the ISO HDLC layer 2 protocol is supported by the HSCC. In addition, layer 1 functions are provided by means of on-chip circuits. Eight different operating modes can be selected to clock the serial data stream.

In the power-down mode, all internal clocks as well as the oscillator circuitry are disabled.

- During the self-clocked operating mode, the transmit clock is recovered from the received data stream by means of an external crystal only. The on-chip DPLL samples the received bit stream and adjusts the clock edge to the center of the data bit.
- The bit stream is synchronized in the externally clocked operation mode by external clock signals. On the whole, four different clock signals separated by direction and channel, can be forwarded.
- In addition to the data clock, an externally supplied strobe signal can be applied to determine the time period during which data is to be received or transmitted. Using another operating mode, a time slot (up to 64) can be programmed for transmitting data and another time slot for receiving data. One time slot consists of eight clock cycles.
- With the point-to-multipoint configuration, comprising a central station (master) and several peripheral stations (slaves), data transmission can be initiated

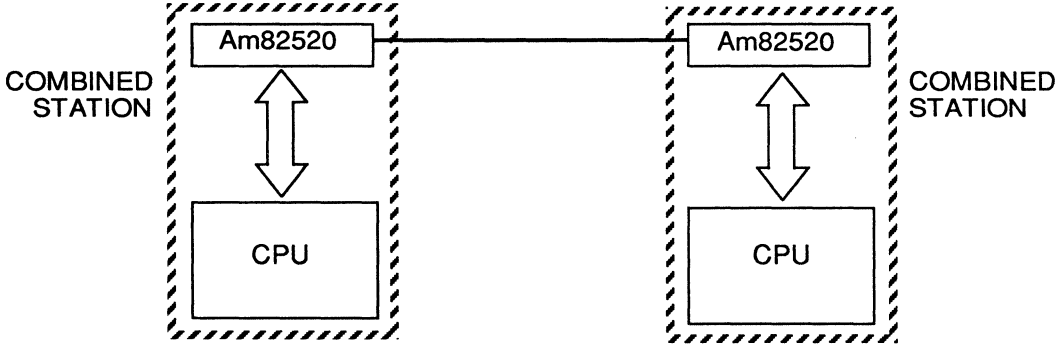


Figure 1. Point-to-Point Configuration

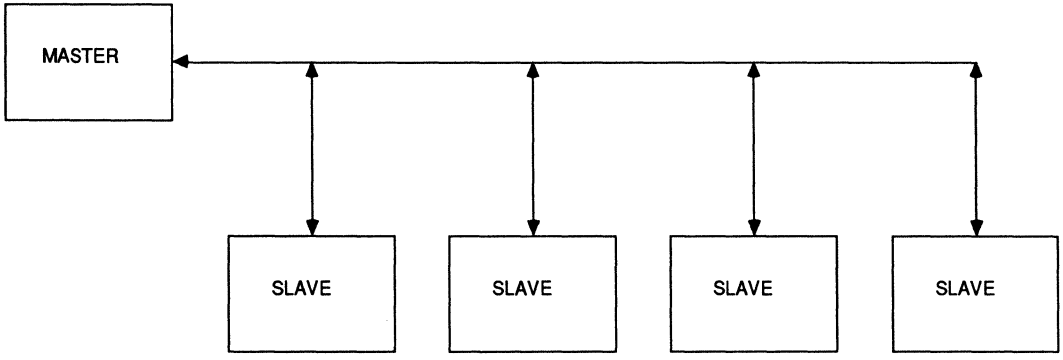


Figure 2. Point-to-Multipoint Configuration

by a slave. If several stations (slaves) transmit data simultaneously, the bus is assigned to one station by a collision-resolution procedure implemented by the HSCC. The bus assignment functions in accordance with the principle applied with the ISDN S interface. Its collision-resolution procedure helps to ensure a sharing of priority among the slave stations.

- The maximum data rate of the externally clocked operating mode is 4 Mbps. In the self-clocked operating mode with an external reference clock or crystal oscillator where the maximum clock rate is 12 MHz, the maximum data rate will be 750 Kbps.

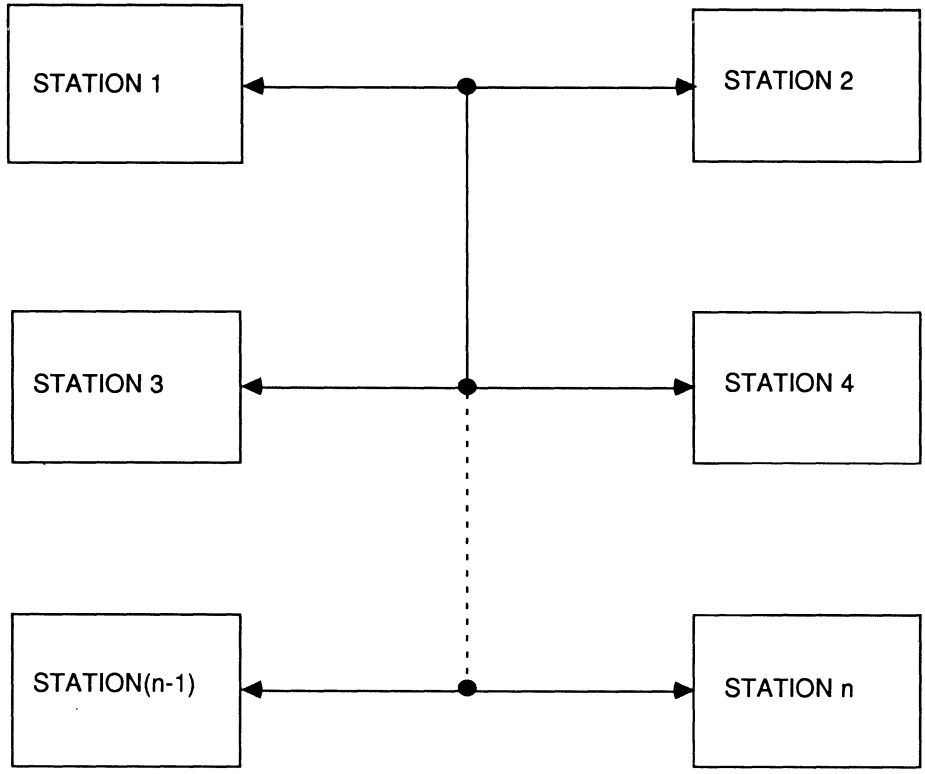


Figure 3. Multimaster Configuration

Absolute Maximum Ratings

Storage temperature (T_{stg})	–65°C to +125°C
Operating temperature: Am82520 (T_A)	0°C to +70°C
Voltage at any pin vs. ground (V)	–0.4 V_{DD} + 0.4 V

DC CHARACTERISTICS

Am82520: T_A = 0 to +70°C; V_{DD} = 5 V ± 10%; V_{SS} = GND = 0 V

Parameter		Test Conditions	Min.	Typical	Max.	Unit
Input Low voltage	V_{IL}		V_{SS}–0.4		0.8	V
Input High voltage	V_{IH}				V_{DD} +0.4	V
Output Low voltage	V_{OL}	I_{OL} = +2 mA			0.45	V
Output High voltage	V_{OH}	I_{OH} = –400 µA	2.4			V
		I_{OH} = –100 µA	V_{DD}–0.5	V_{DD}		V
Input leakage current	I_{IL}	V_{IN} = V_{DD} to 0 V	–10		+10	µA
Output leakage current	I_{OL}	V_{OUT} = V_{DD} to 0 V	–10		+10	µA
V_{DD} supply current						
Inactive	I_{CC}			0.5		mA
Active	I_{CC}	V_{DD} = 5 V t_{CP} = 4 MHz Inputs at V_{SS}/V_{DD} No output loads		5	7	mA

CAPACITANCE

T_A = +25°C; V_{DD} = GND = 0 V

Parameter		Test Conditions	Min.	Typ.	Max.	Unit
Input capacitance	C_{IN}	f_C = 1 MHz		5	10	pF
Input/output capacitance	$C_{I/O}$			10	20	pF
Output capacitance	C_{OUT}	Unmeasured pins returned to GND		8	15	pF

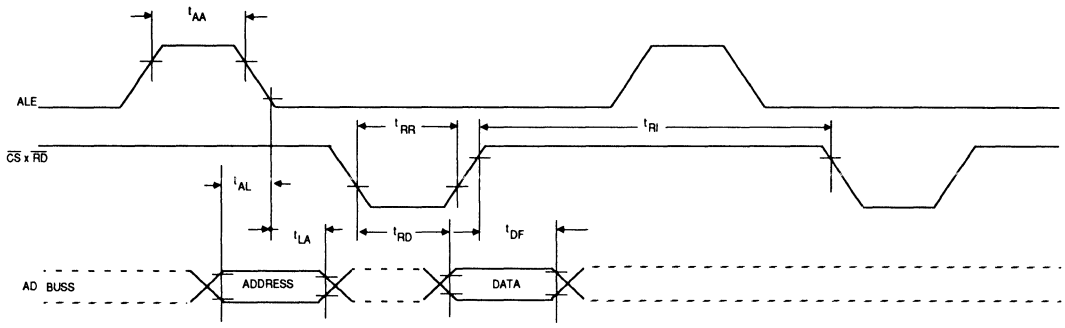


Figure 4. Read Cycle

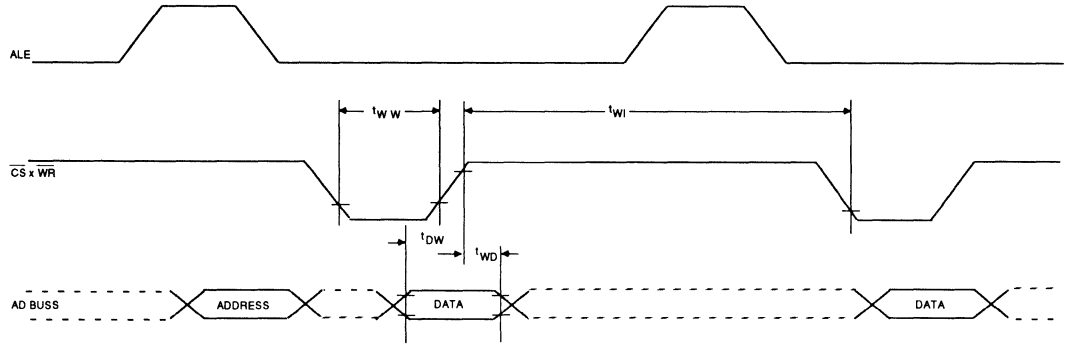


Figure 5. Write Cycle

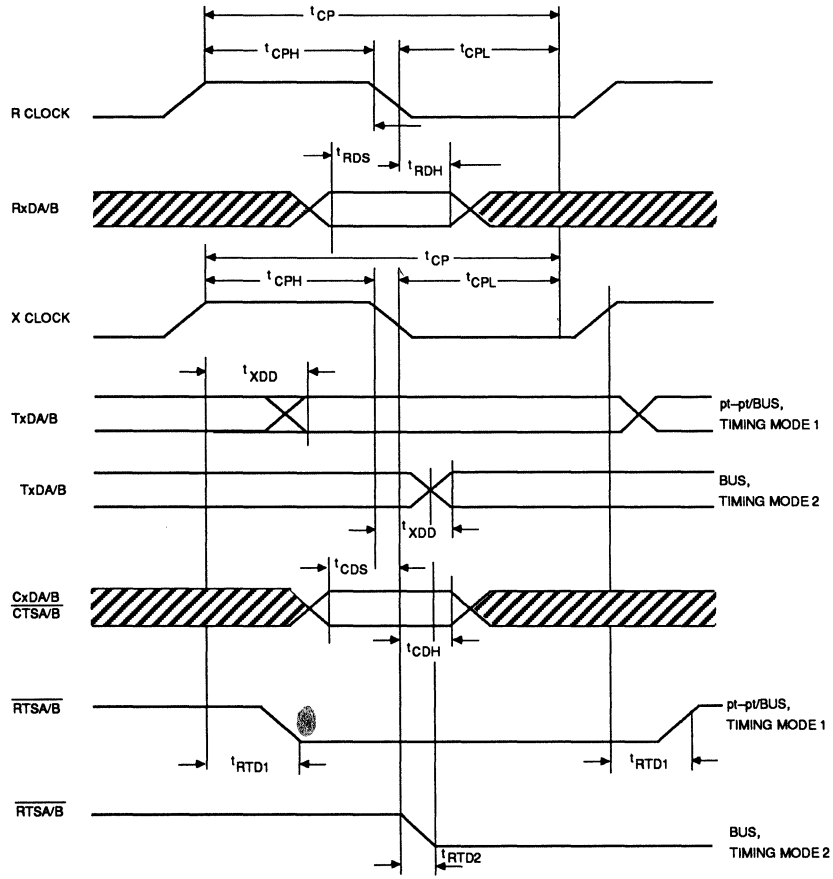
Read Cycle

Parameter		Min.	Max.	Unit
Address hold after $\overline{ALE}$ Low	t_{LA}	25		ns
Address to $\overline{ALE}$ Low setup	t_{AL}	20		ns
Data delay from $\overline{RD}$ Low	t_{RD}		110	ns
$\overline{RD}$ pulse width	t_{RR}	110		ns
Output float delay	t_{DF}		25	ns
$\overline{RD}$ control interval	t_{RI}	60		ns
$\overline{ALE}$ pulse width	t_{AA}	50		ns

Write Cycle

Parameter		Min.	Max.	Unit
$\overline{WR}$ pulse width	t_{WW}	60		ns
Data setup to $\overline{WR}$ High	t_{DW}	30		ns
Data hold after $\overline{WR}$ High	t_{WD}	10		ns
$\overline{WR}$ control interval	t_{WI}	60		ns

SERIAL INTERFACE TIMING



Serial Interface Timing

SWITCHING CHARACTERISTICS

Am82520: T_A = 0 to +70° C; V_{DD} = 5 V ± 10%; V_{SS} = GND = 0 V
T_A = –40 to +85° C; V_{DD} = 5 V ± 5%; V_{SS} = GND = 0 V

Parameter		Min.	Max.	Unit
Receive data setup	t_{RDS}	5		ns
Receive data hold	t_{RDH}	30		ns
Collision data setup	t_{CDS}	0		ns
Collision data hold	t_{CDH}	30		ns
Transmit data delay	t_{XDD}	20	68	ns
Request to send delay 1	t_{RTD1}	30	120	ns
Request to send delay 2	t_{RTD2}	20	85	ns
Clock period	t_{CP}	240		ns
Clock period LOW	t_{CPL}	90		ns
Clock period HIGH	t_{CPH}	100		ns

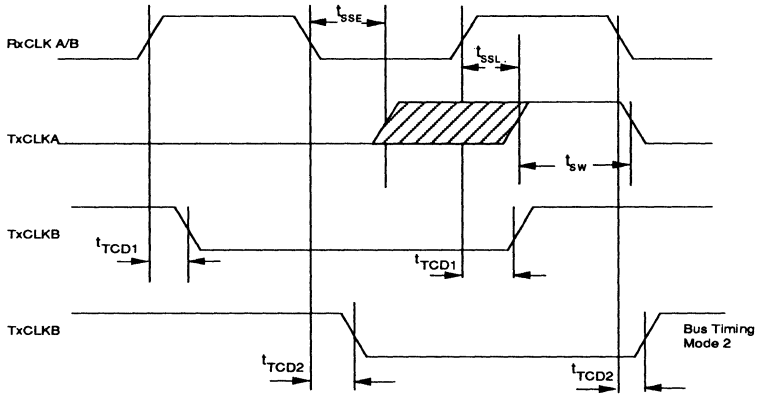


Figure 7. Clock Mode 5

Parameter		Min.	Max.	Unit
Sync pulse start early *	t_{SSE}	30		ns
Sync pulse start late **	t_{SSL}	0	30	ns
Sync pulse width	t_{SW}	40		ns
Time-slot control 2 delay	t_{TCD2}	20	85	ns
Time-slot control 1 delay	t_{TCD1}	30	120	ns

* If sync pulse starts before ↘ edge of RxCLK A/B first bit transmitted occurs on ↘ edge of RxCLK A/B.
** If sync pulse occurs after ↘ edge of RxCLK A/B first bit transmitted occurs on ↘ edge of sync pulse.

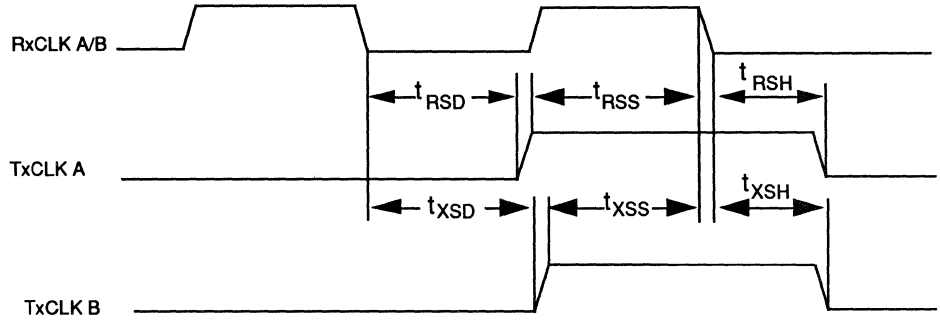


Figure 8. Clock Mode 1

Parameter		Min.	Max.	Unit
Receive strobe delay	t_{RSD}	30		ns
Receive strobe setup	t_{RSS}	70		ns
Receive strobe hold	t_{RSH}	30		ns
Transmit strobe delay	t_{XSD}	30		ns
Transmit strobe setup	t_{XSS}	70		ns
Transmit strobe hold	t_{XSH}	30		ns

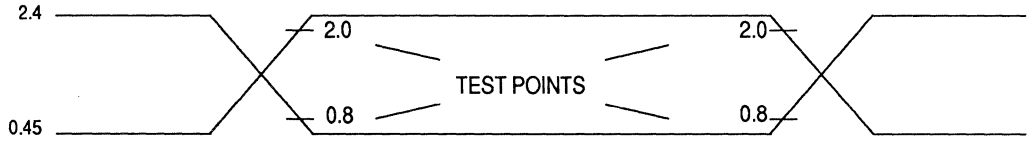


Figure 9. Test Points

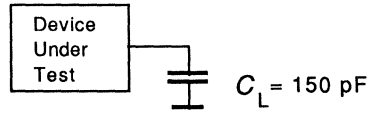
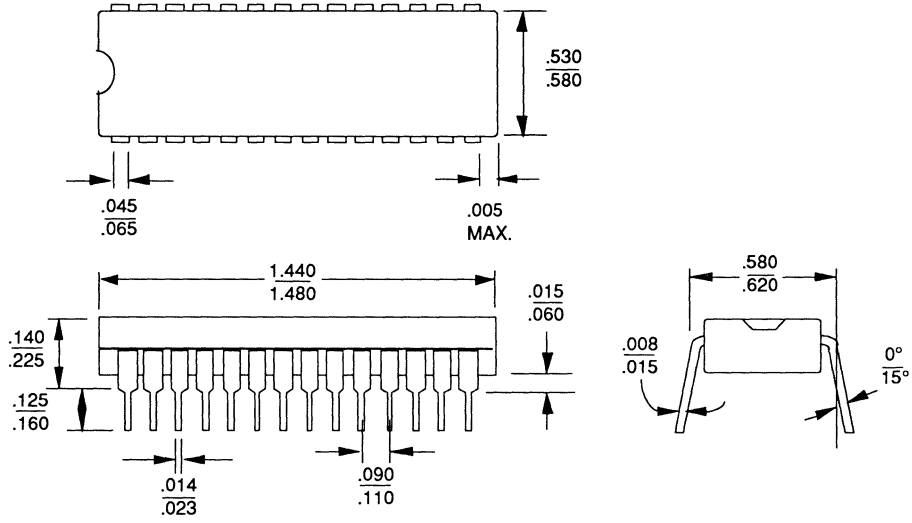


Figure 10. Load Circuit

AC Testing

Inputs are driven at 2.4 V for logical '1' and 0.45 V for logical '0'.
Timing measurements are made at 2.0V for logical '1' and at 0.8 V for logical '0'.

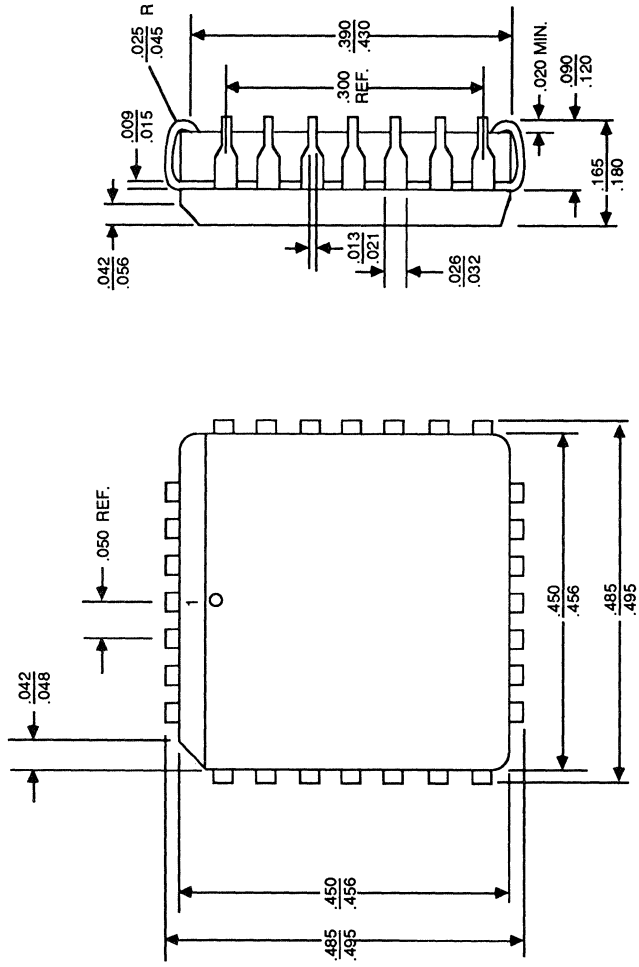
PD 028



06842B

PHYSICAL DIMENSIONS (continued)

PLCC 028



Am82520

High-Level Serial Communication Controller (HSCC)

Advanced Micro Devices

INTERFRAME TIME FILL

The HSCC B1 version transmits multiple eight-bit flag sequences (non-shared zero) according to CCITT X.25 recommendation when flags are selected as interframe time fill in the CCR register (see Figure 1).

Compared to the B1 version the A4 version transmits flags with shared zeros.

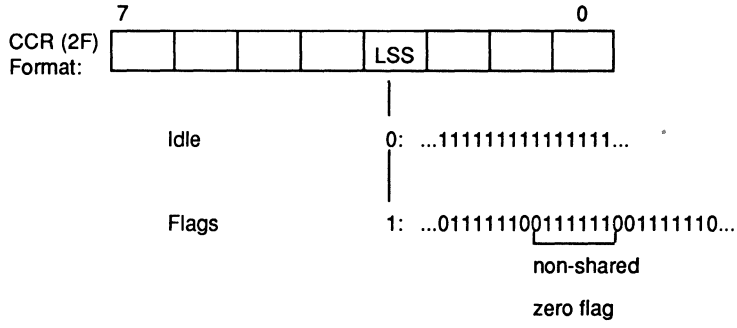


Figure 1. Interframe Time Fill of HSCC B1

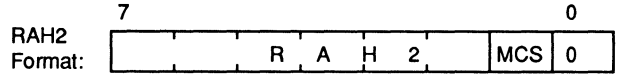
HDLC MULTIPLE FRAME OPERATION MODE

In addition to Modulo 8, the HSCC version B1 supports Modulo 128 operation in all message transfer modes, according to the HDLC protocol.

Modulo 8 or 128 operation can be selected by means of the MCS-bit in Register RAH2. The affected registers are listed below.

RAH2 - Receive Address Byte High Register 2 - (Write)

Address: 37H



Publication # 11138 Rev. B Amendment /1
Issue Date: May 1989

Bit Name	Description
RAH2 bit 7–2	HDLC Receive Address Byte High Value of the second individually programmable HDLC address byte, in ISDN applications (LAPD) the Service Access Point Identifier (SAPI).
MCS	Modulo Count Select - (valid in auto mode only) The MCS bit determines the control field format according to HDLC (ISDN/LAPD). 0 —> basic operation (modulo 8) 1 —> extended operation (modulo 128) Note: When modulo 128 is selected, in auto mode the "RHCR" register contains compressed information of the extended control field (see RHCR, register description).

RHCR - Receive HDLC Control Register - (Read)

Address: 39H

Bit Name	Description
bit 0–7 bit 7–0	Value of the received HDLC control field When modulo 128 is selected (MCS-bit of RAH2) in auto mode, the RHCR register contains compressed information of the extended control field, making it similar to a modulo 8 control field. Bit 0 of the RHCR register has the following meaning: 0 —> an I-frame has been received 1 —> a U-frame has been received (S-frames will be handled autonomously by the HSCC). When message transfer modes other than the auto mode are used and a HDLC protocol is used with modulo 128, then the first octet of the extended control field is available in the RHCR register. The second octet is available in the RFIFO in accordance with the message transfer mode. In extended transparent mode 0 (no address recognition) RHCR contains the second byte of a received frame after the opening flag.

SELF CLOCKED OPERATING MODE

In the self clocked operating mode with an external reference clock or a crystal oscillator, the maximum clock rate depends on the value of the baud rate division factor bit (BDF/TCR).

BDF	Division Factor	Max. Clock Rate (MHz)	Max. Data Rate (kHz)
0	1	19.52	1220
1	≠1	12	375

Final

Z85C30

Enhanced Serial Communications Controller

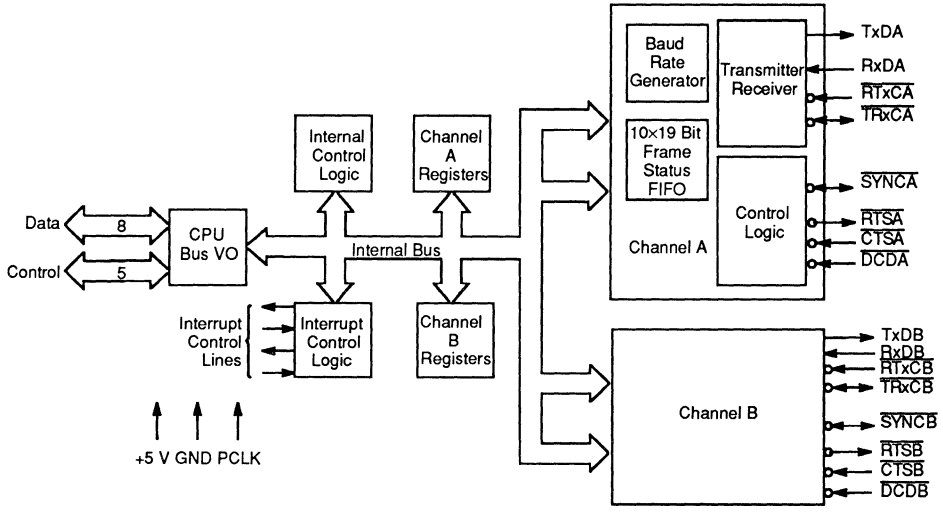
Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Fastest Data Rate of any Z8530**
  - 8.192 MHz / 2.048 Mb/s
  - 10 MHz / 2.5 Mb/s
  - 12.5 MHz / 3 Mb/s
  - 16.384 MHz / 4.096 Mb/s
  - 20 MHz / 5 Mb/s (prelim)
- **Low Power CMOS Technology**
- **Pin and Function Compatible with other NMOS and CMOS Z8530s**
- **Easily Interfaced with most CPUs**
  Compatible with non-multiplexed bus
- **Many Enhancements over NMOS Z8530H**
  - Allows 85C30 to be used more effectively in high-speed applications
  - Improves interface capabilities
- **Two Independent Full-duplex Serial Channels**
- **Asynchronous Mode Features**
  - Programmable stop bits, clock factor, character length and parity
  - Break detection/generation
  - Error detection for framing, overrun and parity
- **Synchronous Mode Features**
  - Supports IBM BISYNC, SDLC, SDLC Loop, HDLC and ADCCP Protocols
  - Programmable CRC generators and checkers
  - SDLC/HDLC support includes frame control, zero insertion and deletion, abort, and residue handling

1

BLOCK DIAGRAM



10216A-001A

BD008260

Publication # 10216 Rev. B Amendment /0
Issue Date: May 1989

DISTINCTIVE CHARACTERISTICS (continued)

- **Enhanced SCC functions support high-speed frame reception using DMA**
  - — 14-bit byte counter
  - — 10 × 19 SDLC/HDLC Frame Status FIFO
  - — Independent Control on both channels
  - — Enhanced operation does not allow special receive conditions to lock the three-byte DATA FIFO when the 10 × 19 FIFO is enabled
- **Local Loopback and Auto Echo Modes**
- **Internal or External Character Synchronization**
- **2 Mb/s FM Encoding Transmit and Receive capability using Internal DPLL for 16.384-MHz product**
- **Internal Synchronization between RxC to PCLK and TxC to PCLK**
  This allows the user to eliminate external sychronization hardware required by the NMOS device when transmitting or receiving data at the maximum rate of 1/4 PCLK frequency.

GENERAL DESCRIPTION

AMD's Z85C30 is an enhanced pin-compatible version of the popular Z8530/Z85C30 Serial Communications Controller. The Enhanced Serial Communications Controller (ESCC) is a high-speed, low-power, multi-protocol communications peripheral designed for use with 8- and 16-bit microprocessors. It has two independent, full duplex channels and functions as a serial-to-parallel, parallel-to-serial converter/controller. AMD's proprietary enhancements make the Z85C30 easier to interface and more effective in high-speed applications due to a reduction in software burden and the elimination of the need for some external glue logic.

The Z85C30 is easy to use due to a variety of sophisticated internal functions, including on-chip baud rate generators, digital phase-locked loops and crystal oscillators which dramatically reduce the need for external logic. The device can generate and check CRC codes in any SYNC mode, and can be programmed to check data integrity in various modes. The ESCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

This versatile device supports virtually any serial data transfer application such as networks, modems, cassettes and tape drivers. The ESCC is designed for non-multiplexed buses and is easily interfaced with most CPUs, such as 80188, 80186, 80286, 8080, Z80, 6800, 68000 and MULTIBUS.

Enhancements which allow the Z85C30 to be used more effectively in high-speed applications include:

- a 10 × 19 bit SDLC/HDLC frame status FIFO array
- a 14-bit SDLC/HDLC frame byte counter
- automatic SDLC/HDLC opening frame flag transmission
- TxD pin forced HIGH in SDLC NRZI mode after closing flag
- automatic SDLC/HDLC Tx underrun/EOM flag reset
- automatic SDLC/HDLC Tx CRC generator reset/preset
- $\overline{RTS}$ synchronization to closing SDLC/HDLC flag
- $\overline{DTR}/\overline{REQ}$ deactivation delay significantly reduced
- external PCLK to $\overline{RxC}$ or $\overline{TxC}$ synchronization requirement eliminated for PCLK divide-by-four operation

Other enhancements to improve the Z85C30 interface capabilities include:

- write data valid setup time to falling edge of $\overline{WR}$ requirement eliminated
- reduced $\overline{INT}$ response time
- reduced access recovery time (t_{RC}) to 3 PCLK best case (3 1/2 PCLK worst case)
- improved $\overline{Wait}$ timing
- write registers WR3, WR4, WR5, and WR10 made readable
- lower priority interrupt masking without $\overline{INTACK}$
- complete SDLC/HDLC CRC character reception

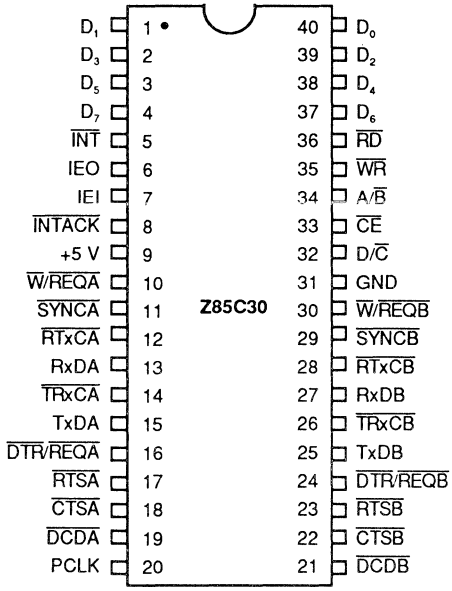
RELATED AMD PRODUCTS

Part No.	Description	Part No.	Description
Am7960	Coded Data Transceiver	Am9517A	DMA Controller
80186	Highly Integrated 16-Bit Microprocessor	5380, 53C80	SCSI Bus Controller
		80188	Highly Integrated 8-Bit Microprocessor
80286, 80C286	High-Performance 16-Bit Microprocessor		

CONNECTION DIAGRAMS

Top View

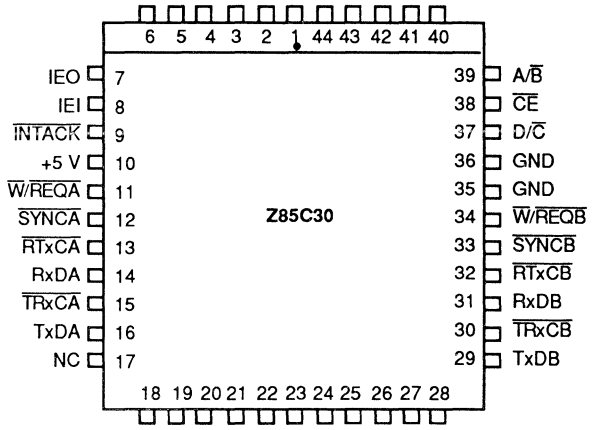
DIPs



10216A-002A

CD011530

PLCC/LCC



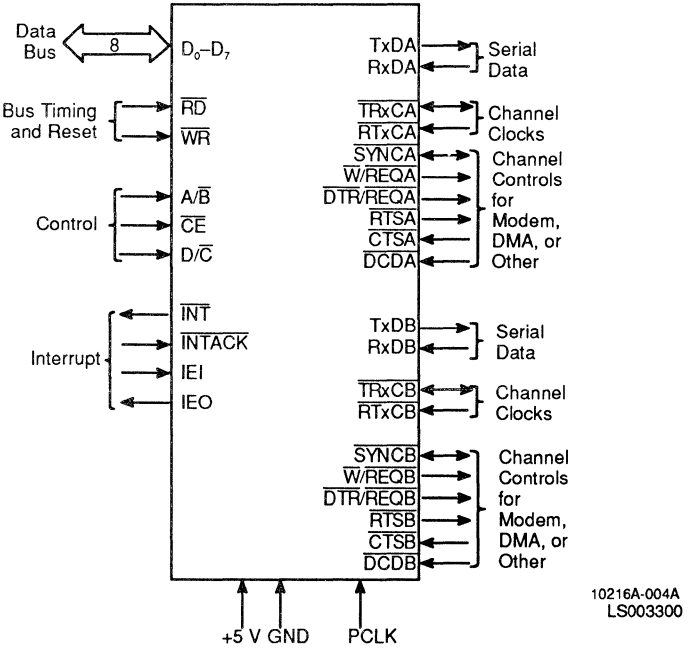
10216A-003A

CD011540

Note: Pin 1 is marked for orientation.

1

LOGIC SYMBOL



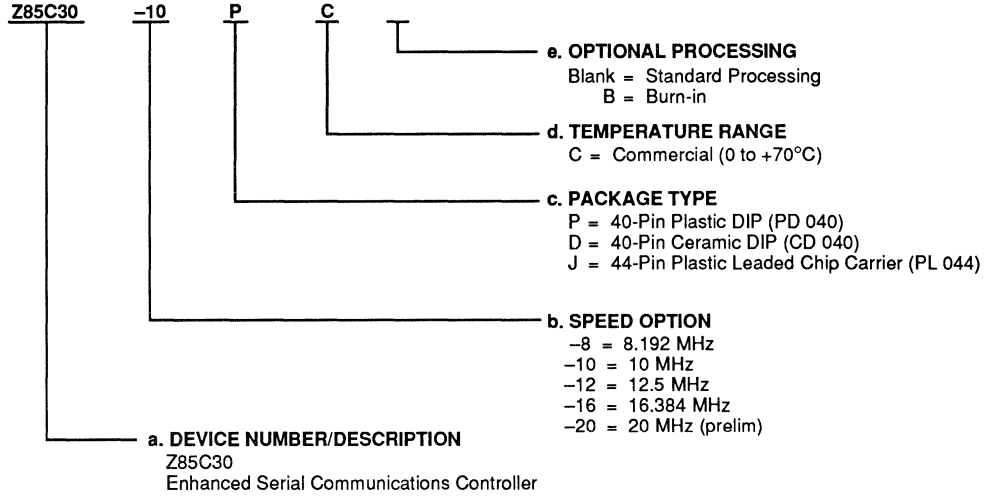
10216A-004A
LS003300

ORDERING INFORMATION

Commodity Products

AMD commodity products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
Z85C30-8 Z85C30-10 Z85C30-12 Z85C30-16 Z85C30-20 (prelim)	PC, DC, JC DCB

Valid Combinations

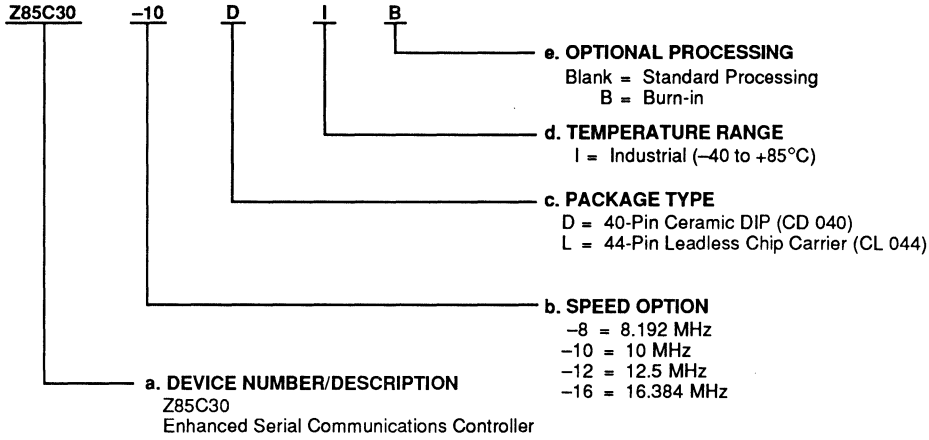
Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

ORDERING INFORMATION

Industrial Products

AMD industrial products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
Z85C30-8 Z85C30-10 Z85C30-12 Z85C30-16	DIB, LIB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

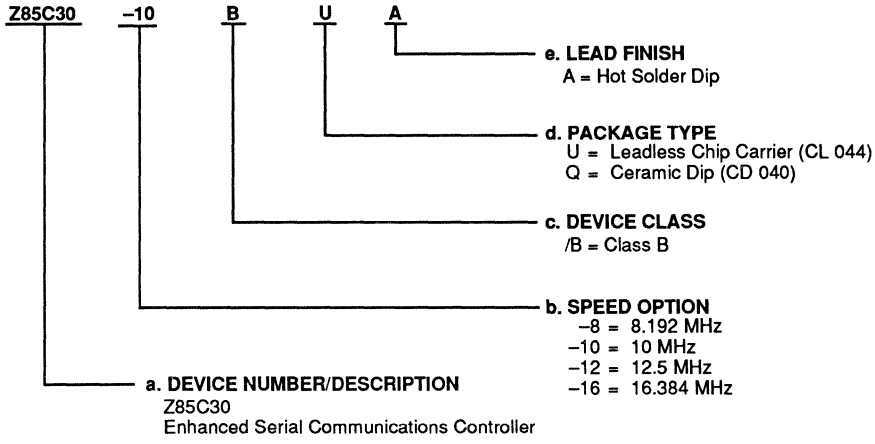
1

MILITARY ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
Z85C30-8 Z85C30-10 Z85C30-12 Z85C30-16	BQA, BUA

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Bus Timing and Reset

$\overline{RD}$
Read (Input; Active LOW)

This signal indicates a read operation and, when the SCC is selected, enables the SCC's bus drivers. During the Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.

$\overline{WR}$
Write (Input; Active LOW)

When the SCC is selected, this signal indicates a write operation. The coincidence of $\overline{RD}$ and $\overline{WR}$ is interpreted as a reset.

Channel Clocks

$\overline{RTxCA}$, $\overline{RTxCB}$
Receive/Transmit Clocks (Inputs; Active LOW)

These pins can be programmed in several different modes of operation. In each channel, $\overline{RTxC}$ may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock of the digital phase-locked loop. These pins can also be programmed for use with the respective $\overline{SYNC}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.

$\overline{TRxCA}$, $\overline{TRxCB}$
Transmit/Receive Clocks
(Inputs/Outputs; Active LOW)

These pins can be programmed in several different modes of operation. $\overline{TRxC}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the digital phase-locked loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.

Channel Controls for Modem, DMA, or Other

$\overline{CTSA}$, $\overline{CTSB}$
Clear to Send (Inputs; Active LOW)

If these pins are programmed as Auto Enables, a LOW on these inputs enables their respective transmitter. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects pulses on these inputs and may interrupt the CPU on both logic level transitions.

$\overline{DCDA}$, $\overline{DCDB}$
Data Carrier Detect (Inputs; Active LOW)

These pins function as receiver enables if they are programmed as Auto Enables; otherwise, they may be used as general-purpose input pins. Both are Schmitt-trigger buffered to accommodate slow rise-time signals. The SCC detects pulses on these pins and may interrupt the CPU on both logic level transitions.

$\overline{DTR/REQA}$
Data Terminal Ready/Request
(Outputs; Active LOW)

These outputs follow the inverted state programmed into the DTR bit in WR5. They can also be used as general-purpose outputs or as Request Lines for a DMA controller.

$\overline{RTSA}$, $\overline{RTSB}$
Request to Send (Outputs; Active LOW)

When the Request to Send RTS bit in Write Register 5 is set, the $\overline{RTS}$ signal goes LOW. When the RTS bit is reset in the asynchronous mode and Auto Enable is on, the signal goes HIGH after the transmitter is empty. In SYNC mode or in asynchronous mode with Auto Enable off, the $\overline{RTS}$ pins strictly follow the inverted state of the RTS bit. Both pins can be used as general-purpose outputs.

$\overline{SYNCA}$, $\overline{SYNCB}$
Synchronization (Inputs/Outputs; Active LOW)

These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to $\overline{CTS}$ and $\overline{DCD}$. In this mode, transitions on these lines affect the state of the Sync/Hunt status bits in Read Register 0 but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, $\overline{SYNC}$ must be driven LOW two receive clock cycles after the last bit in the SYNC character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of $\overline{SYNC}$.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which SYNC characters are recognized. The SYNC condition is not latched, so these outputs are active each time a SYNC pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

$\overline{W/REQA}$, $\overline{W/REQB}$
Wait/Request (Outputs; Open drain when programmed for a Wait function, driven HIGH or LOW when programmed for a Request function)

These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait.

PIN DESCRIPTION (continued)

Control

A/$\overline{B}$
Channel A/$\overline{\text{Channel B}}$ Select (Input)

This signal selects the channel in which the read or write operation occurs.

$\overline{CE}$
$\overline{\text{Chip Enable}}$ (Input; Active LOW)

This signal selects the SCC for a read or write operation.

D/$\overline{C}$
Data/$\overline{\text{Control}}$ Select (Input)

This signal defines the type of information transferred to or from the SCC. A HIGH means data is transferred; a LOW indicates a command is transferred.

Data Bus

D_0–D_7
Data Bus (Input/Output; Three State)

These lines carry data and commands to and from the SCC.

Interrupt

IEI
Interrupt Enable In (Input; Active HIGH)

IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A HIGH IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.

IEO
Interrupt Enable Out (Output; Active HIGH)

IEO is HIGH only if IEI is HIGH and the CPU is not servicing an SCC interrupt or the SCC is not requesting an interrupt (interrupt acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

$\overline{INT}$
$\overline{\text{Interrupt Request}}$ (Output; Active LOW, Open Drain)

This signal is activated when the SCC requests an interrupt.

$\overline{INTACK}$
$\overline{\text{Interrupt Acknowledge}}$ (Input; Active LOW)

This signal indicates an active interrupt acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When $\overline{RD}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is HIGH). $\overline{INTACK}$ is latched by the rising edge of PCLK.

Serial Data

RxDa, RxDB
Receive Data (Inputs; Active HIGH)

These input signals receive serial data at standard TTL levels.

TxDA, TxDB
Transmit Data (Outputs; Active HIGH)

These output signals transmit serial data at standard TTL levels.

Miscellaneous

GND
Ground

PCLK
Clock (Input)

This is the master SCC clock used to synchronize internal signals. PCLK is not required to have any phase relationship with the master system clock. PCLK is a TTL-level signal. Maximum transmit rate is 1/4 PCLK.

V_{CC}
+ 5 V Power Supply

ARCHITECTURE

The ESCC internal structure includes two full-duplex channels, two 10 × 19 bit SDLC/HDLC frame status FIFOs, two baud rate generators, internal control and interrupt logic, and a bus interface to a non-multiplexed bus. Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface with modems or other external devices (see Logic Symbol).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modem control.

The register set for each channel includes ten control (write) registers, two SYNC character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel, a write-only Master Interrupt Control register and three read registers: one containing the vector with status information (Channel B only), one containing the vector without status (A only), and one containing the interrupt Pending bits (A only).

The registers for each channel are designated as follows:

WR0–WR15—Write Registers 0 through 15. An additional write register, WR7 Prime (WR7'), is available for enabling or disabling additional SDLC/HDLC enhancements if bit D0 of WR15 is set.

RR0–RR3, RR10, RR12, RR13, RR15—Read Registers 0 through 3, 10, 12, 13, 15.

If bit D2 of WR15 is set, then two additional Read Registers, RR6 and RR7, are available. These registers are used with the 10 × 19 bit Frame Status FIFO.

Table 1 lists the functions assigned to each read and write register. The ESCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

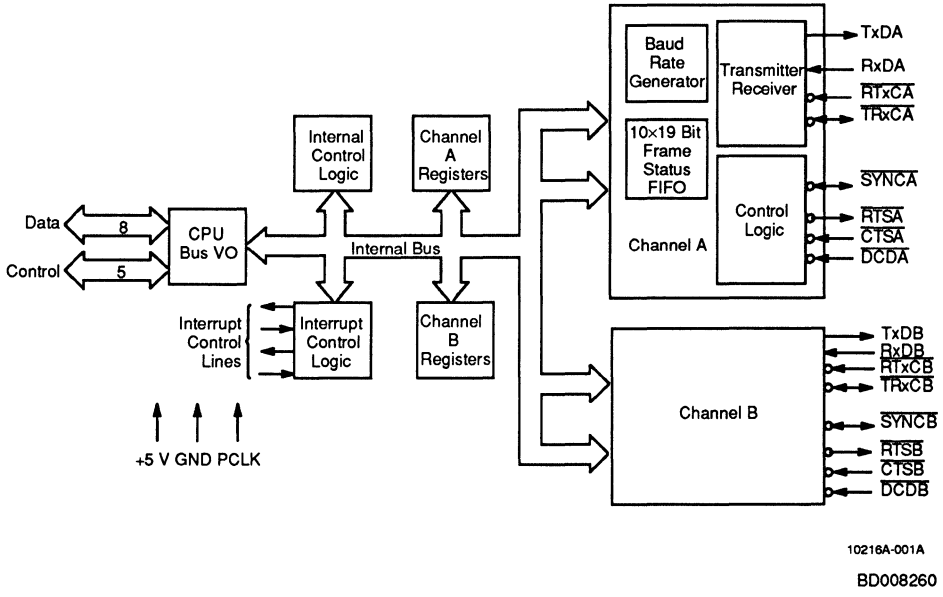


Figure 1. Block Diagram of ESCC Architecture

Data Path

The transmit and receive data path illustrated in Figure 2 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high-speed data. Incoming data are routed through one of several paths (data or CRC) depending on the selected mode (the character length in asynchronous modes also determines the data path).

The transmitter has an 8-bit transmit data buffer register loaded from the internal data bus and a 20-bit transmit shift register that can be loaded either from the sync-character registers or from the transmit data register. Depending on the operational mode, outgoing data are routed through one of four main paths before they are transmitted from the Transmit Data output (TxD).

Table 1. Read and Write Register Functions

Read Register Functions		Write Register Functions	
RR0	Transmit/Receive buffer status and External status	**WR0**	Command Register, Register Pointers CRC initialize, initialization commands for the various modes, shift right/shift left command
RR1	Special Receive Condition status (also 10 × 19 bit FIFO Frame Reception Status if WR15 bit D2 is set)	**WR1**	Interrupt conditions and data transfer mode definition
RR2	Modified interrupt vector (Channel B only) Unmodified interrupt vector (Channel A only)	**WR2**	Interrupt vector (accessed through either channel)
RR3	Interrupt Pending bits (Channel A only)	**WR3**	Receive parameters and control
RR6	LSB Byte Count (14-bit counter) (if WR15 bit D2 set)	**WR4**	Transmit/Receive miscellaneous parameters and modes
RR7	MSB Byte Count (14-bit counter) and 10 × 19 bit FIFO Status (if WR15 bit D2 is set)	**WR5**	Transmit parameters and controls
RR8	Receive buffer	**WR6**	Sync character or SDLC address field
RR10	Miscellaneous XMTR, RCVR status	**WR7**	Sync character or SDLC flag
RR12	Lower byte of baud rate generator time constant	**WR7′**	SDLC/HDLC enhancements (if bit D0 of WR15 set)
RR13	Upper byte of baud rate generator time constant	**WR8**	Transmit buffer
RR15	External/Status interrupt information	**WR9**	Master interrupt control and reset (accessed through either channel)
		WR10	Miscellaneous transmitter/receiver control bits, data encoding
		WR11	Clock mode control, Rx and Tx clock source
		WR12	Lower byte of baud rate generator time constant
		WR13	Upper byte of baud rate generator time constant
		WR14	Miscellaneous control bits, DPLL control
		WR15	External/Status interrupt control

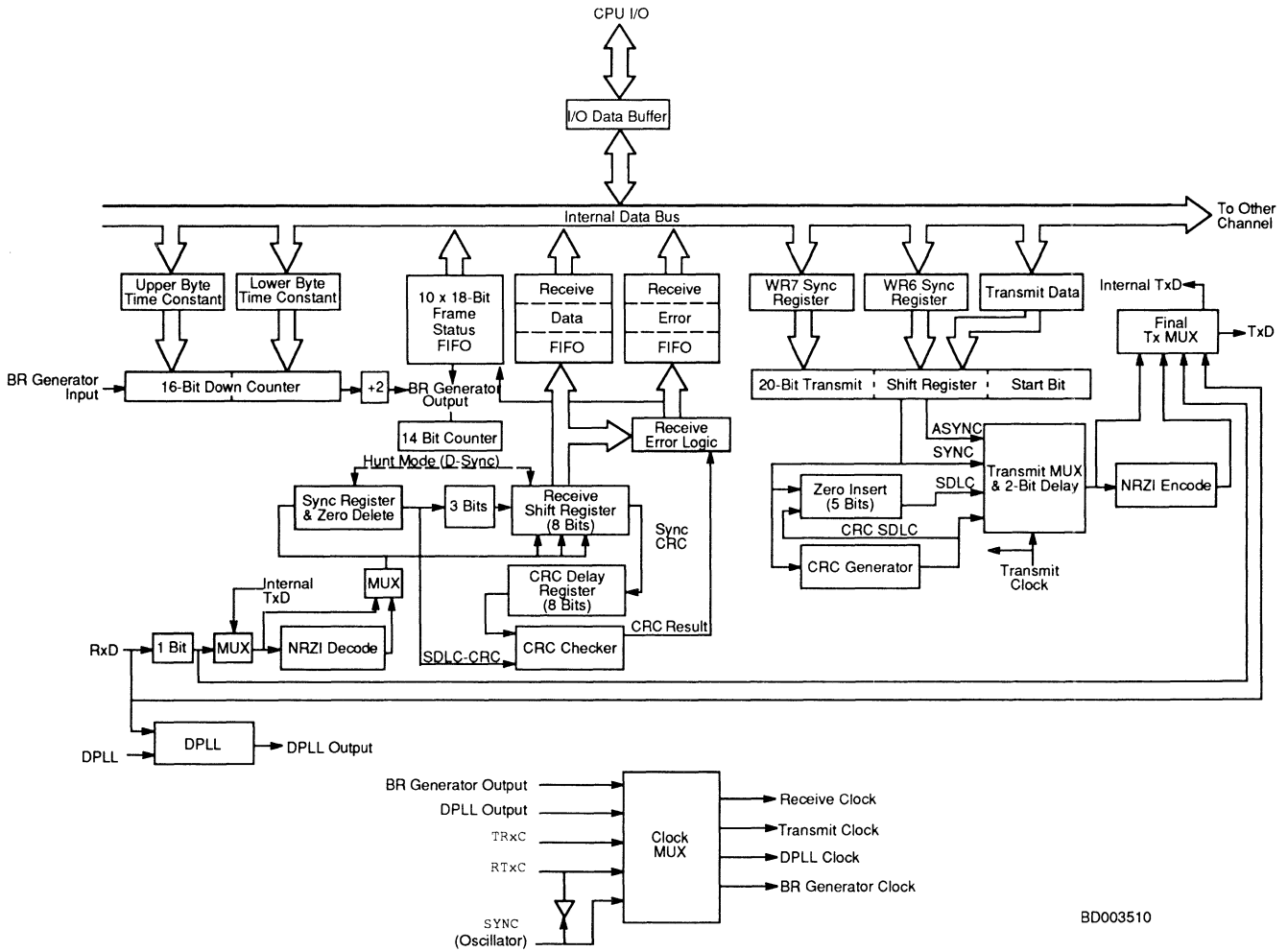


Figure 2. Data Path

DETAILED DESCRIPTION

The functional capabilities of the ESCC can be described from two different points of view; as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, it interacts with the CPU and provides vectored interrupts and handshaking signals.

Data Communications Capabilities

The ESCC provides two independent full-duplex channels programmable for use in any common asynchronous or SYNC data-communication protocol. Figure 3 and the following description briefly detail these protocols.

Asynchronous Modes

Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input. If the LOW does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing of error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of framing error as a new start bit; a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The ESCC does not require symmetric transmit and receive clock signals—a feature allowing use of the wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs. In asynchronous modes, the $\overline{SYNC}$ pin may be programmed as an input used for functions, such as monitoring a ring indicator.

Synchronous Modes

The ESCC supports both byte-oriented and bit-oriented synchronous communication. SYNC byte-oriented protocols can be handled in several modes, allowing character synchronization with a 6-bit or 8-bit SYNC character (Monosync), any 12-bit or 16-bit SYNC pattern (Bisync), or with an external SYNC signal. Leading SYNC characters can be removed without interrupting the CPU.

Five- or 7-bit SYNC characters are detected with 8- or 16-bit patterns in the ESCC by overlapping the larger pattern across multiple incoming SYNC characters as shown in Figure 4.

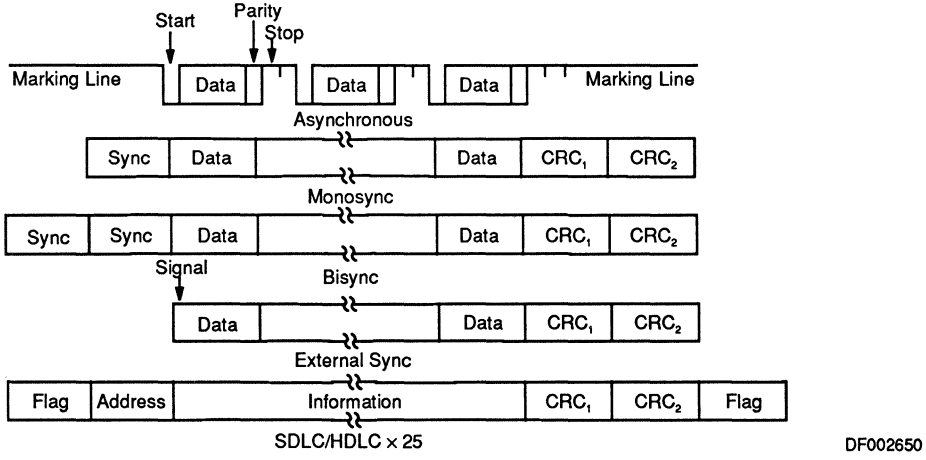


Figure 3. SCC Protocols

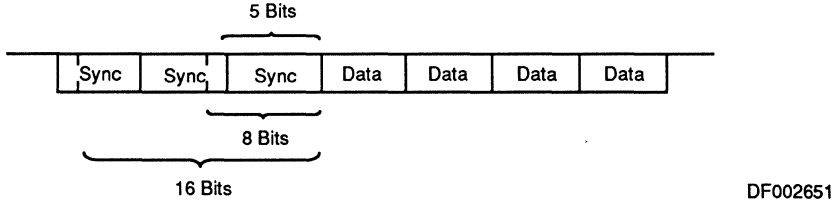


Figure 4. Detecting 5- or 7-Bit Synchronous Characters

CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols, such as IBM BISYNC.

Both CRC-16 ($X^{16} + X^{15} + X^2 + 1$) and CCITT ($X^{16} + X^{12} + X^5 + 1$) error checking polynomials are supported. Either polynomial may be selected in BISYNC and MONOSYNC modes. Users may preset the CRC generator and checker to all "1"s or all "0"s. The ESCC also provides a feature that automatically transmits CRC data when no other data are available for transmission. This allows for high-speed transmissions under DMA control with no need for CPU intervention at the end of a message. When there are no data or CRC to send in SYNC modes, the transmitter inserts 6-, 8-, or 16-bit SYNC characters, regardless of the programmed character length.

The ESCC supports SYNC bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero bit insertion, and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message, the ESCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. The ESCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the $\overline{SYNC}$ pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all "0"s inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the ESCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all "1"s or all "0"s. The CRC is inverted before transmission and the receiver checks against the bit pattern "0001110100001111."

NRZ, NRZI or FM coding may be used in any 1X mode. The parity options available in asynchronous modes are available in synchronous modes.

The ESCC can be conveniently used under DMA control to provide high-speed reception or transmission. In reception, for example, the ESCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The ESCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received. The CPU may also enable the DMA first and have the ESCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

SDLC Loop Mode

The ESCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow and any number of secondary stations. In SDLC Loop mode, the ESCC performs the functions of a secondary station while an ESCC operating in regular SDLC mode can act as a controller (Figure 5).

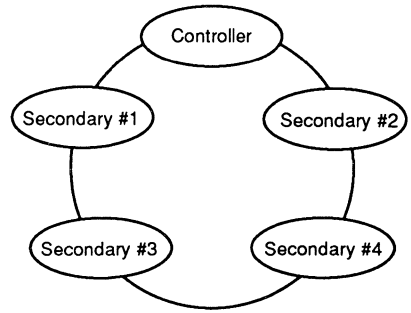


Figure 5. An SDLC Loop

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop and, in fact, must pass these messages to the rest of the loop by retransmitting them with a 1-bit time delay. The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End of Poll), around the loop. The EOP character is the bit pattern "11111110." Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary one of the EOP to a zero before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send

merely echo the incoming messages and are prohibited from placing messages on the loop (except upon recognizing an EOP).

SDLC Loop mode is a programmable option in the ESCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

Baud Rate Generator

Each channel in the ESCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On start-up, the flip-flop on the output is set in a High state; the value in the time constant register is loaded into the counter; and the counter starts counting down. The output of the baud rate generator toggles upon reaching zero; the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the digital phase-locked loop (see next section).

If the receive clock or transmit clock is not programmed to come from the $\overline{TRxC}$ pin, the output of the baud rate generator may be echoed out via the $\overline{TRxC}$ pin.

The following formula relates the time constant to the baud rate where PCLK or $\overline{RTxC}$ is the baud rate generator input frequency in Hz. The clock mode is X1, X16, X32, or X64 as selected in Write Register 4, bits D6 and D7. Synchronous operation modes should select X1 and Asynchronous should select X16, X32, or X64.

$$\text{Time Constant} = \left[\frac{\text{PCLK or RTxC Frequency}}{2\,(\text{Baud Rate})(\text{Clock Mode})}\right] - 2$$

The following formula relates the time constant to the baud rate. (The baud rate is in bits/second and the BR clock period is in seconds given by Clock Mode/Clock Frequency.)

$$\text{baud rate} = \frac{1}{2\,(\text{Time Constant} + 2) \times (\text{BR Clock Period})}$$

Time Constant Values for Standard Baud Rates at BR Clock = 3.9936 MHz			
Rate (Baud)	Time Constant (decimal/Hex notation)		Error
19200	102	(0066)	0
9600	206	(00CE)	0
7200	275	(0113)	0.12%
4800	414	(019E)	0
3600	553	(0229)	0.06%
2400	830	(033E)	0
2000	996	(03E4)	0.04%
1800	1107	(0453)	0.03%
1200	1662	(067E)	0
600	3326	(0CFE)	0
300	6654	(19FE)	0
150	13310	(33FE)	0
134.5	14844	(39FC)	0.0007%
110	18151	(46E7)	0.0015%
75	26622	(67FE)	0
50	39934	(98FE)	0

Digital Phase-Locked Loop

The ESCC contains a digital phase-locked loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the SCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32X clock to create nominal bit times. As the 32X clock is counted, the DPLL is searching the incoming data stream for edges (either 1/0 or 0/1). As long as no transitions are detected, the DPLL output will be free running and its input clock source will be divided by 32, producing an output clock without any phase jitter. Upon detecting a transition the DPLL will adjust its clock output (during the next counting cycle) by adding or subtracting a count of 1, thus producing a terminal count closer to the center of the bit cell. The adding or subtracting of a count of 1 will produce a phase jitter of ±5.63° on the output of the DPLL. Because the SCC's DPLL uses both edges of the incoming signal to compare with its clock source, the mark-space ratio (50%) of the incoming signal should not deviate by more than ±1.5% if proper locking is to occur.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15/16 counting transition.

The 32X clock for the DPLL can be programmed to come from either the $\overline{RTxC}$ input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the $\overline{TRxC}$ pin (if this pin is not being used as an input).

Crystal Oscillator

When using a crystal oscillator to supply the receive or transmit clocks to a channel of the SCC, the user should:

1. Select a crystal oscillator which satisfies the following specifications:
   - 30 ppm @ 25°C
   - 50 ppm over temperature of –20° to 70°C
   - 5 ppm/yr aging
   - 5 mW drive level
2. Place crystal across $\overline{RTxC}$ and $\overline{SYNC}$ pins
3. Place 30 pF capacitors to ground from both $\overline{RTxC}$ and $\overline{SYNC}$ pins
4. Set bit D_7 of WR11 to "1."

Data Encoding

The ESCC may be programmed to encode and decode the serial data in four different ways (Figure 6). In NRZ encoding, a "1" is represented by a High level, and a "0" is represented by a Low level. In NRZI encoding, a "1" is represented by no change in level, and a "0" is represented by a change in level. In FM_1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell. A "1" is represented by an additional transition at the center of the bit cell, and a "0" is represented by no additional transition at the center of the bit cell. In FM_0 (bi-phase space), a transition occurs at the beginning of every bit cell. A "0" is represented by an additional transition at the center of the bit cell, and a "1" is represented by no additional transition at the center of the bit cell. In addition to these four methods, the ESCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0/1, the bit is a "0." If the transition is 1/0, the bit is a "1."

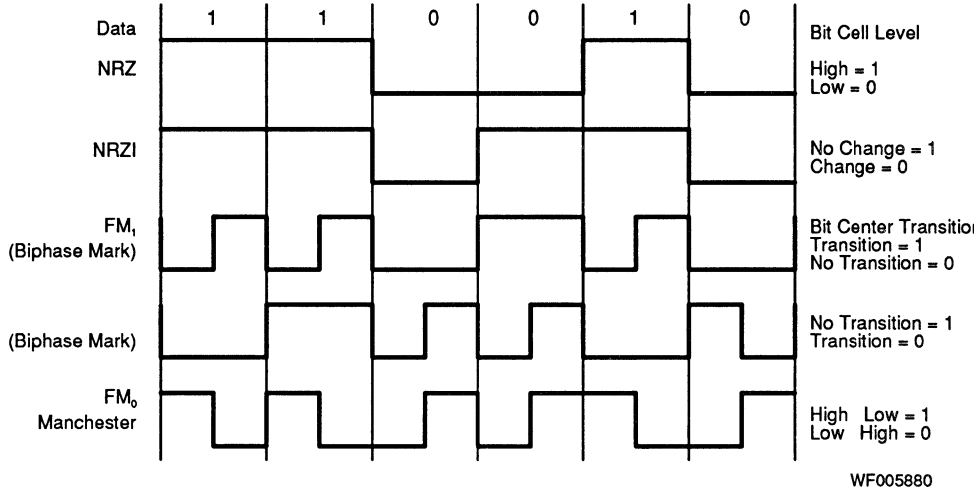


Figure 6. Data Encoding Methods

Auto Echo and Local Loopback

The ESCC is capable of automatically echoing everything it receives. This feature is useful mainly in asynchronous modes but works in SYNC and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In Auto Echo mode, the $\overline{CTS}$ input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed, and the programmer is responsible for disabling transmitter interrupts and $\overline{WAIT/REQUEST}$ on transmit.

The ESCC is also capable of Local Loopback. In this mode, TxD is RxD just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data, and RxD is ignored (except to be echoed out via TxD). The $\overline{CTS}$ and $\overline{DCD}$ inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in asynchronous, SYNC and SDLC modes with NRZ, NRZI or FM coding of the data stream.

I/O Interface Capabilities

The ESCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

Polling

All interrupts are disabled. Three status registers in the ESCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

Interrupts

When an ESCC responds to an Interrupt Acknowledge signal (INTACK) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures 8 and 9).

To speed interrupt response time, the ESCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the ESCC (Transmit, Receive and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write-only.

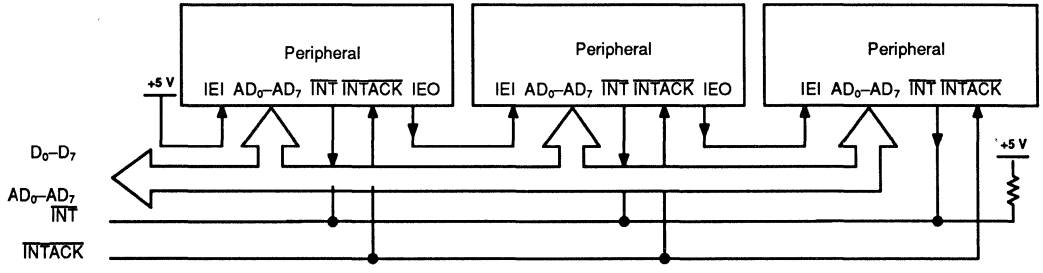
The other two bits are related to the Z-Bus interrupt priority chain (Figure 7). As a Z-Bus peripheral, the ESCC may request an interrupt only when no higher priority device is requesting one, for example, when IEI is HIGH. If the device in question requests an interrupt, it pulls down $\overline{INT}$. The CPU then responds with $\overline{INTACK}$, and the interrupting device places the vector on the A/D bus.

In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is set to "1" and the IEI input is HIGH, the $\overline{INT}$ output is pulled LOW, requesting an interrupt. In the ESCC, if the IE bit is set for an interrupt, then the IP for that source can never be set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the ESCC and external to the ESCC are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the ESCC being pulled LOW and propagated to subsequent peripherals. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts.

There are three types of interrupts: Transmit, Receive and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receive, Transmit and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the Receive can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive condition
- Interrupt on all Receive Characters or Special Receive condition
- Interrupt on Special Receive condition only



AF002770

Figure 7. Z-Bus Interrupt Schedule

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in asynchronous mode, end-of-frame in SDLC mode, and optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary Receive Character Available interrupt only in the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt can occur from Special Receive Conditions any time after the first Receive Character Interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{CTS}$, $\overline{DCD}$, and $\overline{SYNC}$ pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition, a zero count in the baud rate generator, the detection of a Break (asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the ESCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the wishes of the primary station to regain control of the loop during a poll sequence.

CPU/DMA Block Transfer

The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{WAIT/REQUEST}$ output in conjunction with the Wait/Request bits in WR1. The $\overline{WAIT/REQUEST}$ output can be defined under software control as a $\overline{WAIT}$ line in the CPU Block Transfer mode or as a $\overline{REQUEST}$ line in the DMA Block Transfer mode.

To a DMA controller, the ESCC $\overline{REQUEST}$ output indicates that the ESCC is ready to transfer data to or from memory. To the CPU, the $\overline{WAIT}$ line indicates that the SCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The $\overline{DTR/REQUEST}$ can be used as the transmit request line, thus allowing full-duplex operation under DMA control.

PROGRAMMING INFORMATION

Each channel has fifteen Write registers that are individually programmed from the system bus to configure the functional personality of each channel. Each channel also has eight Read registers from which the system can read Status, Baud rate, or Interrupt information.

On the Z85C30, only four data registers (Read and Write for Channels A and B) are directly selected by a HIGH on the D/$\overline{C}$ input and the appropriate levels on the $\overline{RD}$, $\overline{WR}$ and A/$\overline{B}$ pins. All other registers are addressed indirectly by the content of Write Register 0 in conjunction with a LOW on the D/$\overline{C}$ input and the appropriate levels on the $\overline{RD}$, $\overline{WR}$ and A/$\overline{B}$ pins. If bit D3 in WR0 is 1 and bits 5 and 6 are 0, then bits 0, 1, 2 address the higher registers 8 through 15. If bits 4, 5, 6 contain a different code, bits 0, 1, 2 address the lower registers 0 through 7 as shown in Table 2.

Writing to or reading from any register except RR0, WR0 and the Data Registers thus involves two operations:

First, write the appropriate code into WR0, then follow this by a write or read operation on the register thus specified. Bits 0 through 4 in WR0 are automatically cleared after this operation, so that WR0 then points to WR0 or RR0 again.

Channel A/Channel B selection is made by the A/$\overline{B}$ input (HIGH = A, LOW = B)

The system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, the asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set and, finally, receiver or transmitter enable.

Table 2. Register Addressing

D/$\overline{C}$	"Point High" Code In WR0:	D2, D1, D0 In WR0:			Write Register	Read Register
HIGH	Either Way	x	x	x	Data	Data
LOW	Not True	0	0	0	0	0
LOW	Not True	0	0	1	1	1
LOW	Not True	0	1	0	2	2
LOW	Not True	0	1	1	3	3
LOW	Not True	1	0	0	4	(0)
LOW	Not True	1	0	1	5	(1)
LOW	Not True	1	1	0	6	(2)
LOW	Not True	1	1	1	7	(3)
LOW	True	0	0	0	Data	Data
LOW	True	0	0	1	9	–
LOW	True	0	1	0	10	10
LOW	True	0	1	1	11	(15)
LOW	True	1	0	0	12	12
LOW	True	1	0	1	13	13
LOW	True	1	1	0	14	(10)
LOW	True	1	1	1	15	15

Read Registers

The ESCC contains eight read registers [actually nine, counting the receive buffer (RR8) in each channel]. Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). In addition, if bit D2 of WR15 is set, RR6 and RR7 are available for providing frame status from the 10 × 19 bit Frame Status FIFO. Figure 8 shows the formats for each read register.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring, for example, when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).

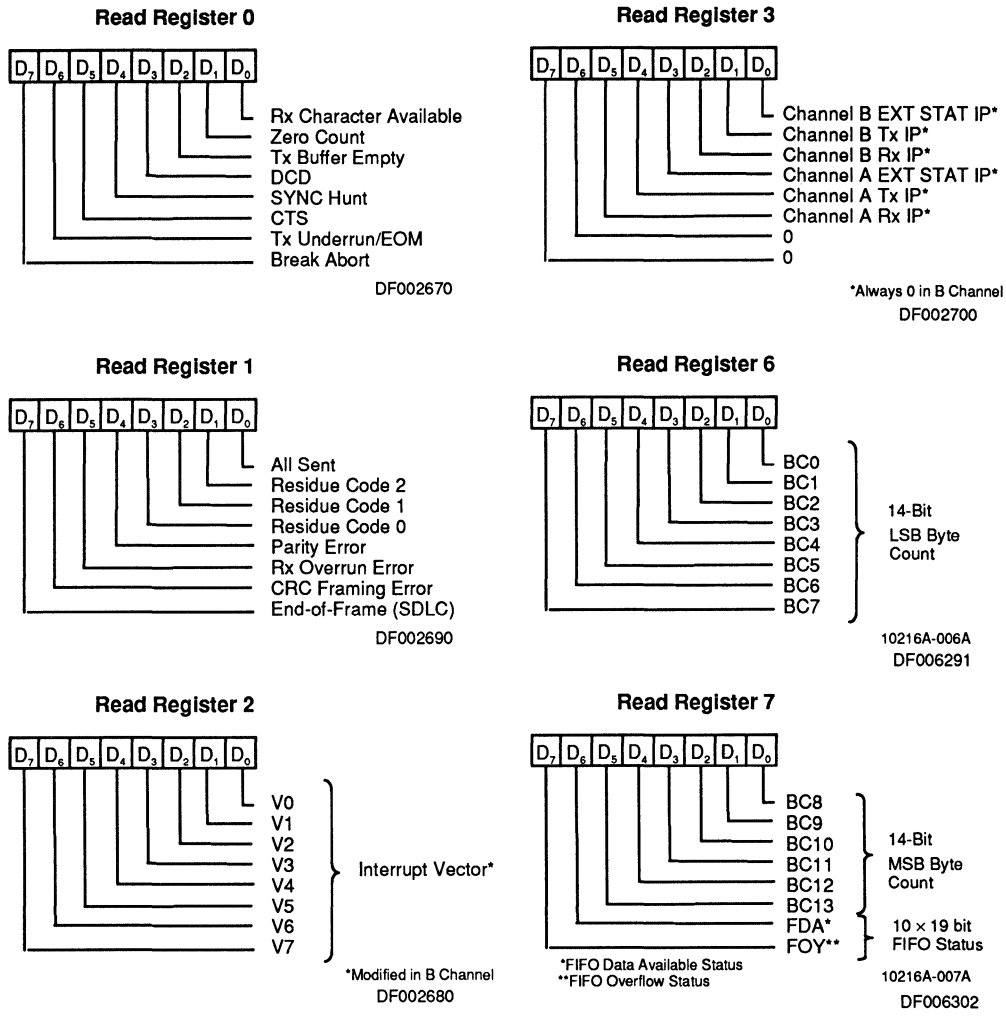


Figure 8. Read Register Bit Functions

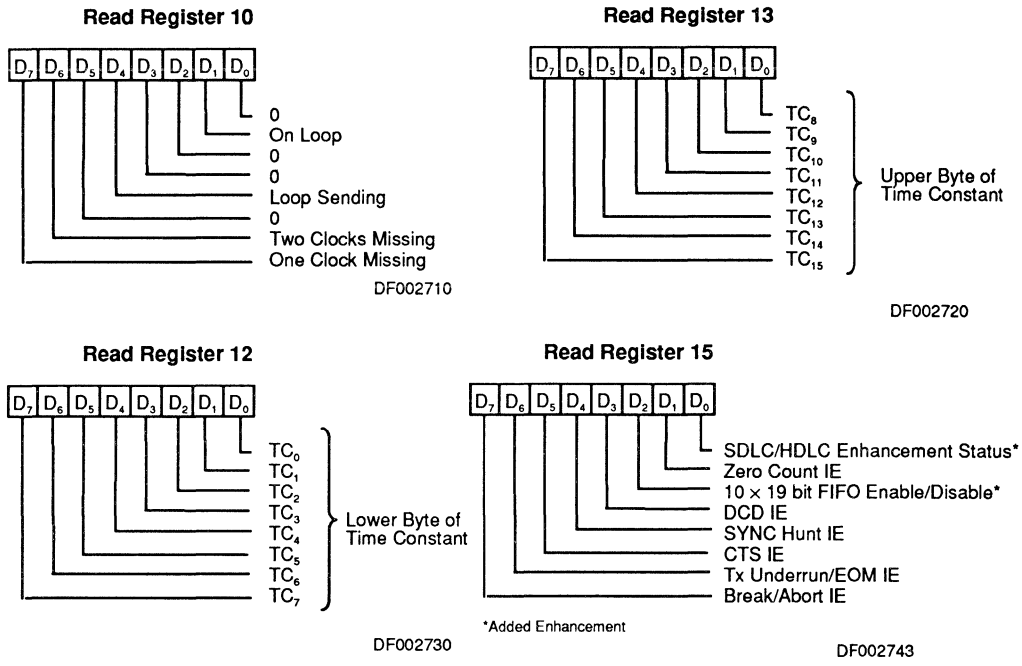


Figure 8. Read Register Bit Functions (continued)

Write Registers

The ESCC contains 15 write registers (16 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. Two registers (WR2 and WR9) are shared by the two channels that can be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. In addition, if bit D0 of WR15 is set, write register seven prime (WR7') is available for programming additional SDLC/HDLC enhancements. When bit D0 of WR15 is set, executing a write to WR7 actually writes to WR7' to further enhance the functional "personality" of each channel. Figure 9 shows the format of each write register.

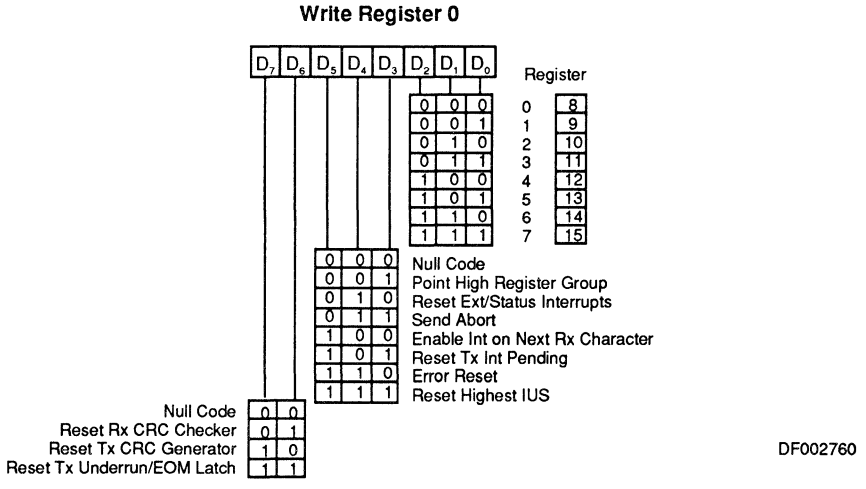


Figure 9. Write Register Bit Functions

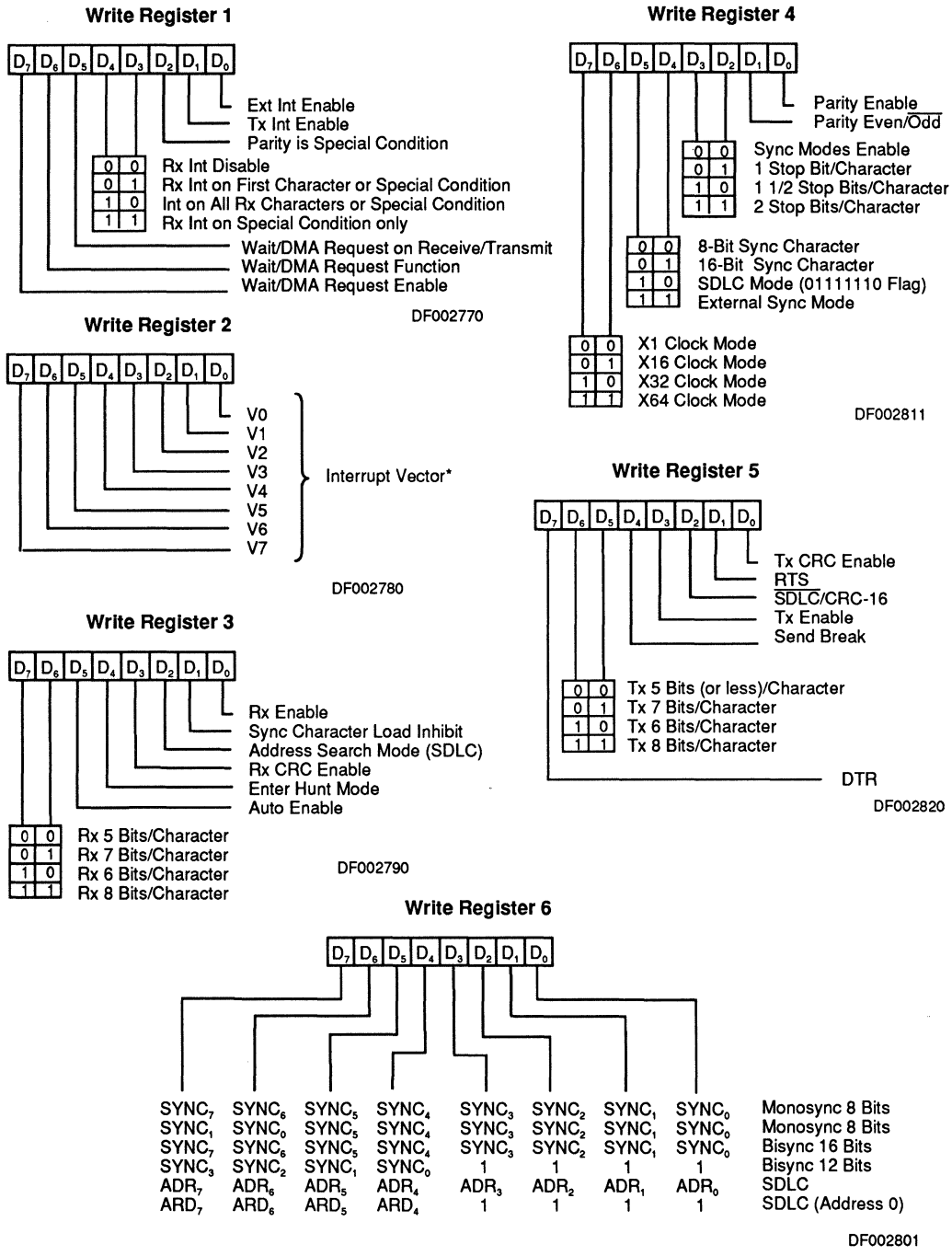
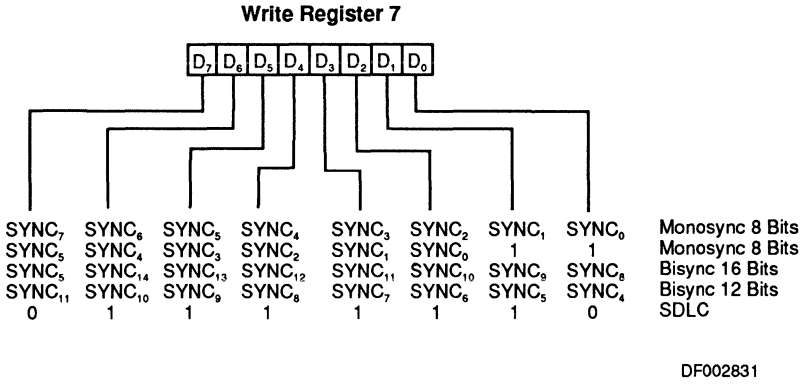
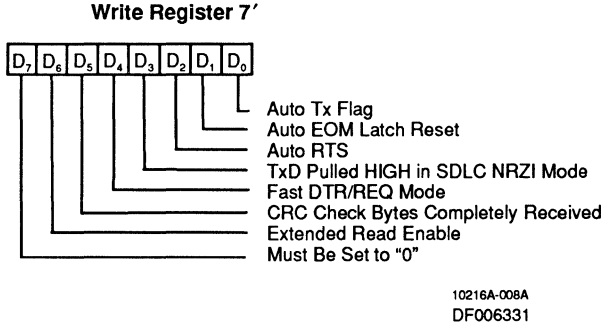


Figure 9. Write Register Bit Functions (continued)





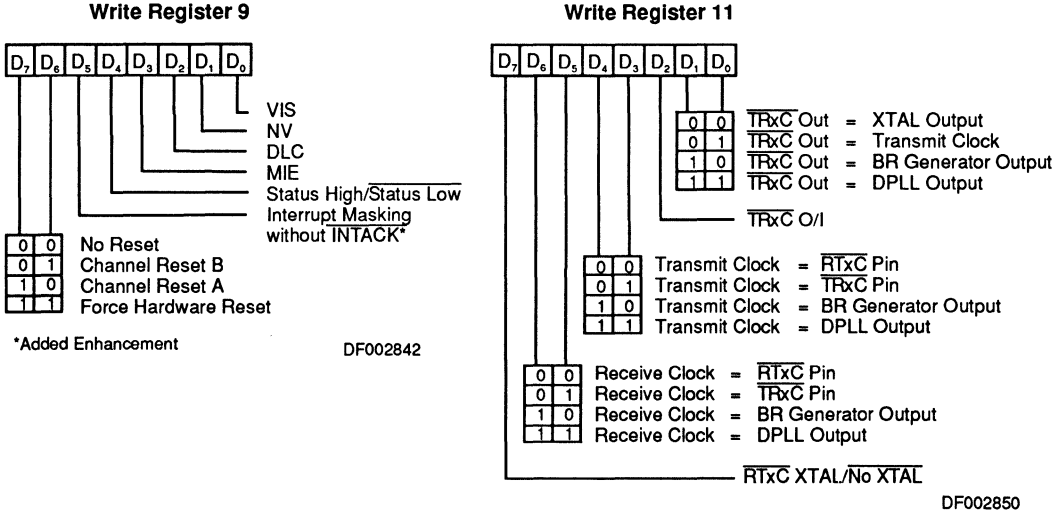


Figure 9. Write Register Bit Functions (continued)

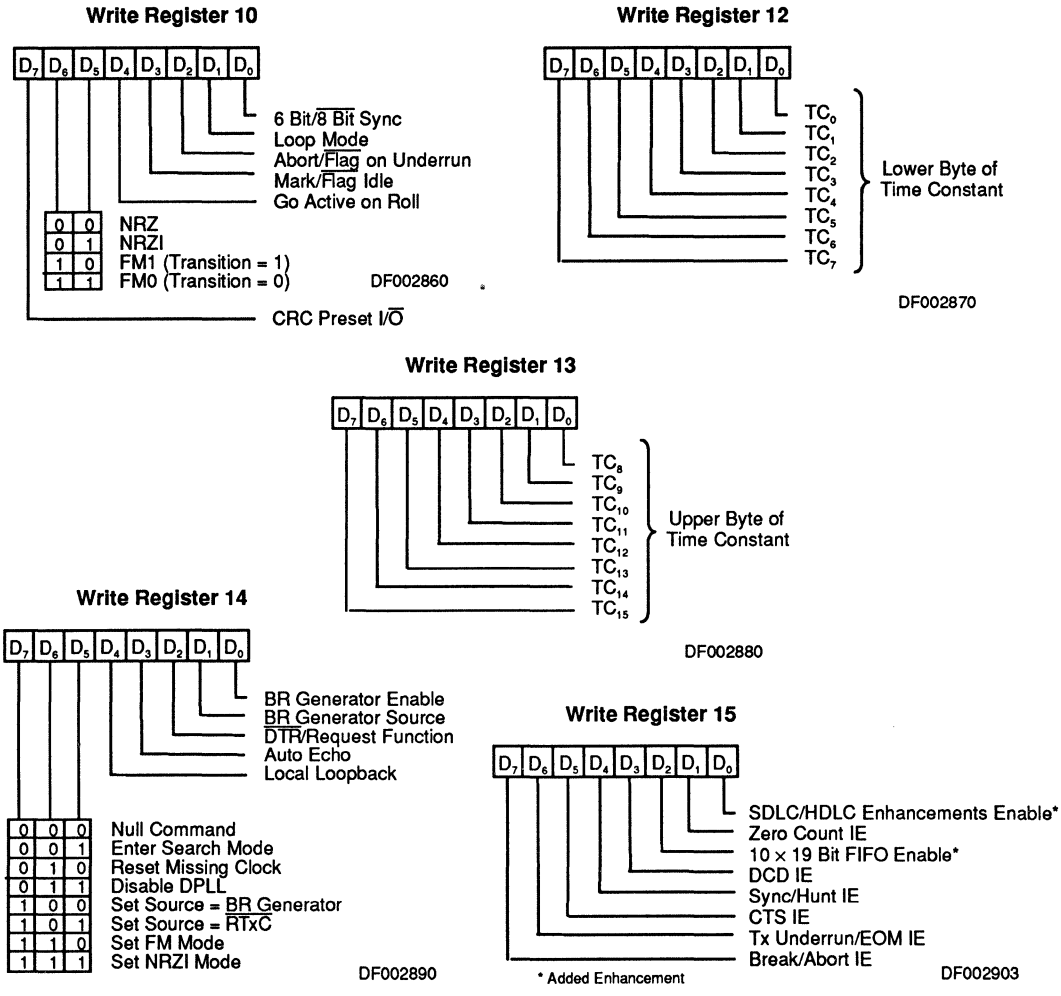


Figure 9. Write Register Bit Functions (continued)

Z85C30 Timing

The ESCC generates internal control signals from $\overline{WR}$ and $\overline{RD}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{WR}$ and $\overline{RD}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the ESCC. The recovery time required for proper operation is specified from the falling edge of $\overline{WR}$ or $\overline{RD}$ in the first transaction involving the ESCC, to the falling edge of $\overline{WR}$ or $\overline{RD}$ in the second transaction involving the ESCC. This time must be at least 3 1/2 PCLK regardless of which register or channel is being accessed.

Read Cycle Timing

Figure 10 illustrates Read cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{RD}$ falls or if it rises before $\overline{RD}$ rises, the effective $\overline{RD}$ is shortened.

Write Cycle Timing

Figure 11 illustrates Write Cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{WR}$ falls or if it rises before $\overline{WR}$ rises, the effective $\overline{WR}$ is shortened. Data must be valid before the rising edge of $\overline{WR}$.

Interrupt Acknowledge Cycle Timing

Figure 12 illustrates Interrupt Acknowledge cycle timing. Between the time $\overline{INTACK}$ goes LOW and the falling edge of $\overline{RD}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the ESCC and IEI is HIGH when $\overline{RD}$ falls, the Acknowledge cycle is intended for the SCC. In this case, the ESCC may be programmed to respond to $\overline{RD}$ LOW by placing its interrupt vector on D0–D7 and it then sets the appropriate Interrupt-Under-Service latch internally.

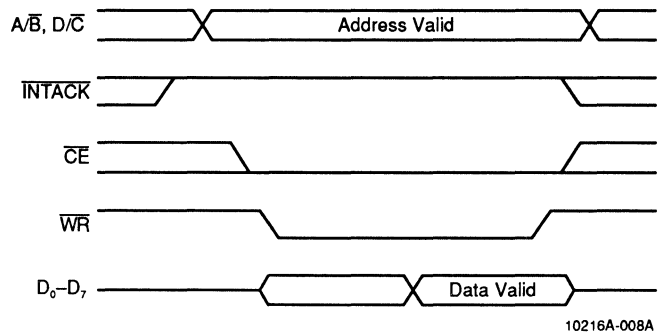


Figure 10. Read Cycle Timing

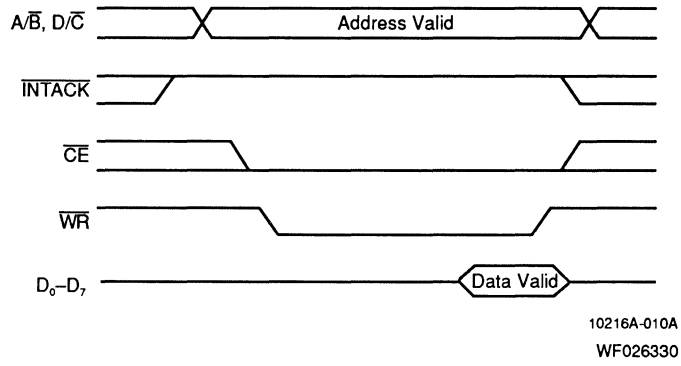


Figure 11. Write Cycle Timing

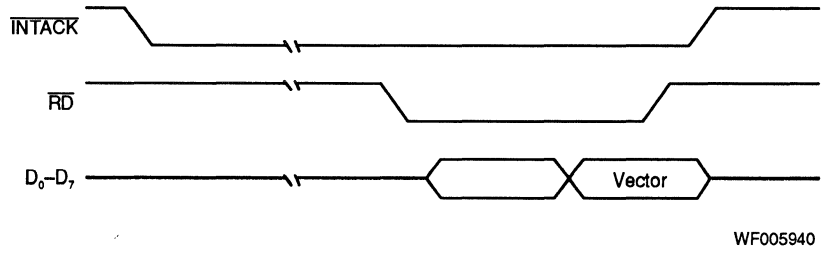


Figure 12. Interrupt Acknowledge Cycle Timing

FIFO

FIFO Enhancements

When used with a DMA controller, the Z85C30 Frame Status FIFO enhancement maximizes the ESCC's ability to receive high-speed back-to-back SDLC messages while minimizing frame overruns due to CPU latencies in responding to interrupts.

Additional logic was added to the industry-standard NMOS SCC consisting of a 10-deep by 19-bit status FIFO, a 14-bit receive byte counter, and control logic as shown in Figure 13. The 10 × 19 bit status FIFO is separate from the existing three-byte receive data and Error FIFOs.

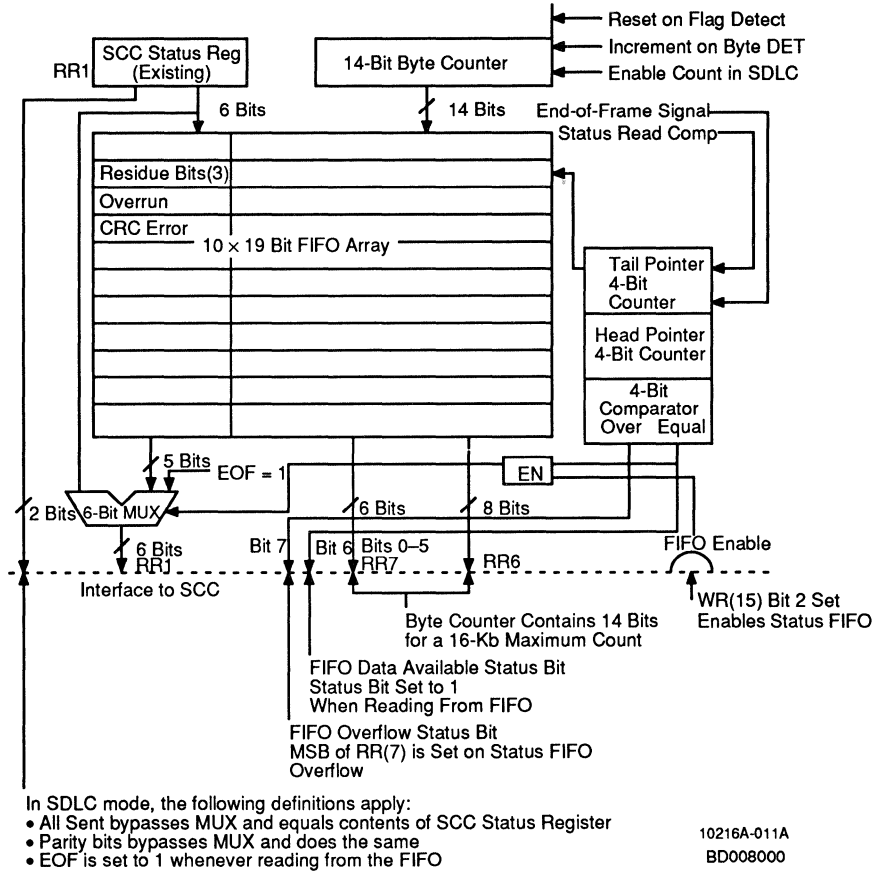


Figure 13. SCC Status Register Modifications

When the enhancement is enabled, the status in Read Register 1 (RR1) and byte count for the SDLC frame will be stored in the 10 × 19 bit status FIFO. This allows the DMA controller to transfer the next frame into memory while the CPU verifies that the message was properly received.

Summarizing the operation, data is received, assembled, and loaded into the three-byte receive FIFO before being transferred to memory by the DMA controller. When a flag is received at the end of an SDLC frame, the frame byte count from the 14-bit counter and five status bits are loaded into the status FIFO for verification by the CPU. The CRC checker is automatically reset in preparation for the next frame which can begin immediately.

Since the byte count and status are saved for each frame, the message integrity can be verified at a later time. Status information for up to 10 frames can be stored before a status FIFO overrun could occur.

If receive interrupts are enabled while the 10 × 19 FIFO is enabled, an SDLC end-of-frame special condition will not lock the three-byte Receive data FIFO. An SDLC end-of-frame still locks the three-byte Receive data FIFO in "Interrupt on first Receive Character or Special Condition" and "Interrupt on Special Condition Only" modes when the 10 × 19 FIFO is disabled. This feature allows the 10 × 19 SDLC FIFO to accept multiple SDLC frames without CPU intervention at the end of each frame.

FIFO Detail

For a better understanding of details of the FIFO operation, refer to the block diagram contained in Figure 13.

Enable/Disable

This FIFO is implemented so that it is enabled when WR15 bit 2 is set and the ESCC is in the SDLC/HDLC mode, otherwise the status register contents bypass the FIFO and go directly to the bus interface (the FIFO pointer logic is reset either when disabled or via a channel or power-on reset). When the FIFO mode is disabled, the ESCC is completely downward-compatible with the NMOS Z8530. The FIFO mode is disabled on power-up (WR15 bit 2 is set to "0" on reset). The effects of backward compatibility on the register set are that RR4 is an image of RR0, RR5 is an image of RR1, RR6 is an image of RR2, and RR7 is an image of RR3. For the details of the added registers, refer to Figure 15. The status of the FIFO Enable signal can be obtained by reading RR15 bit 2. If the FIFO is enabled, the bit will be set to "1"; otherwise, it will be reset.

Read Operation

When WR15 bit 2 is set and the FIFO is not empty, the next read to status register RR1 or the additional registers RR7 and RR6 will actually be from the FIFO. Reading status register RR1 causes one location of the FIFO to be emptied, so status should be read after reading the byte count, otherwise the count will be incorrect. Before the FIFO underflows, it is disabled. In this case, the multiplexer is switched to allow status to be read directly from the status register, and reads from RR7 and RR6 will contain bits that are undefined. Bit 6 of RR7 (FIFO Data Available) can be used to determine if status data is coming from the FIFO or directly from the status register, since it is set to "1" whenever the FIFO is not empty.

Because not all status bits are stored in the FIFO, the All Sent, Parity, and EOF bits will bypass the FIFO. The status bits sent through the FIFO will be Residue Bits (3), Overrun, and CRC Error.

The sequence for proper operation of the byte count and FIFO logic is to read the registers in the following order, RR7, RR6, and RR1 (reading RR6 is optional). Additional logic prevents the FIFO from being emptied by multiple reads from RR1. The read from RR7 latches the FIFO empty/full status bit (bit 6) and steers the status multiplexer to read from the SCC megacell instead of the status FIFO (since the status FIFO is empty). The read from RR1 allows an entry to be read from the FIFO (if the FIFO was empty, logic is added to prevent a FIFO underflow condition).

Write Operation

When the end of an SDLC frame (EOF) has been received and the FIFO is enabled, the contents of the status and byte-count registers are loaded into the FIFO. The EOF signal is used to increment the FIFO. If the FIFO overflows, the MSB of RR7 (FIFO Overflow) is set to indicate the overflow. This bit and the FIFO control logic is reset by disabling and re-enabling the FIFO control bit (WR15 bit 2). For details of FIFO control timing during an SDLC frame, refer to Figure 14.

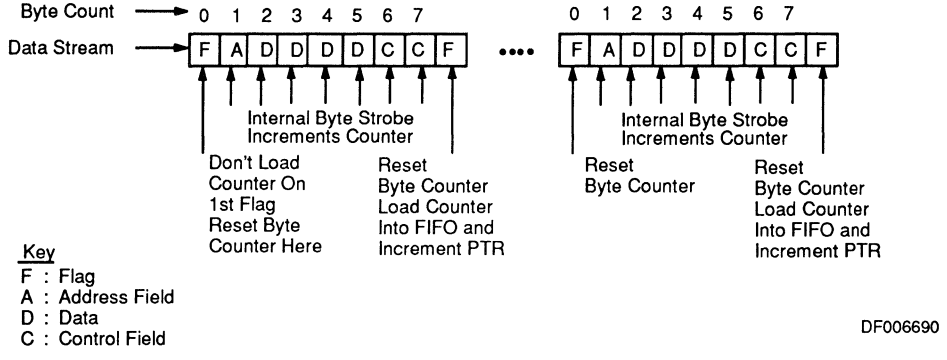


Figure 14. SDLC Byte Counting Detail

Byte Counter Detail

The 14-bit byte counter allows for packets up to 16K bytes to be received. For a better understanding of its operation, refer to Figures 13 and 14.

Enable

The byte counter is enabled when the SCC is in the SDLC/HDLC mode and WR15 bit 2 is set to "1."

Reset

The byte counter is reset whenever an SDLC flag character is received. The reset is timed so that the contents of the byte counter are successfully written into the FIFO.

Increment

The byte counter is incremented by writes to the data FIFO. The counter represents the number of bytes received by the SCC, rather than the number of bytes transferred from the SCC. (These counts may differ by up to the number of bytes in the receive data FIFO contained in the SCC.)

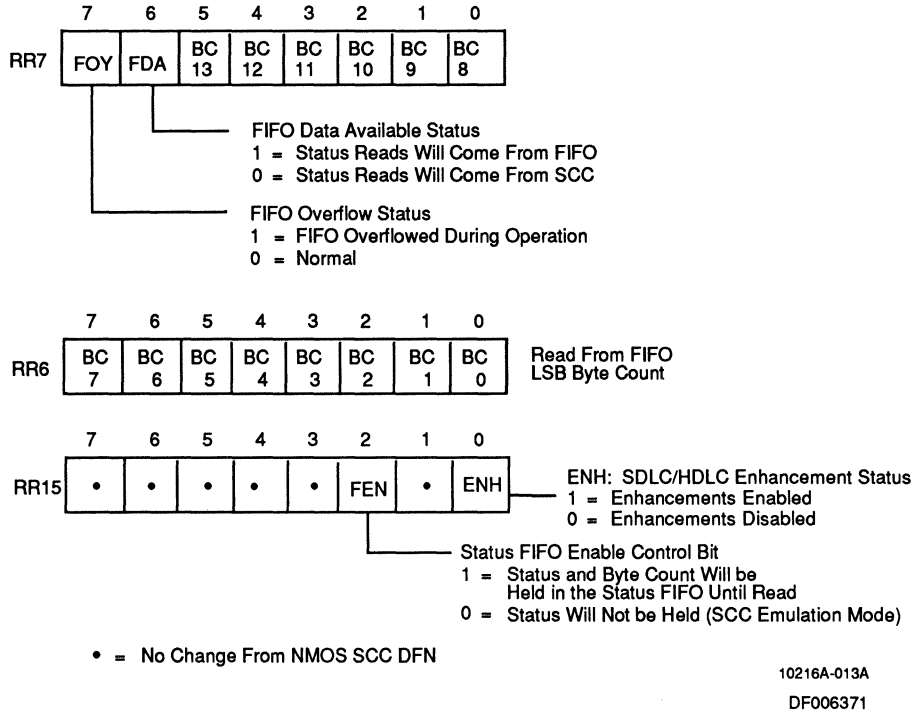


Figure 15. SCC Additional Registers

Z85C30 SDLC/HDLC Enhancement Register Access

SDLC/HDLC enhancements on the Z85C30 are enabled or disabled via bits D2 or D0 in WR15. Bit D2 determines whether or not the 10 × 19 bit SDLC/HDLC frame status FIFO is enabled while bit D0 determines whether or not other enhancements are enabled via WR7'. Table 3 shows what functions on the Z85C30 are enabled when these bits are set.

Table 3. Enhancement Options

WR15 Bit D2 10 × 19 Bit FIFO Enabled	WR15 Bit D0 SDLC/HDLC Enhancement Enabled	WR7' Bit D6 Extended Read Enabled	Functions Enabled
1	0	x	10 × 19 bit FIFO enhancement enabled only
0	1	0	SDLC/HDLC enhancements enabled only
0	1	1	SDLC/HDLC enhancements enabled with extended read enabled
1	1	0	10 × 19 bit FIFO and SDLC/HDLC enhancements enabled
1	1	1	10 × 19 bit FIFO and SDLC/HDLC enhancements with extended read enabled

When bit D2 of WR15 is set to "1," two additional registers (RR6 and RR7) per channel specific to the 10 × 19 bit Frame Status FIFO are made available. The Z85C30 register map when this function is enabled is shown in Table 4.

Table 4. 10 × 19 Bit FIFO Enabled

A/$\overline{B}$	PNT_2	PNT_1	PNT_0	Write	Read
0	0	0	0	WR0B	RR0B
0	0	0	1	WR1B	RR1B
0	0	1	0	WR2	RR2B
0	0	1	1	WR3B	RR3B
0	1	0	0	WR4B	(RR0B)
0	1	0	1	WR5B	(RR1B)
0	1	1	0	WR6B	RR6B
0	1	1	1	WR7B	RR7B
1	0	0	0	WR0A	RR0A
1	0	0	1	WR1A	RR1A
1	0	1	0	WR2	RR2A
1	0	1	1	WR3A	RR3A
1	1	0	0	WR4A	(RR0A)
1	1	0	1	WR5A	(RR1A)
1	1	1	0	WR6A	RR6A
1	1	1	1	WR7A	RR7A
		With the Point High command:			
0	0	0	0	WR8B	RR8B
0	0	0	1	WR9	RR13B
0	0	1	0	WR10B	RR10B
0	0	1	1	WR11B	(RR15B)
0	1	0	0	WR12B	RR12B
0	1	0	1	WR13B	RR13B
0	1	1	0	WR14B	(RR10B)
0	1	1	1	WR15B	RR15B
1	0	0	0	WR8A	RR8A
1	0	0	1	WR9	(RR13A)
1	0	1	0	WR10A	RR10A
1	0	1	1	WR11A	(RR15A)
1	1	0	0	WR12A	RR12A
1	1	0	1	WR13A	RR13A
1	1	1	0	WR14A	(RR10A)
1	1	1	1	WR15A	RR15A

Bit D0 of WR15 determines whether or not other enhancements pertinent only to SDLC/HDLC Mode operation are available for programming via WR7′ as shown below. Write Register 7 prime (WR7′) can be written to when bit D0 of WR15 is set to "1." When this bit is set, writing to WR7 (flag register) actually writes to WR7′. If bit D6 of this register is set to "1," previously unreadable registers WR3, WR4, WR5, and WR10 are readable by the processor. In addition, WR7′ is also readable by having this bit set. WR3 is read when a bogus RR9 register is accessed during a read cycle. WR10 is read by accessing RR11, and WR7′ is accessed by executing a read to RR14. The Z85C30 register map with bit D0 of WR15 and bit D6 of WR7′ set is shown in Table 5.

D7	D6	D5	D4	D3	D2	D1	D0
Must Be Set to 0	Ext. Read Enable	Rx comp. CRC	$\overline{DTR}/\overline{REQ}$ Fast Mode	Force TxD High	SDLC/HDLC Auto $\overline{RTS}$ Turnoff	SDLC/HDLC Auto EOM Reset	SDLC/HDLC Auto Tx Flag

WR7′—SDLC/HDLC Programmable Enhancements*

*Note: Options 3, 4, 5, and 6 may be used regardless of whether SDLC/HDLC mode is selected.

Table 5. SDLC/HDLC Enhancements Enabled

A/$\overline{B}$	PNT_2	PNT_1	PNT_0	Write	Read
0	0	0	0	WR0B	RR0B
0	0	0	1	WR1B	RR1B
0	0	1	0	WR2	RR2B
0	0	1	1	WR3B	RR3B
0	1	0	0	WR4B	RR4B (WR4B)
0	1	0	1	WR5B	RR5B (WR5B)
0	1	1	0	WR6B	(RR2B)
0	1	1	1	WR7B	(RR3B)
1	0	0	0	WR0A	RR0A
1	0	0	1	WR1A	RR1A
1	0	1	0	WR2	RR2A
1	0	1	1	WR3A	RR3A
1	1	0	0	WR4A	RR4A (WR4A)
1	1	0	1	WR5A	RR5A (WR5A)
1	1	1	0	WR6A	(RR2A)
1	1	1	1	WR7A	(RR3A)
				With the Point High command:	
0	0	0	0	WR8B	RR8B
0	0	0	1	WR9	RR9 (WR3B)
0	0	1	0	WR10B	RR10B
0	0	1	1	WR11B	RR11B (WR10B)
0	1	0	0	WR12B	RR12B
0	1	0	1	WR13B	RR13B
0	1	1	0	WR14B	RR14B (WR7'B)
0	1	1	1	WR15B	RR15B
1	0	0	0	WR8A	RR8A
1	0	0	1	WR9	RR9A (WR3A)
1	0	1	0	WR10A	RR10A
1	0	1	1	WR11A	RR11A (WR10A)
1	1	0	0	WR12A	RR12A
1	1	0	1	WR13A	RR13A
1	1	1	0	WR14A	RR14A (WR7A)
1	1	1	1	WR15A	RR15A

If both bits D0 and D2 of WR15 are set to "1" and D6 of WR7′ is set to "1," then the Z85C30 register map is as shown in Table 6.

Table 6. SDLC/HDLC Enhancements and 10 × 19 Bit FIFO Enabled

A/$\overline{B}$	PNT_2	PNT_1	PNT_0	Write	Read
0	0	0	0	WR0B	RR0B
0	0	0	1	WR1B	RR1B
0	0	1	0	WR2	RR2B
0	0	1	1	WR3B	RR3B
0	1	0	0	WR4B	RR4B (WR4B)
0	1	0	1	WR5B	RR5B (WR5B)
0	1	1	0	WR6B	RR6B
0	1	1	1	WR7B	RR7B
1	0	0	0	WR0A	RR0A
1	0	0	1	WR1A	RR1A
1	0	1	0	WR2	RR2A
1	0	1	1	WR3A	RR3A
1	1	0	0	WR4A	RR4A (WR4A)
1	1	0	1	WR5A	RR5A (WR5A)
1	1	1	0	WR6A	RR6A
1	1	1	1	WR7A	RR7A
				With the Point High command:	
0	0	0	0	WR8B	RR8B
0	0	0	1	WR9	RR9 (WR3B)
0	0	1	0	WR10B	RR10B
0	0	1	1	WR11B	RR11B (WR10B)
0	1	0	0	WR12B	RR12B
0	1	0	1	WR13B	RR13B
0	1	1	0	WR14B	RR14B (WR7'B)
0	1	1	1	WR15B	RR15B
1	0	0	0	WR8A	RR8A
1	0	0	1	WR9	RR9A (WR3A)
1	0	1	0	WR10A	RR10A
1	0	1	1	WR11A	RR11A (WR10A)
1	1	0	0	WR12A	RR12A
1	1	0	1	WR13A	RR13A
1	1	1	0	WR14A	RR14A (WR7'A)
1	1	1	1	WR15A	RR15A

Auto RTS Reset

On the CMOS ESCC, if bit D0 of WR15 and bit D2 of WR7' are set to "1" and the channel is in SDLC Mode, the $\overline{RTS}$ pin may be reset early in the Tx Underrun routine and the $\overline{RTS}$ pin will remain active until the last zero bit of the closing flag leaves the TxD pin as shown in Figure 16. Note that in order for this to function properly, bits D3 and D2 of WR10 must be set to "1" and "0" respectively.

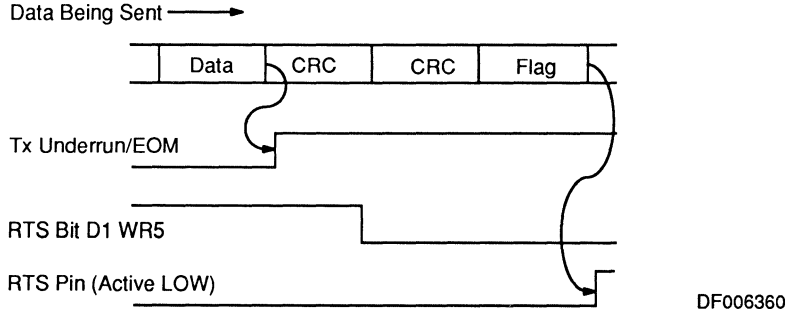


Figure 16. Auto $\overline{RTS}$ Reset Mode

CRC Character Reception

NMOS Z8530H

On the NMOS 8530H , when the end-of-frame flag is detected, the contents of the Receive Shift Register are transferred to the Receive Data FIFO regardless of the number of bits accumulated. Because of the 3-bit delay between the Receive SYNC Register and Receive Shift Register, the last two bits of the CRC check character received are never transferred to the Receive Data FIFO. Thus, the received CRC characters are unavailable for use.

CMOS Z85C30

On the Z85C30, the option of being able to receive the complete CRC characters generated by the Transmitter is provided when both bit D0 of WR15 and bit D5 of WR7' are set to "1." When these two bits are set and an end-of-frame flag is detected, the last two bits of the CRC will be clocked into the Receive Shift Register before its contents are transferred to the Receive Data FIFO.The data-CRC boundary and CRC character bit formatsfor each Residue Code provided is shown in Figures17A through 17D for each character length selected.

Residue Code 012/001

D	D	D	D	D	C_0	C_1	C_2
C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/101

D	D	D	D	D	D	C_0	C_1
D	C_0	C_1	C_2	C_3	C_4	C_5	C_6
C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/100

D	D	D	D	D	D	D	C_0
D	D	C_0	C_1	C_2	C_3	C_4	C_5
C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/010

D	D	D	D	D	D	D	D
D	D	D	C_0	C_1	C_2	C_3	C_4
C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/110

D	D	D	D	D	D	D	D
D	D	D	D	C_0	C_1	C_2	C_3
C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

10216A-015A
TB001170

Figure 17A. 5 Bits/Character

Residue Code 012/010

D	D	D	D	D	D	C_0	C_1
C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/110

D	D	D	D	D	D	D	C_0
D	C_0	C_1	C_2	C_3	C_4	C_5	C_6
C_5	6_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/001

D	D	D	D	D	D	D	D
D	D	C_0	C_1	C_2	C_3	C_4	C_5
C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/101

D	D	D	D	D	D	D	D
D	D	D	C_0	C_1	C_2	C_3	C_4
C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/011

D	D	D	D	D	D	D	D
D	D	D	D	C_0	C_1	C_2	C_3
C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/100

D	D	D	D	D	D	D	D
D	D	D	D	D	C_0	C_1	C_2
C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

10216A-016A
TB001180

Figure 17B. 6 Bits/Character

Residue Code 012 111

D	D	D	D	D	D	D	C_0
C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 100

D	D	D	D	D	D	D	D
D	C_0	C_1	C_2	C_3	C_4	C_5	C_6
C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 010

D	D	D	D	D	D	D	D
D	D	C_0	C_1	C_2	C_3	C_4	C_5
C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 110

D	D	D	D	D	D	D	D
D	D	D	C_0	C_1	C_2	C_3	C_4
C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 001

D	D	D	D	D	D	D	D
D	D	D	D	C_0	C_1	C_2	C_3
C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 101

D	D	D	D	D	D	D	D
D	D	D	D	D	C_0	C_1	C_2
C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012 011

D	D	D	D	D	D	D	D
D	D	D	D	D	D	C_0	C_1
C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

10216A-017A
TB001190

Figure 17C. 7 Bits/Character

Residue Code 012/011 (No Residue)

D	D	D	D	D	D	D	D
C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/111 (1 Residue Bit)

D	D	D	D	D	D	D	D
D	C_0	C_1	C_2	C_3	C_4	C_5	C_6
C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/000 (2 Residue Bits)

D	D	D	D	D	D	D	D
D	D	C_0	C_1	C_2	C_3	C_4	C_5
C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/100 (3 Residue Bits)

D	D	D	D	D	D	D	D
D	D	D	C_0	C_1	C_2	C_3	C_4
C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/010 (4 Residue Bits)

D	D	D	D	D	D	D	D
D	D	D	D	C_0	C_1	C_2	C_3
C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/110 (5 Residue Bits)

D	D	D	D	D	D	D	D
D	D	D	D	D	C_0	C_1	C_2
C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/001 (6 Residue Bits)

D	D	D	D	D	D	D	D
D	D	D	D	D	D	C_0	C_1
C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

Residue Code 012/101 (7 Residue Bits)

D	D	D	D	D	D	D	D
D	D	D	D	D	D	D	C_0
C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}

10216A-018A
TB001-200

Figure 17D. 8 Bits/Character

Auto Flag Mode

On the NMOS Z8530H, if the transmitter is actively mark idling and a frame of data is ready to be transmitted, the MARK/FLAG idle bit must be set to "0" before data is written to WR8, otherwise the opening flag will not be sent properly. However, care must be exercised in doing this because the mark idle pattern (eight "1" bits) is transmitted eight bits at a time, and all eight bits must have transferred out the Transmit Shift Register before a flag may be loaded and sent. If data is written into the Transmit Buffer (WR8) before the flag is loaded into the Transmit Shift Register, the data character written to WR8 will supersede flag transmission and the opening flag will not be transmitted.

On the CMOS Z85C30, if bit D0 of WR15 is set to "1," and the ESCC is programmed for SDLC operation, an option is provided via bit D0 of WR7′ that eliminates this requirement. If bit D0 of WR7′ is set to "1" and a character is written to the Transmit Buffer while the Transmitter is mark idling, the Mark/Flag Idle bit in WR10 need not be reset to "0" in order to have the opening flag sent because the Transmitter will automatically send it before commencing to send data.

In addition, as long as bit D0 of WR15 and bit D1 of WR7′ are set to "1," the CRC transmit generator will be automatically preset to the initial state programmed by bit D7 of WR10 (so the Reset Tx CRC Generator command is also not necessary), and the Tx Underrun/EOM latch will be reset automatically on every new frame sent. This ensures that an opening flag and proper CRC generation and transmission will always be sent without processor intervention under varying bus latency conditions.

Auto Transmit CRC Generator Preset

The NMOS Z8530H does not automatically preset the CRC generator prior to frame transmission. This must be done in software, usually during the initialization routine. This is accomplished by issuing the Reset Tx CRC Generator Command via WR0. For proper results, this command must be issued while the transmitter is enabled and idling and before any data are written to the Transmit Buffer.

In addition, if CRC is to be used, the transmit CRC generator must be enabled by setting bit D0 of WR5 to "1." CRC is normally calculated on all characters between opening and closing flags, so this bit should be set to "1" at initialization and never changed.

On the CMOS Z85C30, setting bit D0 of WR15 to "1" will cause the transmit CRC generator to be preset automatically every time an opening flag is sent, so the Reset Tx CRC Generator Command is not necessary.

Auto Tx Underrun/EOM Latch Reset

On the ESCC, the transmission of the CRC check characters is controlled by the Transmit CRC Enable bit in WR5 (D0) and the Tx Underrun/EOM bit in RR0 (D6). However, if the Transmit Enable bit is set to "0" when a transmit underrun (i.e., both the Transmit Buffer and Transmit Shift Register go empty) occurs, the CRC check characters will not be sent regardless of the state of the Tx Underrun/EOM bit.

If the Transmit Enable bit is set to "1" when an underrun occurs, then the state of the Tx Underrun/EOM bit and the Abort/Flag on Underrun bit in WR10 (D2) determine the action taken by the Transmitter. The Abort/Flag on Underrun bit may be set or reset by the processor, whereas, the Tx Underrun/EOM bit is set by the Transmitter and can only be reset by the processor via the Reset Tx Underrun/EOM Command in WR0.

If the Tx Underrun/EOM bit is set to "1" when an underrun occurs, the Transmitter will close the frame by sending a flag; however, if this bit is set to "0," the frame data will be appended with either the accumulated CRC characters followed by a flag or an abort pattern followed by a flag, depending on the state of the Abort/Flag on Underrun bit in the WR10 (D2). In either case, after the closing flag is sent, the Transmitter will idle the transmission line as specified by the Mark/Flag Idle bit D3 in WR10.

Hence, if the CRC check characters are to be properly appended to a frame, the Abort/Flag on Underrun bit must be set to "0," and the Reset Tx Underrun/EOM Command must be issued after the first but before the last character is written to the Transmit Buffer. This will ensure that either an abort or the CRC will be transmitted if an underrun occurs. Normally, the Abort/Flag on Underrun bit in WR10 should be set to "1" around the same time that the Tx Underrun/EOM bit is reset so that an abort will be sent if the transmitter accidentally underruns, and then set to "0" near the end of the frame to allow the correct transmission of CRC.

On the Z85C30, if bit D0 of WR15 is set to "1," the option of having the Tx Underrun/EOM bit reset automatically at the start of every frame is provided via bit D1 of WR7′. This helps alleviate the software burden of having to respond within one character time when high-speed data are being sent.

SDLC/HDLC NRZI Transmitter Disabling

On the NMOS Z8530H, if NRZI encoding is being used and the Transmitter is disabled, the state of the TxD pin will depend on the last bit sent. That is, the TxD pin may either idle in a Low or High state as shown in Figure 18.

On the CMOS Z85C30, an option is provided that allows setting the TxD pin High when operating in SDLC Mode with NRZI encoding enabled. If bit D0 of WR15 is set to "1," then bit D3 of WR7′ can be used to set the TxD pin High. Note that the operation of this bit is independent of the Tx Enable bit in WR5. The Tx Enable bit in WR5 is used to disable and enable the transmitter, whereas bit D3 of WR7′ acts as a pseudo transmitter disable and enable by just forcing the TxD pin High when set even though the transmitter may actually be mark or flag idling. Care must be used when setting this bit because any character being transmitted at the time this bit is set will be "chopped off," and data written to the Transmit Buffer while this bit is set will be lost.

When the transmit underrun occurs and the CRC and closing flag have been sent, bit D3 can be set to pull TxD High. When ready to start sending data again this bit must be reset to "0" before the first character is written to the Transmit Buffer. Note that resetting this bit causes the TxD pin to take whatever state the NRZI encoder is in at the time so synchronization at the Receiver may take longer because the first transition seen on the TxD pin may not coincide with a bit boundary. Note that in order for this to function properly, bits D3 and D2 of WR10 must be set to "1" and "0" respectively.

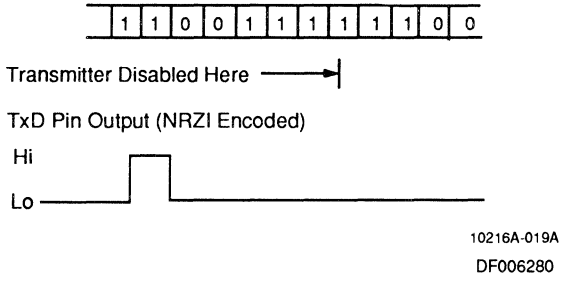


Figure 18. Transmitter Disabling with NRZI Encoding

Interrupt Masking Without $\overline{\text{INTACK}}$

The NMOS Z8530H's ability to mask lower priority interrupts is done via the IUS bit. This bit is internal to the SCC and is not observable by the processor. Being able to automatically mask lower priority interrupts allows a modular approach to coding interrupt routines. However, using the masking capabilities of the NMOS SCC requires that the $\overline{\text{INTACK}}$ cycle be generated. In standalone applications, having to generate $\overline{\text{INTACK}}$ through external hardware in order to use this capability is an unnecessary expense.

On the CMOS Z85C30, if bit D5 in WR9 is set to "1," the $\overline{\text{INTACK}}$ cycle does not need to be generated in order to have the IUS bit set. This allows the user to respond to ESCC interrupt requests with a software acknowledgment through RR2. When bit D5 in WR9 is set and an interrupt occurs, a read to RR2 emulates a hardware Interrupt Acknowledge cycle as it functions in Vectored Mode. In this case the CPU must first read RR2 to determine the internal interrupt source and then jump to the appropriate interrupt routine. Reading RR2 sets the IUS bit for the highest priority IP. After the interrupting condition is cleared, the routine can then read RR3 to determine if any other IPs are set and clear them. At the end of the interrupt routine, a Reset IUS Command must be issued to unlock the internal daisy chain.

Since the CPU can acknowledge the ESCC of highest priority with a read of its RR2 interrupt vector, there is no need for an external daisy chain. IEI for all ESCC devices should be tied active HIGH. When acknowledging an ESCC interrupt request, the CPU must issue one read to RR2 per interrupt request. The modified interrupt vector can be read from Channel B, or the original vector stored in WR2 can be read from Channel A. Either action will produce the same internal actions on the IUS logic. Note that the No Vector and Vector Includes Status bits in WR9 are ignored when bit D5 in WR9 is set to "1."

2 Mb/s FM Data Transmission and Reception

The 16-MHz version of the CMOS Z85C30 (Z85C30-16) is capable of transmitting and receiving FM-encoded data at the rate of 2 Mb/s. This is accomplished by applying a 32-MHz clock to the $\overline{\text{RTxC}}$ pin and assigning this waveform to drive the Internal Digital Phase Locked Loop (DPLL) clock. This feature allows the user to send both clock and data information over the same line at 2 Mb/s and can eliminate external DPLLs required for high-speed NRZ data clock generation.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	–65 to +150°C
Voltage at any Pin Relative to V_{SS}	–0.5 to +7.0 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices	
Ambient Temperature (T_A)	0 to +70°C
Supply Voltage (V_{CC})	+5 V ± 10%
Industrial (I) Devices	
Ambient Temperature (T_A)	–40 to +85°C
Supply Voltage (V_{CC})	5 V ±10%
Military (M) Devices	
Case Temperature (T_C)	–55° to 125°
Supply Voltage (V_{CC})	5 V ±10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{IH}	Input HIGH Voltage	Commercial	2.2	V_{CC} +0.3*	V
V_{IL}	Input LOW Voltage		–0.3*	0.8	V
V_{OH1}	Output HIGH Voltage	I_{OH} = –1.6 mA	2.4		V
V_{OH2}	Output HIGH Voltage	I_{OH} = –250 µA	V_{CC}–0.8		V
V_{OL}	Output LOW Voltage	I_{OL} = +2.0 mA		0.4	V
I_{IL}	Input Leakage	0.4 V ≤ V_{IN} ≤ 2.4 V		±10.0	µA
I_{OL}	Output Leakage	0.4 V ≤ V_{OUT} ≤ 2.4 V		±10.0	µA
I_{CC1}	V_{CC} Supply Current	8.192 MHz		18	mA
		10 MHz Inputs at		18	mA
		12 MHz voltage rails,		22	mA
		16.384 MHz output unloaded		22	mA
		20 MHz		30	mA
C_{IN}	Input Capacitance	Unmeasured pins returned		10	pF
C_{OUT}	Output Capacitance	to ground ! = 1 MHz over		15	pF
C_{MO}	Bidirectional Capacitance	specified temperature range		20	pF

* V_{IH} Max. and V_{IL} Min. not tested. Guaranteed by design.

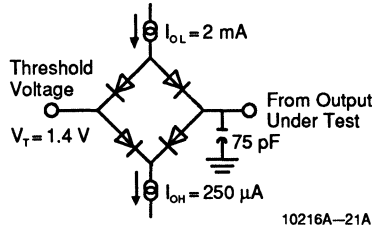
Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

+4.5 V ≤ V_{CC} ≤ +5.5 V
GND = 0 V
0°C ≤ T_A ≤ 70°C

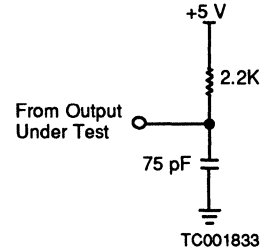
SWITCHING TEST CIRCUITS

Standard Test Dynamic Load Circuit



10216A–21A
TC004800

Open-Drain Test Load



TC001833

SWITCHING CHARACTERISTICS over COMMERCIAL operating range

General Timing (see Figure 19)

No.	Parameter Symbol	Parameter Description	8.192 MHz Min.	8.192 MHz Max.	10 MHz Min.	10 MHz Max.	12.5 MHz Min.	12.5 MHz Max.	Unit
1	TdPC(REQ)	PCLK ↓ to $\overline{W/REQ}$ Valid Delay		250		150		120	nsec
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		350		250		220	nsec
3	TsRXC(PC)	$\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4 & 8)	NA	NA	NA	NA	NA	NA	
4	TsRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Setup Time (XI Mode) (Note 1)	0		0		0		nsec
5	ThRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Hold Time (XI Mode) (Note 1)	150		125		100		nsec
6	TsRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Setup Time (XI Mode) (Notes 1, 5)	0		0		0		nsec
7	ThRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Hold Time (XI Mode) (Notes 1, 5)	150		125		100		nsec
8	TsSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Setup Time (Note 1)	−200		−150		−125		nsec
9	ThSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Hold Time (Note 1)	5TcPC		5TcPC		5TcPC		nsec
10	TsTXC(PC)	$\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8)	NA		NA		NA		
11	TdTXCf(TXD)	$\overline{TxC}$ ↓ to TxD Delay (XI Mode) (Note 2)		200		150		130	nsec
12	TdTXCr(TXD)	$\overline{TxC}$ ↑ to TxD Delay (XI Mode) (Notes 2, 5)		200		150		130	nsec
13	TdTXD(TRX)	TxD to $\overline{TRxC}$ Delay (Send Clock Echo)		200		140		120	nsec
14a	TwRTXh	$\overline{RTxC}$ HIGH Width (Note 6)	150		120		100		nsec
14b	TwRTxh(E)	$\overline{RTxC}$ HIGH Width (Note 9)	50		40		34		nsec
15a	TwRTXl	$\overline{RTxC}$ LOW Width (Note 6)	150		120		100		nsec
15b	TwRTXl(E)	$\overline{RTxC}$ LOW Width (Note 9)	50		40		34		nsec
16a	TcRTX	$\overline{RTxC}$ Cycle Time (Notes 6, 7)	488		400		320		nsec
16b	TcRTx(E)	$\overline{RTxC}$ Cycle Time (Note 9)	125		100		80		nsec
17	TcRTXX	Crystal Oscillator Period (Note 3)	125	1000	100	1000	80	1000	nsec
18	TwTRXh	$\overline{TRxC}$ HIGH Width (Note 6)	150		120		100		nsec
19	TwTRXl	$\overline{TRxC}$ LOW Width (Note 6)	150		120		100		nsec
20	TcTRX	$\overline{TRxC}$ Cycle Time (Notes 6, 7)	488		400		320		nsec
21	TwEXT	$\overline{DCD}$ or $\overline{CTS}$ Pulse Width	200		120		100		nsec
22	TwSY	$\overline{SYNC}$ Pulse Width	200		120		100		nsec

Notes:
1. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{SYNC}$ have 30-pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{RxC}$ and PCLK or $\overline{TxC}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver, DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to $\overline{RxC}$ or $\overline{TxC}$ synchronization requirement eliminated for PCLK divide-by-four operation.

   $\overline{TRxC}$ and $\overline{RTxC}$ rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

   Tx and Rx input clock slow rates should be kept to a maximum of 30 nsec. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is worst case.
9. ENHANCED FEATURE— $\overline{RTxC}$ used as input to internal DPLL only.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

General Timing (see Figure 19)

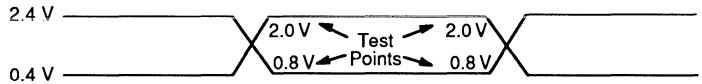
No.	Parameter Symbol	Parameter Description	16.384 MHz Min.	16.384 MHz Max.	20 MHz Min.	20 MHz Max.	Unit
1	TdPC(REQ)	PCLK ↓ to W/REQ Valid Delay		80		70	nsec
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		180		170	nsec
3	TsRXC(PC)	RxC ↑ to PCLK ↑ Setup Time (Notes 1, 4 & 8)	NA	NA	NA	NA	
4	TsRXD(RXCr)	RxD to RxC ↑ Setup Time (XI Mode) (Note 1)	0		0		nsec
5	ThRXD(RXCr)	RxD to RxC ↑ Hold Time (XI Mode) (Note 1)	50		45		nsec
6	TsRXD(RXCf)	RxD to RxC ↓ Setup Time (XI Mode) (Notes 1, 5)	0		0		nsec
7	ThRXD(RXCf)	RxD to RxC ↓ Hold Time (XI Mode) (Notes 1, 5)	50		45		nsec
8	TsSY(RXC)	SYNC to RxC ↑ Setup Time (Note 1)	–100		–90		nsec
9	ThSY(RXC)	SYNC to RxC ↑ Hold Time (Note 1)	5TcPc		5TcPc		nsec
10	TsTXC(PC)	TxC ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8)	NA		NA		
11	TdTXCf(TXD)	TxC ↓ to TxD Delay (XI Mode) (Note 2)		80		70	nsec
12	TdTXCr(TXD)	TxC ↑ to TxD Delay (XI Mode) (Notes 2, 5)		80		70	nsec
13	TdTXD(TRX)	TxD to TRxC Delay (Send Clock Echo)		80		70	nsec
14a	TwRTXh	RTxC HIGH Width (Note 6)	80		70		nsec
14b	TwRTxh(E)	RTxC HIGH Width (Note 9)	15.6		15.6		nsec
15a	TwRTXl	RTxC LOW Width (Note 6)	80		70		nsec
15b	TwRTXl(E)	RTxC LOW Width (Note 9)	15.6		15.6		nsec
16a	TcRTX	RTxC Cycle Time (Notes 6, 7)	244		200		nsec
16b	TcRTx(E)	RTxC Cycle Time (Note 9)	31.25		31.25		nsec
17	TcRTXX	Crystal Oscillator Period (Note 3)	62	1000	50	1000	nsec
18	TwTRXh	TRxC HIGH Width (Note 6)	80		70		nsec
19	TwTRXl	TRxC LOW Width (Note 6)	80		70		nsec
20	TcTRX	TRxC Cycle Time (Notes 6, 7)	244		200		nsec
21	TwEXT	DCD or CTS Pulse Width	70		65		nsec
22	TwSY	SYNC Pulse Width	70		65		nsec

Notes:
1. RxC is RTxC or TRxC, whichever is supplying the receive clock.
2. TxC is TRxC or RTxC, whichever is supplying the transmit clock.
3. Both RTxC and SYNC have 30-pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between RxC and PCLK or TxC and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver, DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to RxC or TxC synchronization requirement eliminated for PCLK divide-by-four operation.

   TRxC and RTxC rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

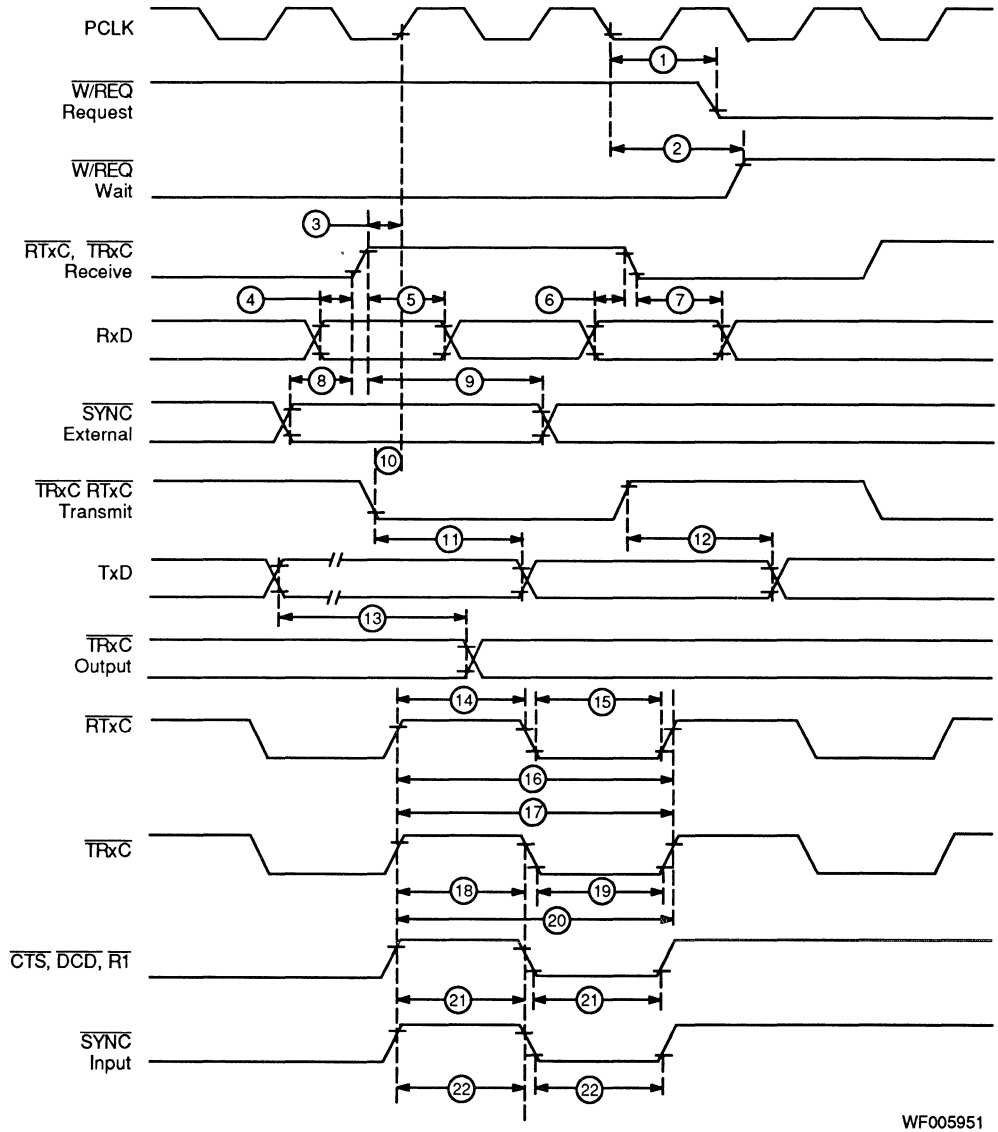
   Tx and Rx input clock slow rates should be kept to a maximum of 30 nsec. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is worst case.
9. ENHANCED FEATURE— RTxC used as input to internal DPLL only.

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WR006354

AC testing: Inputs are driven at 2.4 V for a logic "1" and 0.4 V for a logic "0."
Timing measurements are made at 2.0 V for a logic "1" and 0.8 V for logic "0."



WF005951

Figure 19. General Timing

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

System Timing (see Figure 20)

No.	Parameter Symbol	Parameter Description	8.192 MHz Min.	8.192 MHz Max.	10 MHz Min.	10 MHz Max.	12.5 MHz Min.	12.5 MHz Max.	Unit
1	TdRXC(REQ)	$\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	8	12	TcPc
2	TdRXC(W)	$\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	8	14	8	14	TcPc
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	4	7	TcPc
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	10	16	TcPc
5	TdTXC(REQ)	$\overline{TxC}$ ↑ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	5	8	TcPc
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	5	11	TcPc
7a	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	4	7	TcPc
7b	TdTXC(EDRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Notes 3, 4)	5	8	5	8	5	8	TcPc
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	6	10	TcPc
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	2	6	TcPc
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	2	6	TcPc

No.	Parameter Symbol	Parameter Description	16.384 MHz Min.	16.384 MHz Max.	20 MHz Min.	20 MHz Max.	Unit
1	TdRXC(REQ)	$\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	TcPc
2	TdRXC(W)	$\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	8	14	TcPc
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	TcPc
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	TcPc
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	TcPc
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	TcPc
7a	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	TcPc
7b	TdTXC(EDRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Notes 3, 4)	5	8	5	8	TcPc
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	TcPc
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPc
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPc

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
4. Parameter applies to Enhanced Request mode only.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

Read and Write Timing (see Figure 21)

No.	Parameter Symbol	Parameter Description	8.192 MHz		10 MHz		12.5 MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
1	TwPCl	PCLK LOW Width	50	2000	40	2000	34	2000	nsec
2	TwPCh	PCLK HIGH Width	50	2000	40	2000	34	2000	nsec
3	TfPC	PCLK Fall Time		15		12		10	nsec
4	TrPC	PCLK Rise Time		15		12		10	nsec
5	TcPC	PCLK Cycle Time	122	4000	100	4000	80	4000	nsec
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	70		50		45		nsec
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		0		nsec
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	70		50		45		nsec
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		0		nsec
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	20		20		15		nsec
11	TsIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1)	145		120		95		nsec
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		0		nsec
13	TsIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time (Note 1)	145		120		95		nsec
14	ThIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		0		nsec
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	40		30		20		nsec
16	TsCEl(WR)	$\overline{CE}$ LOW to $\overline{WR}$ ↓ Setup Time	0		0		0		nsec
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		0		nsec
18	TsCEh(WR)	$\overline{CE}$ HIGH to $\overline{WR}$ ↓ Setup Time	60		50		40		nsec
19	TsCEl(RD)	$\overline{CE}$ LOW to $\overline{RD}$ ↓ Setup Time (Note 1)	0		0		0		nsec
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note1)	0		0		0		nsec
21	TsCEh(RD)	$\overline{CE}$ HIGH to $\overline{RD}$ ↓ Setup Time (Note 1)	60		50		40		nsec
22	TwRDl	$\overline{RD}$ LOW Width (Note 1)	150		125		90		nsec
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		0		nsec
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		0		nsec
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		140		120		85	nsec
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		40		35		25	nsec

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC Load and minimum AC load.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

Read and Write Timing (see Figure 21)

No.	Parameter Symbol	Parameter Description	16.384 MHz		20 MHz		Unit
			Min.	Max.	Min.	Max.	
1	TwPCl	PCLK LOW Width	26	2000	22	1000	nsec
2	TwPCh	PCLK HIGH Width	26	2000	22	1000	nsec
3	TfPC	PCLK Fall Time		8		5	nsec
4	TrPC	PCLK Rise Time		8		5	nsec
5	TcPC	PCLK Cycle Time	61	4000	50	2000	nsec
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	35		30		nsec
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		nsec
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	35		30		nsec
9	ThA(RD)	Address to RD ↑ Hold Time	0		0		nsec
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	15		15		nsec
11	TsIAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1)	70		65		nsec
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		nsec
13	TsIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time (Note 1)	70		65		nsec
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		nsec
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	15		15		nsec
16	TsCEl(WR)	$\overline{CE}$ LOW to $\overline{WR}$ ↓ Setup Time	0		0		nsec
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		nsec
18	TsCEh(WR)	$\overline{CE}$ HIGH to $\overline{WR}$ ↓ Setup Time	30		25		nsec
19	TsCEl(RD)	$\overline{CE}$ LOW to $\overline{RD}$ ↓ Setup Time (Note 1)	0		0		nsec
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note1)	0		0		nsec
21	TsCEh(RD)	$\overline{CE}$ HIGH to $\overline{RD}$ ↓ Setup Time (Note 1)	30		25		nsec
22	TwRDl	$\overline{RD}$ LOW Width (Note 1)	75		65		nsec
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		nsec
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		nsec
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		70		60	nsec
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		20		20	nsec

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC load and minimum AC load.

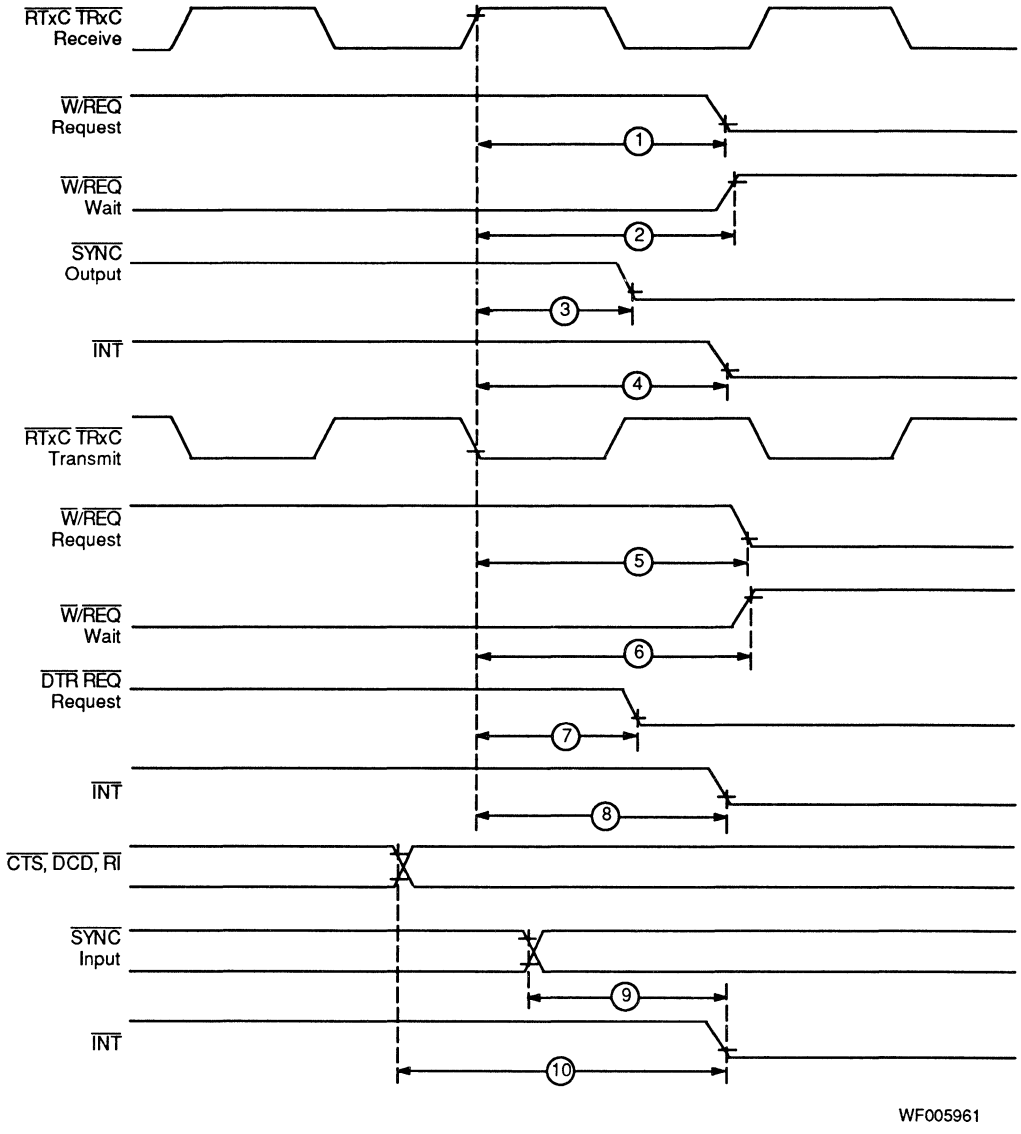


Figure 20. System Timing

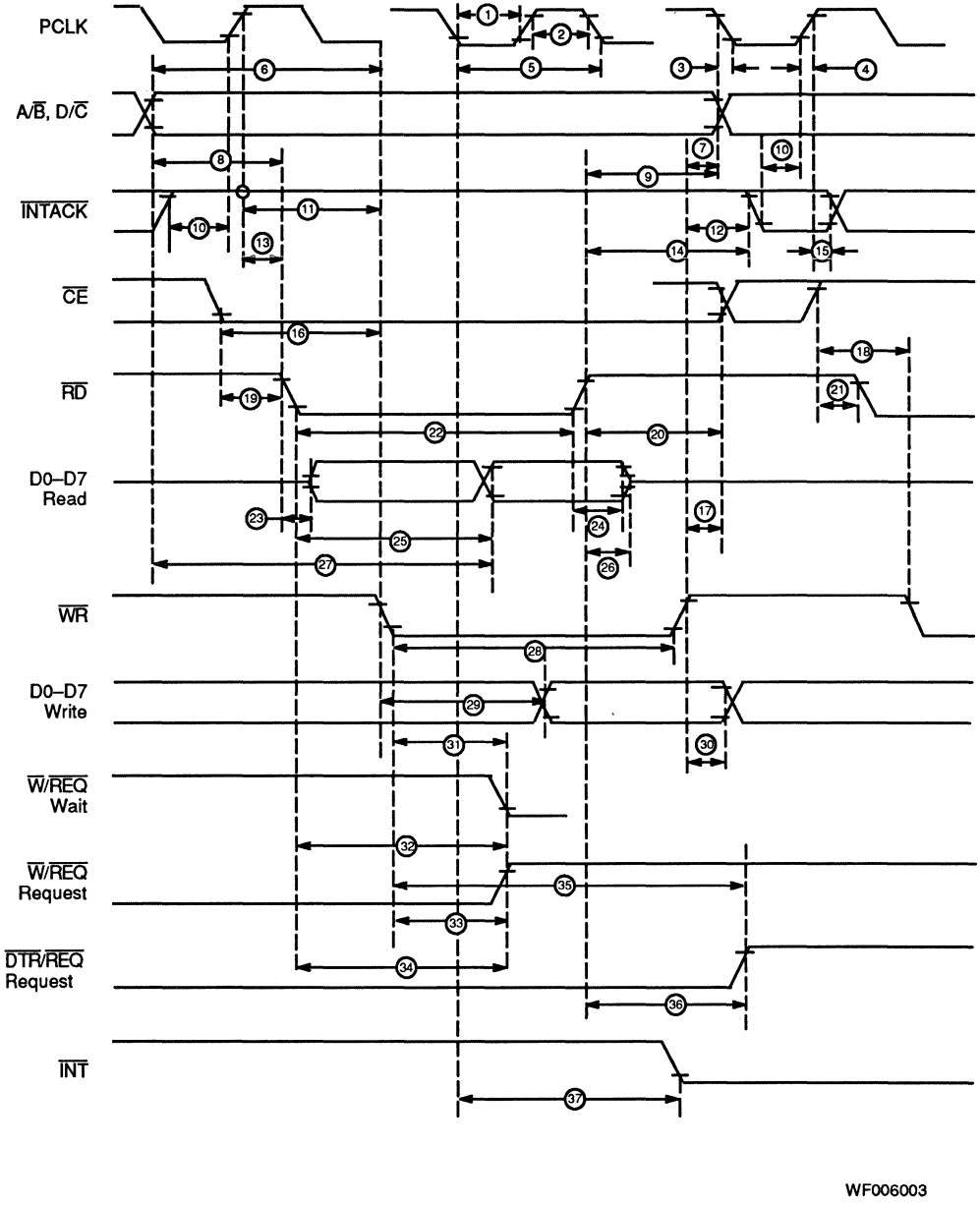


Figure 21. Read and Write Timing

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 22–24)

No.	Parameter Symbol	Parameter Description	8.192 MHz Min.	8.192 MHz Max.	10 MHz Min.	10 MHz Max.	12.5 MHz Min.	12.5 MHz Max.	Unit
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		220		160		120	nsec
28	TwWRI	$\overline{WR}$ LOW Width	150		125		90		nsec
29	TdWRf(DW)	$\overline{WR}$ ↓ to Write Data Valid		35		35		25	nsec
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		0		nsec
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 2)		170		100		70	nsec
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 2)		170		100		70	nsec
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		170		120		100	nsec
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		170		120		100	nsec
35a	TdWRr(REQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		4.0TcPc		4.0TcPc		4.0TcPc	nsec
35b	TdWRr(EREQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		120		120		100	nsec
36	TdRDr(REQ)	$\overline{RD}$ ↑ $\overline{DTR/REQ}$ Not Valid Delay		NA		NA		NA	nsec
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 2)		500		400		350	nsec
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 3)	150		125		95		nsec
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	150		125		95		nsec
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		140		120		90	nsec
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	95		80		65		nsec
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		0		nsec
43	TdIEI(IEO)	IEI to IEO Delay Time		95		80		65	nsec
44	TdPC(IEO)	PCLK ↑ to IEO Delay		200		175		130	nsec
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 2)		450		320		260	nsec
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	15		15		10		nsec
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	15		15		10		nsec
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident LOW for Reset	150		100		85		nsec
49	Trc	Valid Access Recovery Time (Note 1)	3.5		3.5		3.5		TcPc

Notes:
1. Parameter applies only between transactions involving the ESCC, if WR/RD falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of DdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.
4. Parameter applies to Enhanced Request mode only.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 22–24)

No.	Parameter Symbol	Parameter Description	16.384 MHz Min.	16.384 MHz Max.	20 MHz Min.	20 MHz Max.	Unit
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		100		90	nsec
28	TwWRI	$\overline{WR}$ LOW Width	75		65		nsec
29	TdWRf(DW)	$\overline{WR}$ ↓ to Write Data Valid		20		20	nsec
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		nsec
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 2)		50		45	nsec
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 2)		50		45	nsec
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		70		65	nsec
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		70		65	nsec
35a	TdWRr(REQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		4.0TcPc		4.0TcPc	nsec
35b	TdWRr(EREQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay (Note 4)		70		65	nsec
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		NA		NA	nsec
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 2)		175		160	nsec
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 3)	50		45		nsec
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	75		65		nsec
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		70		60	nsec
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	50		45		nsec
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		nsec
43	TdIEI(IEO)	IEI to IEO Delay Time		45		40	nsec
44	TdPC(IEO)	PCLK ↑ to IEO Delay		80		70	nsec
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 2)		200		180	nsec
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	10		10		nsec
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	10		10		nsec
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident LOW for Reset	75		65		nsec
49	Trc	Valid Access Recovery Time (Note 1)	3.5		3.5		TcPc

Notes:
1. Parameter applies only between transactions involving the ESCC. If $\overline{WR}/\overline{RD}$ falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.
4. Parameter applies to Enhanced Request mode only.

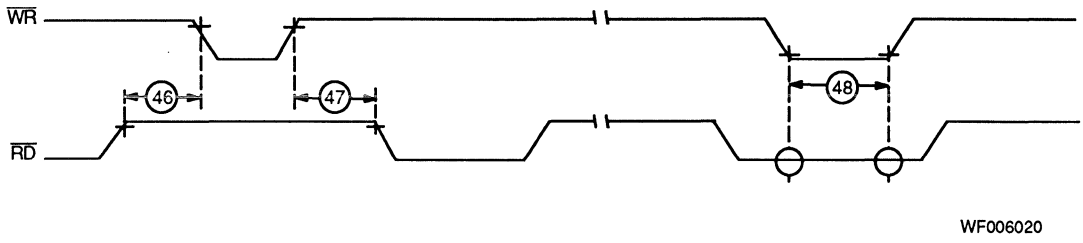


Figure 22. Reset Timing

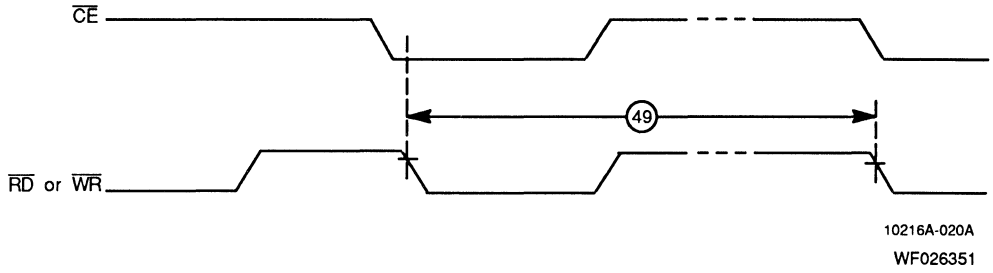


Figure 23. Cycle Timing

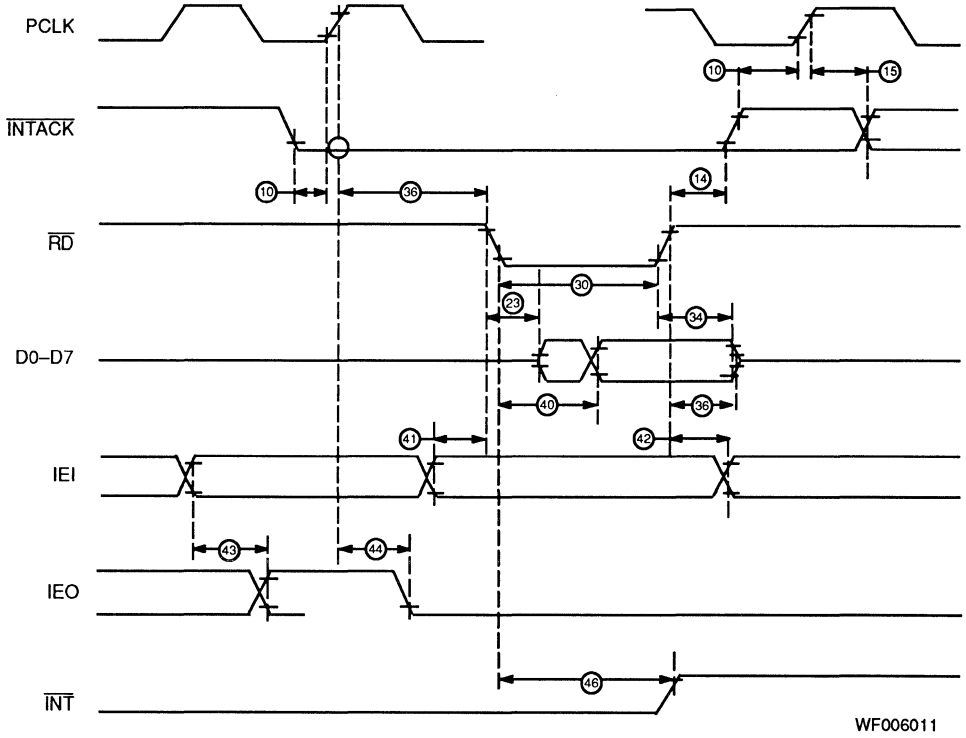


Figure 24. Interrupt Acknowledge Timing

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range

General Timing (see Figure 19)

No.	Parameter Symbol	Parameter Description	8.192 MHz Min.	8.192 MHz Max.	10 MHz Min.	10 MHz Max.	Unit
1	TdPC(REQ)	PCLK ↓ to $\overline{W/REQ}$ Valid Delay		250		150	nsec
2	TdPC(W)	PLCK ↓ to Wait Inactive Delay		350		250	nsec
3	TsRXC(PC)	$\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4 & 8)	NA	NA	NA	NA	
4	TsRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Setup Time (XI Mode) (Note 1)	0		0		nsec
5	ThRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Hold Time (XI Mode) (Note 1)	150		125		nsec
6	TsRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Setup Time (XI Mode) (Notes 1, 5)	0		0		nsec
7	ThRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Hold Time (XI Mode) (Notes 1, 5)	150		125		nsec
8	TsSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Setup Time (Note 1)	–200		–150		nsec
9	ThSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Hold Time (Note 1)	5TcPc		5TcPc		nsec
10	TsTXC(PC)	$\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8)	NA		NA		
11	TdTXCf(TXD)	$\overline{TxC}$ ↓ to TxD Delay (XI Mode) (Note 2)		200		150	nsec
12	TdTXCr(TXD)	$\overline{TxC}$ ↑ to TxD Delay (XI Mode) (Notes 2, 5)		200		150	nsec
13	TdTXD(TRX)	TxD to $\overline{TRxC}$ Delay (Send Clock Echo)		200		140	nsec
14a	TwRTXh	$\overline{RTxC}$ HIGH Width (Note 6)	150		120		nsec
14b	TwRTXh(E)	$\overline{RTxC}$ HIGH Width (Note 9)	50		40		nsec
15a	TwRTXl	$\overline{RTxC}$ LOW Width (Note 6)	150		120		nsec
15b	TwRTXl(E)	$\overline{RTxC}$ LOW Width (Note 9)	50		40		nsec
16a	TcRTX	$\overline{RTxC}$ Cycle Time (Notes 6, 7)	488		400		nsec
16b	TcRTx(E)	$\overline{RTxC}$ Cycle Time (Note 9)	125		100		nsec
17	TXRTXX	Crystal Oscillator Period (Note 3)	125	1000	100	1000	nsec
18	TwTRXh	$\overline{TRxC}$ HIGH Width (Note 6)	150		120		nsec
19	TwTRXl	$\overline{TRxC}$ LOW Width (Note 6)	150		120		nsec
20	TcTRX	$\overline{TRxC}$ Cycle Time (Notes 6, 7)	488		400		nsec
21	TwEXT	$\overline{DCD}$ or $\overline{CTS}$ Pulse Width	200		120		nsec
22	TwSY	$\overline{SYNC}$ Pulse Width	200		120		nsec

Notes:
1. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{SYNC}$ have 30-pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{RxC}$ and PCLK or $\overline{TxC}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver, DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to $\overline{RxC}$ or $\overline{TxC}$ synchronization requirement eliminated for PCLK divide-by-four operation.

   $\overline{TRxC}$ and $\overline{RTxC}$ rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

   Tx and Rx input clock slow rates should be kept to a maximum of 30 nsec. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is worst case.
9. ENHANCED FEATURE— $\overline{RTxC}$ used as input to internal DPLL only.

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

General Timing (see Figure 19)

No.	Parameter Symbol	Parameter Description	12.5 MHz Min.	12.5 MHz Max.	16.384 MHz Min.	16.384 MHz Max.	Unit
1	TdPC(REQ)	PCLK ↓ to W/$\overline{REQ}$ Valid Delay		120		80	nsec
2	TdPC(W)	PLCK ↓ to Wait Inactive Delay		220		180	nsec
3	TsRXC(PC)	$\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4 & 8)	NA	NA	NA	NA	
4	TsRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Setup Time (XI Mode) (Note 1)	0		0		nsec
5	ThRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Hold Time (XI Mode) (Note 1)	100		50		nsec
6	TsRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Setup Time (XI Mode) (Notes 1, 5)	0		0		nsec
7	ThRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Hold Time (XI Mode) (Notes 1, 5)	100		50		nsec
8	TsSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Setup Time (Note 1)	−125		−100		nsec
9	ThSY(RXC)	$\overline{SYNC}$ to $\overline{RxC}$ ↑ Hold Time (Note 1)	5TcPc		5TcPc		nsec
10	TsTXC(PC)	$\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8)	NA		NA		
11	TdTXCf(TXD)	$\overline{TxC}$ ↓ to TxD Delay (XI Mode) (Note 2)		130		80	nsec
12	TdTXCr(TXD)	$\overline{TxC}$ ↑ to TxD Delay (XI Mode) (Notes 2, 5)		130		80	nsec
13	TdTXD(TRX)	TxD to $\overline{TRxC}$ Delay (Send Clock Echo)		120		80	nsec
14a	TwRTXh	$\overline{RTxC}$ HIGH Width (Note 6)	100		80		nsec
14b	TwRTXh(E)	$\overline{RTxC}$ HIGH Width (Note 9)	34		15.6		nsec
15a	TwRTXl	$\overline{RTxC}$ LOW Width (Note 6)	100		80		nsec
15b	TwRTXl(E)	$\overline{RTxC}$ LOW Width (Note 9)	34		15.6		nsec
16a	TcRTX	$\overline{RTxC}$ Cycle Time (Notes 6, 7)	320		244		nsec
16b	TcRTx(E)	$\overline{RTxC}$ Cycle Time (Note 9)	80		31.25		nsec
17	TXRTXX	Crystal Oscillator Period (Note 3)	80	1000	62	1000	nsec
18	TwTRxh	$\overline{TRxC}$ HIGH Width (Note 6)	100		80		nsec
19	TwTRXl	$\overline{TRxC}$ LOW Width (Note 6)	100		80		nsec
20	TcTRX	$\overline{TRxC}$ Cycle Time (Notes 6, 7)	320		244		nsec
21	TwEXT	$\overline{DCD}$ or $\overline{CTS}$ Pulse Width	100		70		nsec
22	TwSY	$\overline{SYNC}$ Pulse Width	100		70		nsec

Notes:
1. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{SYNC}$ have 30-pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{RxC}$ and PCLK or $\overline{TxC}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver, DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to $\overline{RxC}$ or $\overline{TxC}$ synchronization requirement eliminated for PCLK divide-by-four operation.

   $\overline{TRxC}$ and $\overline{RTxC}$ rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

   Tx and Rx input clock slow rates should be kept to a maximum of 30 nsec. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is worst case.
9. ENHANCED FEATURE— $\overline{RTxC}$ used as input to internal DPLL only.

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

System Timing (see Figure 20)

No.	Parameter Symbol	Parameter Description	8.192 MHz Min.	8.192 MHz Max.	10 MHz Min.	10 MHz Max.	Unit
1	TdRXC(REQ)	$\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	TcPc
2	TdRXC(W)	$\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	8	14	TcPc
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	TcPc
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	TcPc
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	TcPc
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	TcPc
7a	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	TcPc
7b	TdTXC(EDRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Notes 3, 4)	5	8	5	8	TcPc
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	TcPc
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPc
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPc

No.	Parameter Symbol	Parameter Description	12.5 MHz Min.	12.5 MHz Max.	16.384 MHz Min.	16.384 MHz Max.	Unit
1	TdRXC(REQ)	$\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	TcPc
2	TdRXC(W)	$\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	14	8	14	TcPc
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	TcPc
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	TcPc
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	TcPc
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	11	5	11	TcPc
7a	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	TcPc
7b	TdTXC(EDRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Notes 3, 4)	5	8	5	8	TcPc
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	TcPc
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPc
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPc

Notes:
1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
4. Parameter applies to Enhanced Request mode only.

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

Read and Write Timing (see Figure 21)

No.	Parameter Symbol	Parameter Description	8.192 MHz		10 MHz		Unit
			Min.	Max.	Min.	Max.	
1	TwPCl	PCLK LOW Width	50	1000	40	1000	nsec
2	TwPCh	PCLK HIGH Width	50	1000	40	1000	nsec
3	TfPC	PCLK Fall Time		15		12	nsec
4	TrPC	PCLK Rise Time		15		12	nsec
5	TcPC	PCLK Cycle Time	122	2000	100	2000	nsec
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	70		50		nsec
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		nsec
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	70		50		nsec
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		nsec
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	20		20		nsec
11	TsIAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1)	145		120		nsec
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		nsec
13	TsIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time (Note 1)	145		120		nsec
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		nsec
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	40		30		nsec
16	TsCEl(WR)	$\overline{CE}$ LOW to $\overline{WR}$ ↓ Setup Time	0		0		nsec
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		nsec
18	TsCEh(WR)	$\overline{CE}$ HIGH to $\overline{WR}$ ↓ Setup Time	60		50		nsec
19	TsCEl(RD)	$\overline{CE}$ LOW to $\overline{RD}$ ↓ Setup Time (Note 1)	0		0		nsec
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note1)	0		0		nsec
21	TsCEh(RD)	$\overline{CE}$ HIGH to $\overline{RD}$ ↓ Setup Time (Note 1)	60		50		nsec
22	TwRDl	$\overline{RD}$ LOW Width (Note 1)	150		125		nsec
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		nsec
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		nsec
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		140		125	nsec
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		40		35	nsec

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.

2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC Load and minimum AC load.

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

Read and Write Timing (see Figure 21)

No.	Parameter Symbol	Parameter Description	12.5 MHz		16.384 MHz		Unit
			Min.	Max.	Min.	Max.	
1	TwPCl	PCLK LOW Width	34	1000	26	1000	nsec
2	TwPCh	PCLK HIGH Width	34	1000	26	1000	nsec
3	TfPC	PCLK Fall Time		10		8	nsec
4	TrPC	PCLK Rise Time		10		8	nsec
5	TcPC	PCLK Cycle Time	80	2000	61	2000	nsec
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	45		35		nsec
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		nsec
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	45		35		nsec
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		nsec
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	15		15		nsec
11	TsIAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1)	95		70		nsec
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		nsec
13	TsIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time	95		70		nsec
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		nsec
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	20		15		nsec
16	TsCEl(WR)	$\overline{CE}$ LOW to $\overline{WR}$ ↓ Setup Time	0		0		nsec
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		nsec
18	TsCEh(WR)	$\overline{CE}$ HIGH to $\overline{WR}$ ↓ Setup Time	40		30		nsec
19	TsCEl(RD)	$\overline{CE}$ LOW to $\overline{RD}$ ↓ Setup Time (Note 1)	0		0		nsec
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note1)	0		0		nsec
21	TsCEh(RD)	$\overline{CE}$ HIGH to $\overline{RD}$ ↓ Setup Time	40		30		nsec
22	TwRDl	$\overline{RD}$ LOW Width (Note 1)	90		75		nsec
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		nsec
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		nsec
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		90		70	nsec
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		25		20	nsec

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.

2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC load and minimum AC load.

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 22–24)

No.	Parameter Symbol	Parameter Description	8.192 MHz Min.	8.192 MHz Max.	10 MHz Min.	10 MHz Max.	Unit
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		220		160	nsec
28	TwWRI	$\overline{WR}$ LOW Width	150		125		nsec
29	TdWRf(DW)	$\overline{WR}$ ↓ to Write Data Valid		35		35	nsec
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		nsec
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 2)		170		100	nsec
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 2)		170		100	nsec
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		170		120	nsec
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		170		120	nsec
35a	TdWRr(REQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		4.0TcPc		4.0TcPc	nsec
35b	TdWRr(EREQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay (Note 4)		120		120	nsec
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		NA		NA	nsec
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 2)		500		400	nsec
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 3)	150		125		nsec
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	150		125		nsec
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		140		120	nsec
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	95		80		nsec
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		nsec
43	TdIEI(IEO)	IEI to IEO Delay Time		95		80	nsec
44	TdPC(IEO)	PCLK ↑ to IEO Delay		200		175	nsec
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 2)		450		320	nsec
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	15		15		nsec
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	15		15		nsec
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident LOW for Reset	150		100		nsec
49	Trc	Valid Access Recovery Time (Note 1)	3.5		3.5		nsec

Notes:
1. Parameter applies only between transactions involving the ESCC. If $\overline{WR}/\overline{RD}$ falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.
4. Parameter applies to Enhanced Request mode only.

SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

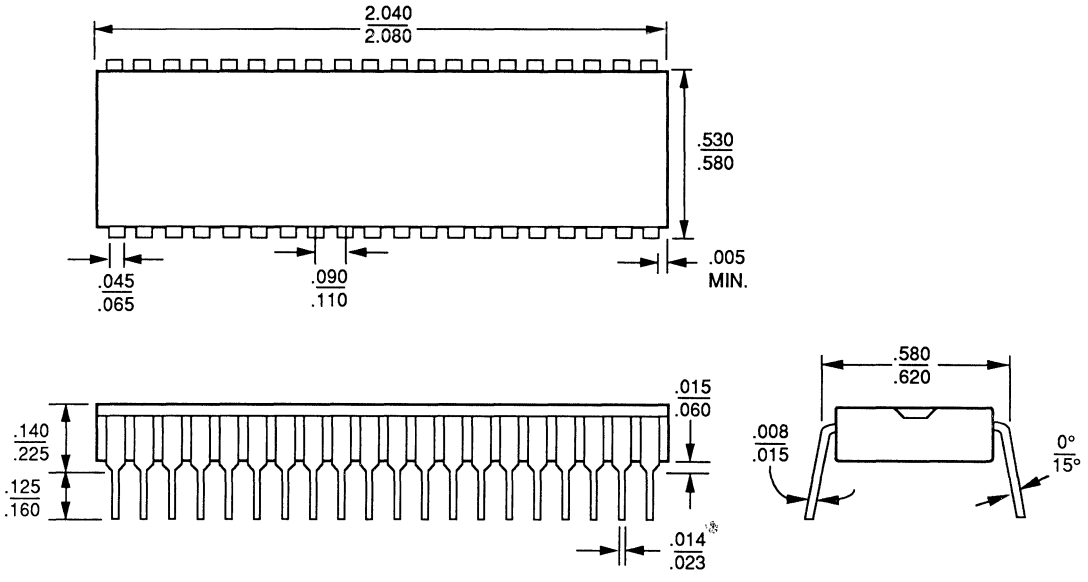
Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 22–24)

No.	Parameter Symbol	Parameter Description	12.5 MHz Min.	12.5 MHz Max.	16.384 MHz Min.	16.384 MHz Max.	Unit
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		120		100	nsec
28	TwWRI	$\overline{WR}$ LOW Width	90		75		nsec
29	TdWRf(DW)	$\overline{WR}$ ↓ to Write Data Valid		25		20	nsec
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		nsec
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 2)		70		50	nsec
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 2)		70		50	nsec
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		100		70	nsec
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		100		70	nsec
35a	TdWRr(REQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		4.0TcPc		4.0TcPc	nsec
35b	TdWRr(EREQ)	$\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay (Note 4)		100		70	nsec
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		NA		NA	nsec
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 2)		350		175	nsec
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 3)	95		50		nsec
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	95		75		nsec
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		90		70	nsec
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	65		50		nsec
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		nsec
43	TdIEI(IEO)	IEI to IEO Delay Time		65		45	nsec
44	TdPC(IEO)	PCLK ↑ to IEO Delay		130		80	nsec
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 2)		260		200	nsec
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	10		10		nsec
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	10		10		nsec
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident LOW for Reset	85		75		nsec
49	Trc	Valid Access Recovery Time (Note 1)	3.5		3.5		TcPc

Notes:
1. Parameter applies only between transactions involving the ESCC. If $\overline{WR/RD}$ falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.
4. Parameter applies to Enhanced Request mode only.

PHYSICAL DIMENSIONS

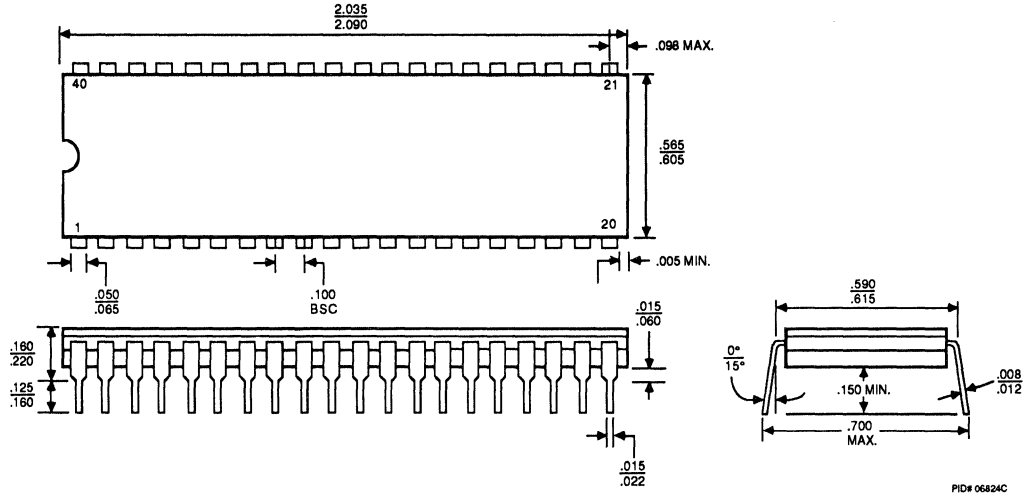
PD 040



PID# 06823B

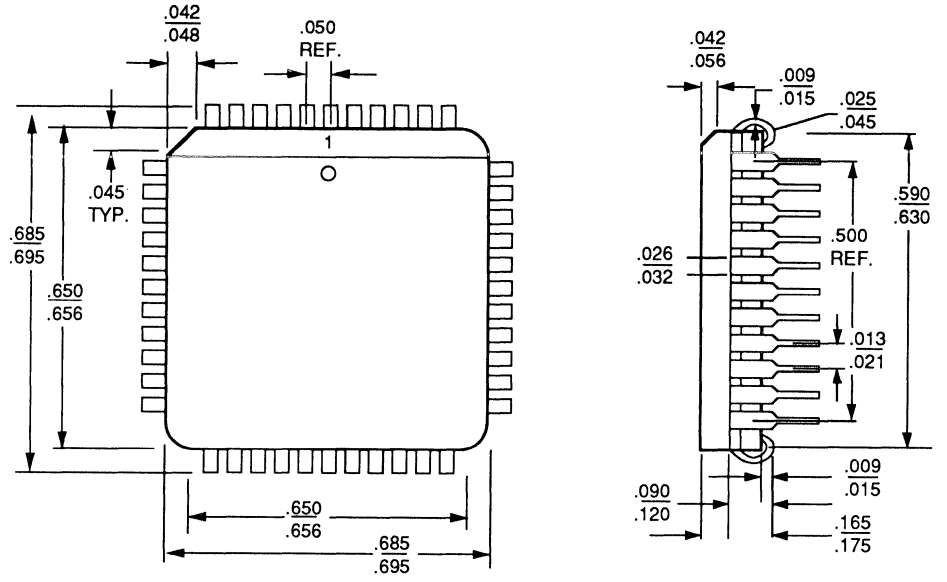
PHYSICAL DIMENSIONS

CD 040



PHYSICAL DIMENSIONS

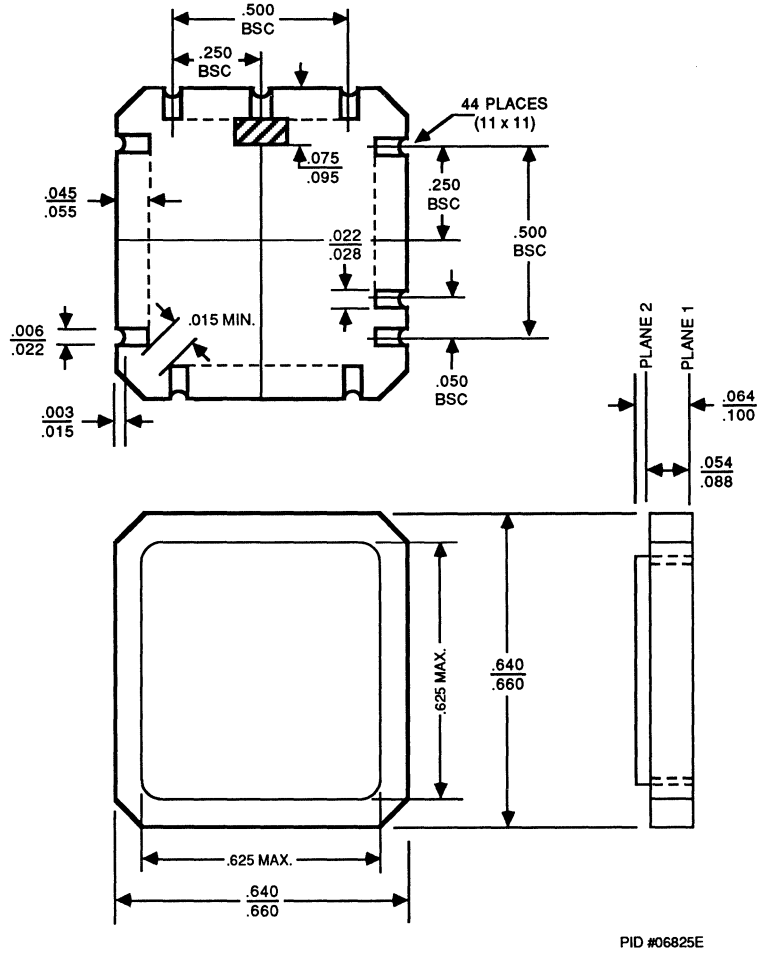
PL 044



06752D

PHYSICAL DIMENSIONS

CL 044



PID #06825E

Advanced
Micro
Devices

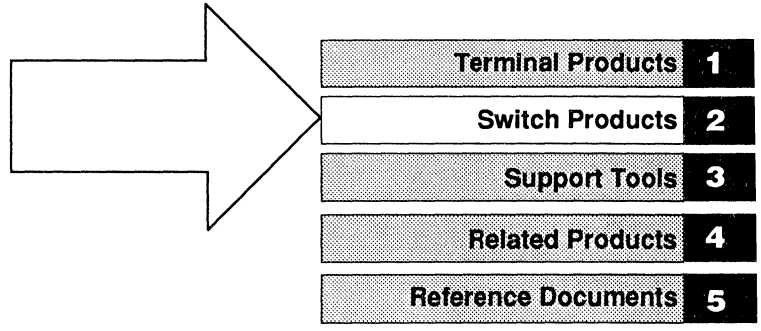


Table of Contents

Chapter 2 **Switch Products**

Preliminary

Am2055

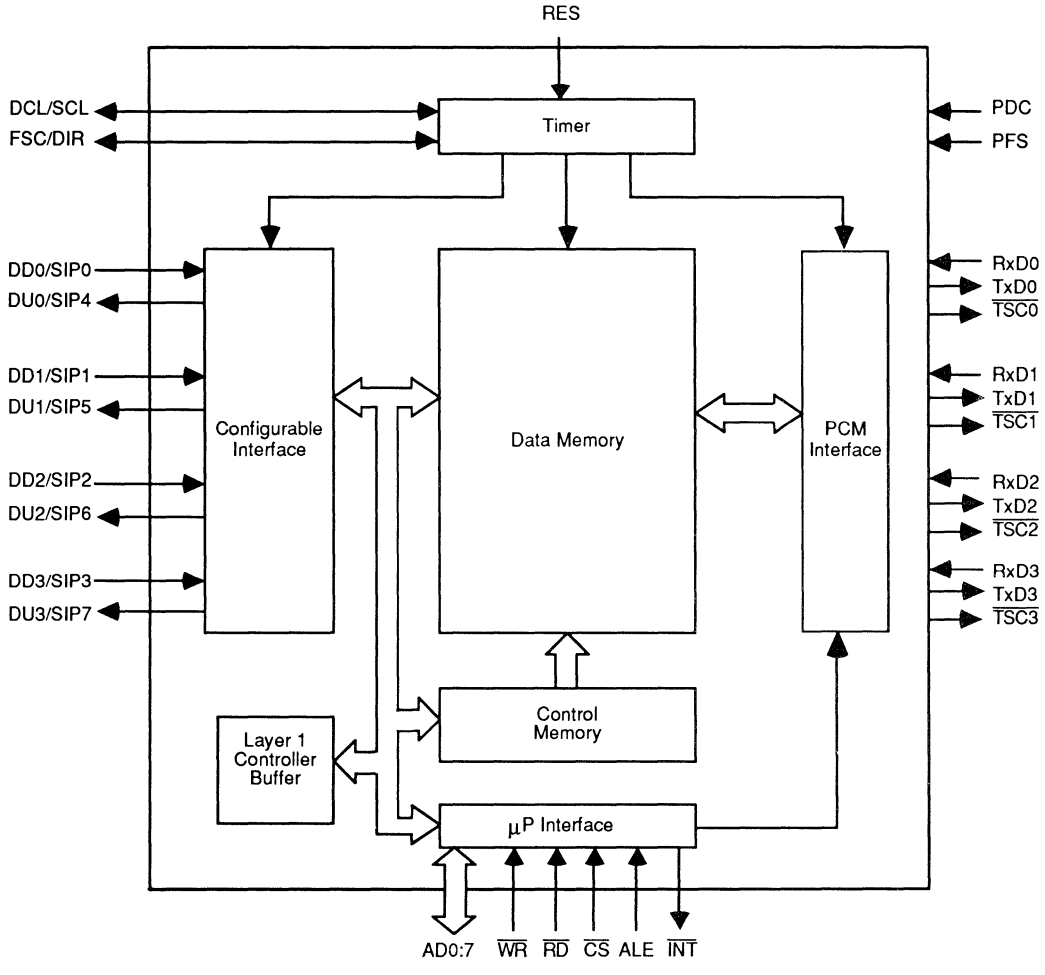
Extended PCM Interface Controller (EPIC)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Board Controller for up to 32 ISDN or 64 voice subscribers
- Nonblocking switch for 128 channels (16-, 32-, or 64-kb/s bandwidth or a mixture thereof)
- Two consecutive 64-kb/s channels can be handled as a single quasi 128-kb/s channel.
- Timeslot assignment freely programmable for all connected subscribers
- Two serial interfaces (PCM and Configurable)
- Programmable for a wide range of data rates (8 to 8192 kb/s)

BLOCK DIAGRAM



Publication # 11149 Rev. C Amendment /0
Issue Date: September 1989

DISTINCTIVE CHARACTERISTICS (continued)

- **Data rates of PCM and configurable interfaces independent from each other (data rate adaptation)**
- **Single and double rate clock selectable**
- **PCM interface**
  - –Tri-state control signals for external drivers
  - –Programmable clock shift
- **Configurable interface**
  - –Configurable for IOM™-compatible devices (4 duplex ports)
  - –Configurable for SLD-compatible devices (8 bidirectional I/O ports)
  - –Configurable for PCM applications
- **Standard microprocessor interface with multiplexed address/data bus or separate address and data buses (PLCC 44)**
- **Handling of Layer 1 functions**
  - –Change detection logic for C/I (IOM configuration) or feature control (SLD configuration) channels
  - –Buffered monitor (IOM configuration) or signaling channel (SLD configuration)
- **Comfortable microprocessor access to two selected channels**
- **40-pin Dual In-Line or 44-pin PLCC package**
- **Advanced low power CMOS technology**

GENERAL DESCRIPTION

The Am2055 Extended PCM Interface Controller (EPIC™) is a monolithic switching device for the path control of up to 128 channels of 16-, 32- or 64-kb/s bandwidth. Two consecutive 64-kb/s channels may also be handled as a quasi-single 128-kb/s channel. For these channels, the EPIC performs non-blocking space time switching between two serial interfaces, the system and the configurable interface.

Both interfaces can be programmed to operate at different data rates between 8 and 8192 kb/s. The system interface consists of up to four duplex ports with a tristate indication signal for each output line. The configurable interface can be selected to incorporate either four duplex or eight bidirectional I/O ports (SLD).

The EPIC can therefore be programmed to communicate either with SLD or with IOM (ISDN Oriented Modular) compatible devices. In both cases, the device handles the Layer 1 functions buffering the C/I and monitor channels for IOM-compatible devices and the feature control and signaling channels for SLD-compatible devices.

Due to its capability to switch channels of different bandwidths, the EPIC can handle up to 32 ISDN subscribers with their 2B + D channel structure in IOM configuration or up to 16 subscribers in SLD configuration. Since its interfaces can operate at different data rates, the EPIC is an ideal device for data rate adaptation.

Moreover, the EPIC is one of the fundamental building blocks for networks with either central, decentralized, or mixed signaling and packet data handling architectures. The other key devices are the IDEC™ (ISDN D-Channel Exchange Controller, Am2075) and the HSCX (Enhanced High Level Serial Communication Controller, Am82525).

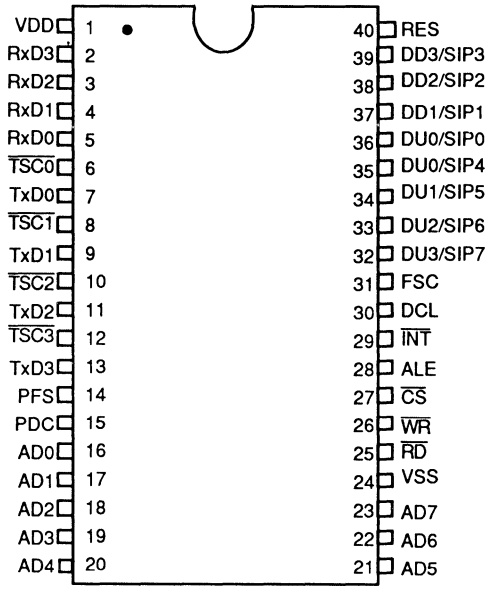
Applications of the EPIC include communication multiplexers, concentrators, and central switches, as well as peripheral ISDN and analog line cards.

The EPIC is available in a 40-pin DIP or a 44-pin PLCC package.

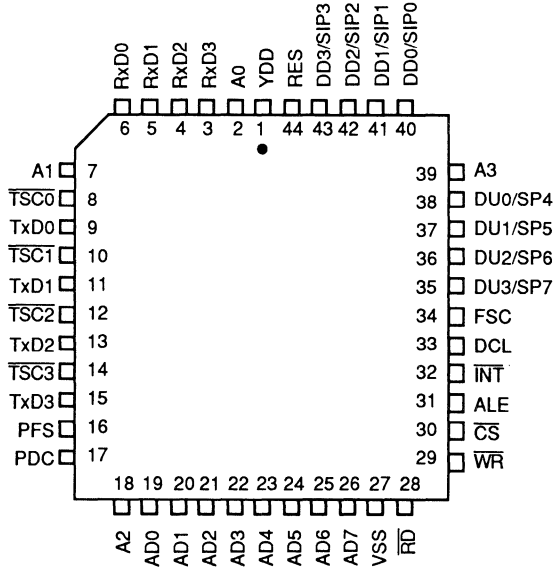
The 40-pin DIP version is controlled by a standard 8-bit parallel microprocessor interface with a multiplexed address-data bus. In the PLCC package, the device may optionally be controlled by separate address and data buses.

CONNECTION DIAGRAMS
Top View

40-Pin DIP

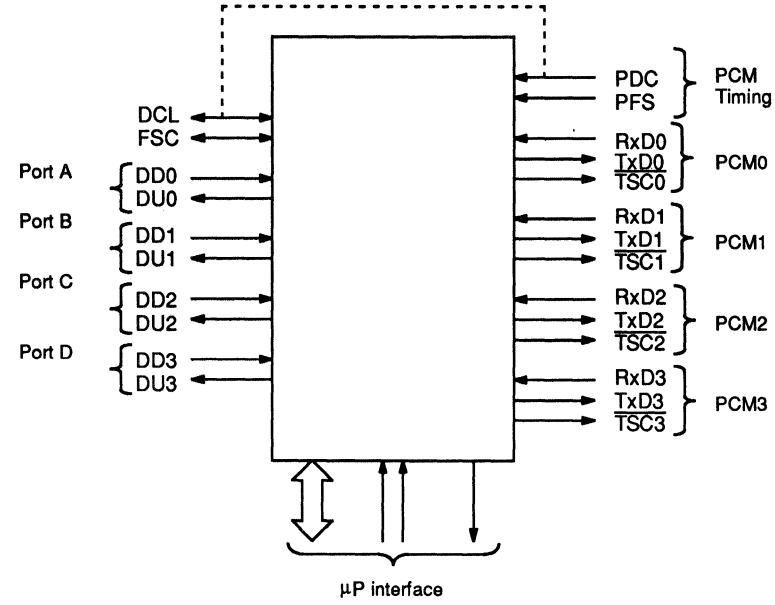


44-Pin PLCC

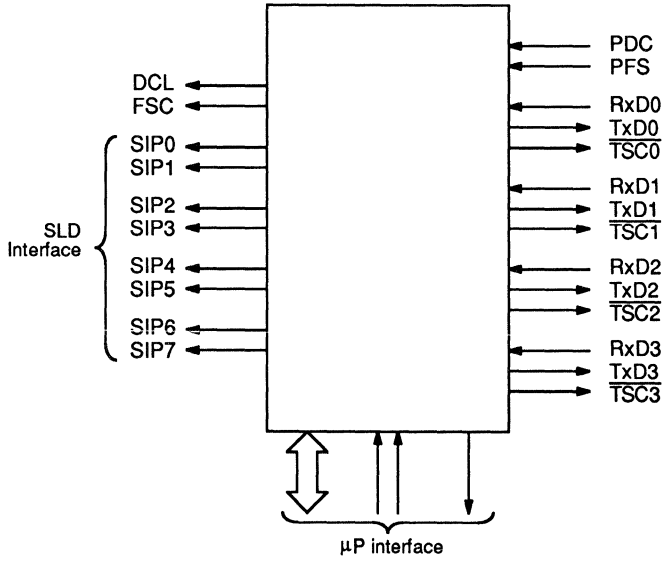


Note: Pin 1 is marked for orientation.

LOGIC SYMBOLS



Functional Symbol for the Duplex Configuration

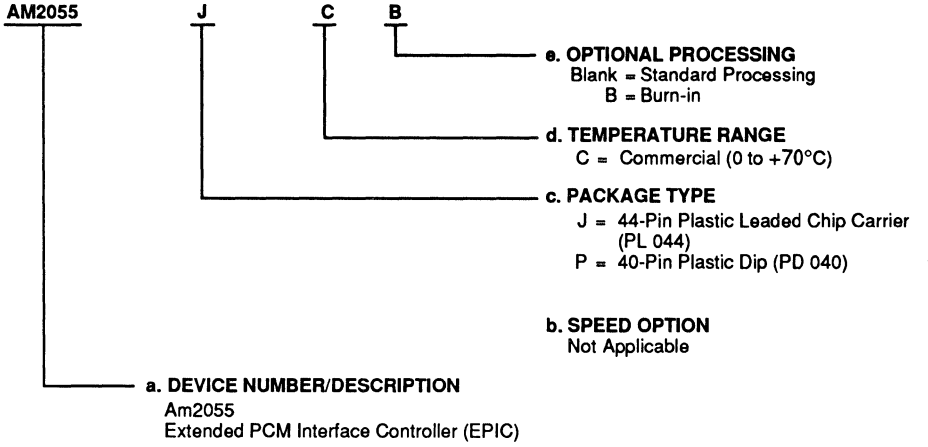


Functional Symbol for the SLD Configuration

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM2055	PC, JC, PCB, JCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

2

PIN DESCRIPTION

A0–3
Address Bus Bits (Input)

This input interfaces with one bit of the system's address bus to select internal registers for a read or write access. This pin is only provided in the PLCC package and only active if a demultiplexed microprocessor interface mode is selected.

AD0–7
Address Data Bus (Input/Output)

If the multiplexed address/data microprocessor interface bus mode is selected, these pins transfer data and commands between the microprocessor and the EPIC. If a demultiplexed mode is used, these bits interface with the system data bus.

ALE
Address Latch Enable (Input)

A logical high on this line indicates an address of an EPIC internal register on the external address/data bus.

$\overline{CS}$
Chip Select (Input)

A low on this line selects the EPIC for a read/write operation.

DCL
Data Clock (Input/Output)

Input or Output in IOM/Slave clock in SLD configuration.

DD0–3/SIP0–3
Data Downstream Inputs (Input/Output)

Inputs in IOM configuration. Subscriber interface ports 0, 1, 2, and 3 in SLD configuration.

DU0–3/SIP4–7
Data Upstream Outputs (Input/Output)

Outputs in IOM configuration. Subscriber interface port 4, 5, 6, and 7 in SLD configurations.

FSC
Frame Synchronization (Input/Output)

Input or Output in IOM configuration/Direction indication signal in SLD configuration.

$\overline{INT}$
Interrupt Line (Open Drain)

Active low.

PDC
PCM Interface Data Clock (Input)

Single or double rate.

PFS
PCM Interface Frame Synchronization Pulse (Input)

$\overline{RD}$
Read (Input)

The signal indicates a read operation, active low.

RES
Reset (Input)

A logical high on this input forces the EPIC into the reset state.

RxD0–3
Receive PCM Interface Data (Input)

Serial data is received at these lines at standard TTL or CMOS levels.

$\overline{TSC0}$–3
Tristate Control for the PCM Interface (Output)

These lines are low when the corresponding TxD outputs are valid.

TxD0–3
Transmit PCM Interface Data (Output)

Serial data are sent by these lines at standard TTL or CMOS levels. These pins can be tristated.

V_{DD}
Supply Voltage (Input)

5 V ± 5%

VSS
Ground (Input)

0 volts.

$\overline{WR}$
Write (Input)

This signal indicates a write operation, active low.

FUNCTIONAL DESCRIPTION

The Am2055 Extended PCM Interface Controller (EPIC) is a peripheral board controller. It combines the non-blocking switching function between the PCM and the configurable interfaces for 128 channels per direction with the Layer 1 control function for connection setup/termination/maintenance on one chip.

In the downstream direction, the input information of a complete frame is stored in the data memory. The incoming channels are written in sequence into fixed positions in the data memory. This is controlled by the downstream input counter with an 8-kHz repetition rate. A cyclic write sequence results.

For the downstream switching, the control memory (CM) is read in sequence. The addressed location contains a pointer to a location in the data memory. The byte in this data memory location is read into the current configurable interface timeslot, resulting in a random read sequence.

The read access of the control memory is controlled by the downstream output counter, correlating the data memory read operations with the downstream output timeslot sequence.

In the upstream direction, the data is written to the data memory randomly, under CM control, and read from there cyclically.

Hence, for the desired connection, the control memory needs to be programmed beforehand using the MAAR, MADR, and MACR registers. The control memory address corresponds to one particular configurable interface timeslot and line number. The contents of this control memory address point to a particular PCM interface timeslot and line number now resident in the data memory.

For upstream output, four control bits per timeslot are provided in the data memory. These control the output driver state of any possible sub-time slot.

In addition to the data memory address, each CM address also points to four code bits determining the bandwidth of the switched channel. These code bits are also used to mark the signaling channels at the CFI.

The C device can be used in two different setups:

- In the SLD setup, every channel at the configurable interface can be programmed to be either input or output. Eight equivalent bidirectional ports at the configurable interface result.
- In the duplex setup, four of the eight lines of the configurable interface are predetermined as outputs, four as inputs. Four duplex ports result.

In both of these setups, the EPIC provides a switching capability for up to 128 channels and direction.

The IOM and the SLD configurations previously mentioned are special cases of these setups.

In the GCI configuration, the EPIC switches the B and D channels of up to 32 subscribers working in the duplex setup. Additionally, the device handles the monitor and C/I channel buffering to the microprocessor.

In the SLD configuration, the EPIC switches up to 64-kb/s channels operating in bidirectional setup. Additionally, the device handles the feature control and signaling channels buffering to the microprocessor.

In the IOM configuration, upon proper programming, the EPIC checks the incoming C/I channels and generates interrupts if changes occur. In the case of the bidirectional configuration, it implements the double last look algorithm with a period adaptable to a wide range of system needs.

For handling the monitor or feature control channel, the EPIC is equipped with a FIFO buffering up to 16 bytes of information. The contents can be transferred or received upon a special command, or, they can be dealt with largely autonomously according to the IOM handshake procedure.

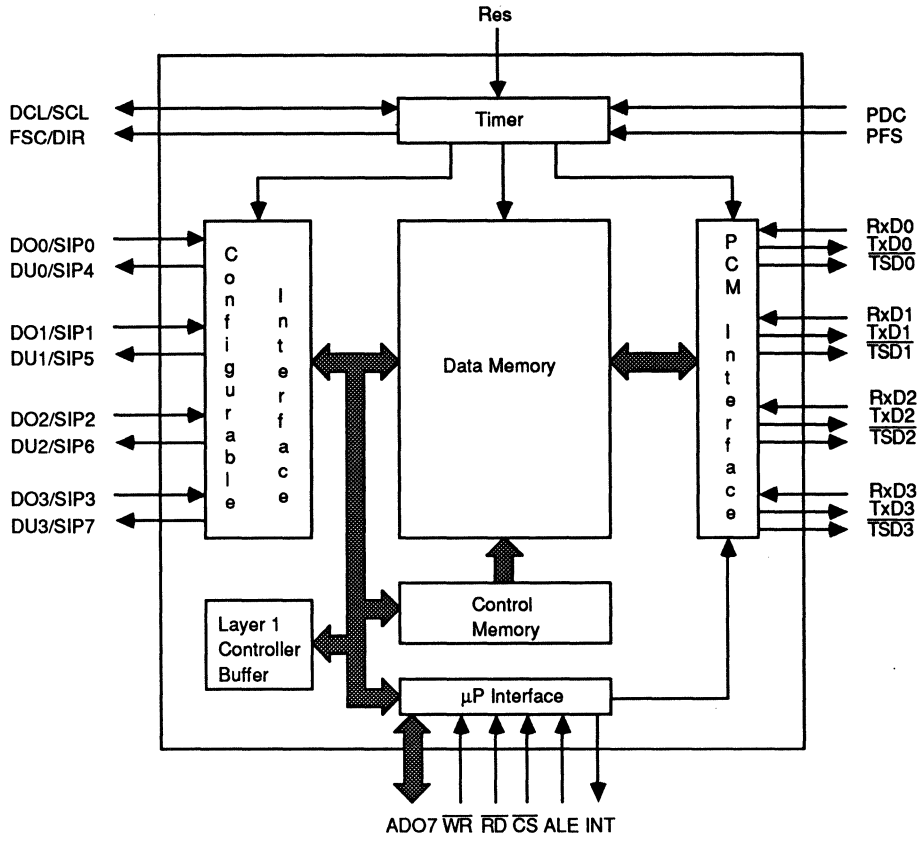


Figure 1. Functional Block Diagram of the EPIC

Table 1. Detailed Register Description

Group	Register Name	Access Write (W) Read (R)	µP Interface Mode MUX AD7–AD0	µP Interface Mode deMUX A3–A0/RBS	Reset Value	Register Content
PCM	PMOD	R/W	20_H	0_H/1	00	PCM Mode Register
	PBNR	R/W	22_H	1_H/1	FF	PCM Bit Number Register
	POFD	R/W	24_H	2_H/1	00	PCM Offset Downstream Register
	POFU	R/W	26_H	3_H/1	00	PCM Offset Upstream Register
	PCSR	R/W	28_H	4_H/1	00	PCM Clock Shift Register
	PICM	R/W	$2A_H$	5_H/1	—	PCM Input Comparison Mismatch Register
CFI	CMD1	R/W	$2C_H$	6_H/1	00	CFI Mode Register 1
	CMD2	R/W	$2E_H$	7_H/1	00	CFI Mode Register 2
	CBNR	R/W	30_H	8_H/1	FF	CFI Bit Number Register
	CTAR	R/W	32_H	9_H/1	00	CFI Time Slot Adjustment Register
	CBSR	R/W	34_H	A_H/1	00	CFI Bit Shift Register
	CSCR	R/W	36_H	B_H/1	00	CFI Subchannel Register

Table 1. (continued)

Group	Register Name	Access Write (W) Read (R)	µP Interface Mode MUX AD7–AD0	µP Interface Mode deMUX A3–A0/RBS	Reset Value	Register Content
MAR	MACR	R/W	00_H	$0_H/0$	—	Memory Access Control Register
	MAAR	R/W	02_H	$1_H/0$	—	Memory Access Address Register
	MADR	R/W	04_H	$2_H/0$	—	Memory Access Data Register
STR	STDA	R/W	06_H	$3_H/0$	—	Synchron Transfer Data Register A
	STDB	R/W	08_H	$4_H/0$	—	Synchron Transfer Data Register B
	SARA	R/W	$0A_H$	$5_H/0$	—	Synchron Transfer Receive Address Register A
	SARB	R/W	$0C_H$	$6_H/0$	—	Synchron Transfer Receive Address Register B
	SAXA	R/W	$0E_H$	$7_H/0$	—	Synchron Transfer Transmit Address Register A
	SAXB	R/W	10_H	$8_H/0$	—	Synchron Transfer Transmit Address Register B
	STCR	R/W	12_H	$9_H/0$	00	Synchron Transfer Control Register
MFCH	MFAIR	R	14_H	$A_H/0$	Undef.	MF Channel Active Indication Register
	MFSAR	W	14_H	$A_H/0$	Undef.	MF Channel Subscriber Address Register
	MFFIFO+	R/W	16_H	$B_H/0$	Empty	MF Channel FIFO
SCR	C/I FIFO	R	18_H	$C_H/0$	Validity 0	Signaling Channel FIFO
	TIMR	W	18_H	$C_H/0$	00	Timer Register
	STAR	R	$1A_H$	$D_H/0$	05	Status Register
	CMDR	W	$1A_H$	$D_H/0$	00	Command Register
	ISTA	R	$1C_H$	$E_H/0$	00	Interrupt Status Register
	MASK	W	$1C_H$	$E_H/0$	00	Mask Register
	OMDR	R/W	$1E_H/3E_H$	F_H/X	00	Operation Mode Register
	VNSR	R	$3A_H$	$D_H/1$		Version Number Register

Registers for the Control of the PCM Interface

PCM Mode Register (PMOD)

Access in the multiplexed microprocessor Interface mode: Read or write, address: 20_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 0_H, OMDR:RBS = 1

Reset value: 00_H

bit 7 **bit 0**

PMD1	PMD0	PCR	PSM	AIS1	AIS0	AIC1	AIC0

PMD1–PMD0 PCM Mode 1 and 0

These bits define the PCM mode according to Table 2. The ports used in the selected PCM mode are stated in Table 3.

Table 2. Modes at the PCM Interface

PMD1	PMD2	PCM Mode	Port Count	Max. Data Rate (kb/s)	Data Rate Stepping (kb/s)
0	0	0	4	2048	8
0	1	1	2	4096	16
1	0	2	1	8192	32

PCR PCM Clock Rate

A logical "0" selects the PCM interface for single clock rate operation, a logical "1" for double clock rate operation. In single clock rate operation, the PCM interface of the EPIC is supplied with clock and data of the same frequency; in double clock rate operation the clock frequency is twice the data frequency. In PCM Mode 2, only single clock rate operation is feasible.

PSM PCM Synchronization Mode

The rising edge of the PFS signal synchronizes the PCM frame. The PFS signal is evaluated with the rising clock slope (logical "1") or with the falling clock slope (logical "0"). Also refer to Figure 2.

AIS1–AIS0 Alternative Input Selection 1 and 0

These bits determine the relationship between the physical pins and the logical port numbers used to program the switching function of the EPIC, as shown in Table 3.

Setting AIS0 or AIS1 in PCM Mode 0 may result in an undefined operational behavior and should therefore be avoided.

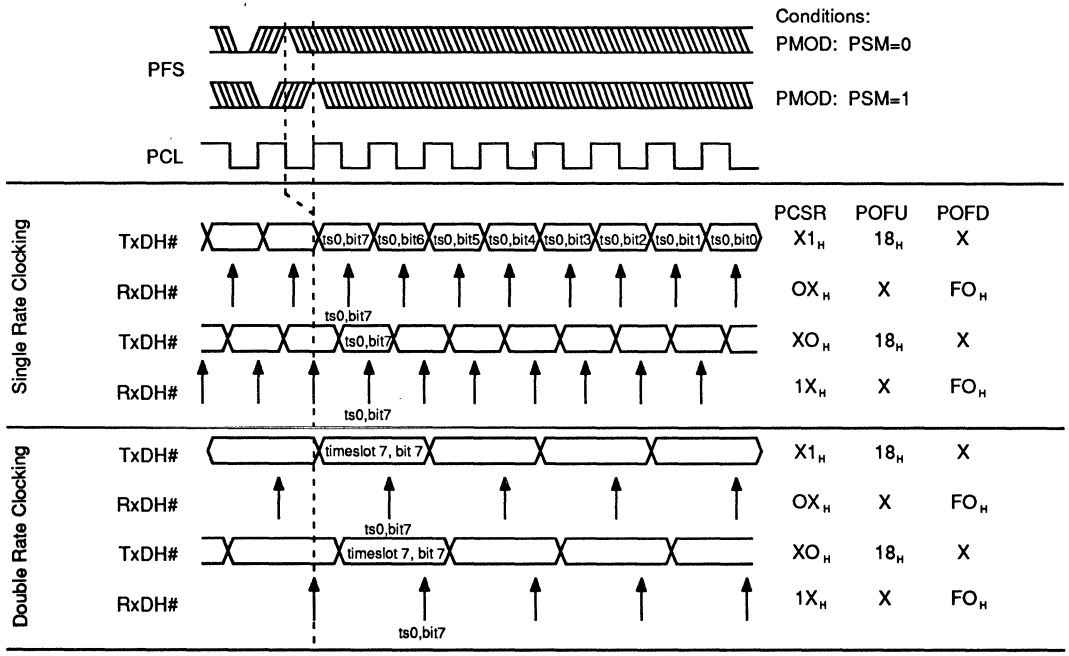


Figure 2. Upstream and Downstream Timing at the PCM Interface for Single and Double Rate Clocking.

Table 3. PCM Pin Configuration

	Port 0			Port 1			Port 2			Port 3		
Mode	RxD0	TxD0	$\overline{TSC0}$	RxD1	TxD1	$\overline{TSC1}$	RxD2	TxD2	$\overline{TSC2}$	RxD3	TxD3	$\overline{TSC3}$
0	IN0	Out 0	Val 0	IN1	Out 1	Val 1	IN2	Out 2	Val 2	IN3	Out 3	Val 3
1	IN0 (AIS0 = 1)	Out 0	Val 0	IN0 (AIS0 = 0)	tristate	AIS0	IN1 (AIS1 = 1)	Out 1	Val 1	IN1 (AIS1 = 0)	tristate	AIS1
2	tristate	Out	Val	tristate	tristate	AIS0	IN (AIS1 = 1)	undef.	undef.	IN (AIS1 = 0)	tristate	AIS1

Note: The TSC pins output either information about the validity of the relevant TxD output (active low) or the content of the respective AIS bit.

AIC1–AIC0 Alternative Input Comparison 1 and 0

These bits control the input comparison function of the EPIC. If AIC1 is programmed to logical "1," the inputs of Ports 2 and 3 are compared in PCM Modes 1 and 2. If AIC0 is programmed to logical "0," the inputs of Port 0 and 1 are compared in PCM Mode 1. A logical "0" disables the respective comparison function.

Bit Number per PCM Frame (PBNR)

Access in the multiplexed microprocessor interface mode: Read or write, address: 22_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 1_H, OMDR:RBS = 1

Reset value: FF_H

bit 7							bit 0
BNF7	BNF6	BNF5	BNF4	BNF3	BNF2	BNF1	BNF0

BNF7–BNF0 Bit Number per PCM Frame

These bits denote the number of bits constituting a PCM frame. BNF0 is the least significant bit, BNF7 the most significant bit. In PCM Mode 0, the EPIC expects the number of bits as programmed to BNF7–BNF0; in PCM Mode 1, it expects twice the number, and in PCM Mode 2, it expects four times this number to be contained in one frame.

PCM Offset Downstream Register (POFD)

Access in the multiplexed microprocessor interface mode: Read or write, address: 24_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 2_H, OMDR:RBS = 1

Reset value: 00_H

bit 7							bit 0
OFD9	OFD8	OFD7	OFD6	OFD5	OFD4	OFD3	OFD2

OFD9–OFD2 Offset Downstream bits 9–2

These bits together with PCSR:OFD1–0 determine the offset of the PCM downstream frame. The positive edge of PFS marks the bit number in the downstream frame (BND) according to the following formulas: (BND = 1: first bit in the frame).

PCM Mode 0: $BND = 17_D + (OFD9–OFD2)_B$

PCM Mode 1: $BND = 33_D + (OFD9–OFD1)_B$

PCM Mode 2: $BND = 65_D + (OFD9–OFD0)_B$

This behavior is also shown in Figure 3.

The stated formulas are valid for a bit number count of 256, 512, or 1024 in PCM Modes 0,1, or 2, respectively. For a lower bit number count the decimal commands have to be increased by the number of bits missing. For example, for a frame consisting of 24 time slots (PCM Mode 0), OFD has to be programmed 10110000. Then the positive PFS edge marks bit 7 of time slot 0.

PCM Offset Upstream Register (POFU)

Access in the multiplexed microprocessor interface mode: Read or write, address: 26_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 3_H, OMDR:RBS = 1

Reset value: 00_H

bit 7							bit 0
OFU9	OFU8	OFU7	OFU6	OFU5	OFU4	OFU3	OFU2

OFU7–OFU0 Offset upstream bits 9–2

These bits together with PCSR:OFU1–OFU0 determine the offset of the PCM upstream frame. The positive PFS edge marks the bit number in the upstream frame (BNU) according to the following formulas (BNU = 1: first bit in frame).

PCM Mode 0: $BNU = (OFU9–OFU2)_B - 23_D$

PCM Mode 1: $BNU = (OFU9–OFU1)_B - 47_D$

PCM Mode 2: $BNU = (OFU9–OFU0)_B - 95_D$

This behavior is also shown in Figure 3.

PCM Clock Shift Register (PCSR)

Access in the multiplexed microprocessor interface mode: Read or write, address: 28_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 4_H, OMDR:RBS = 1

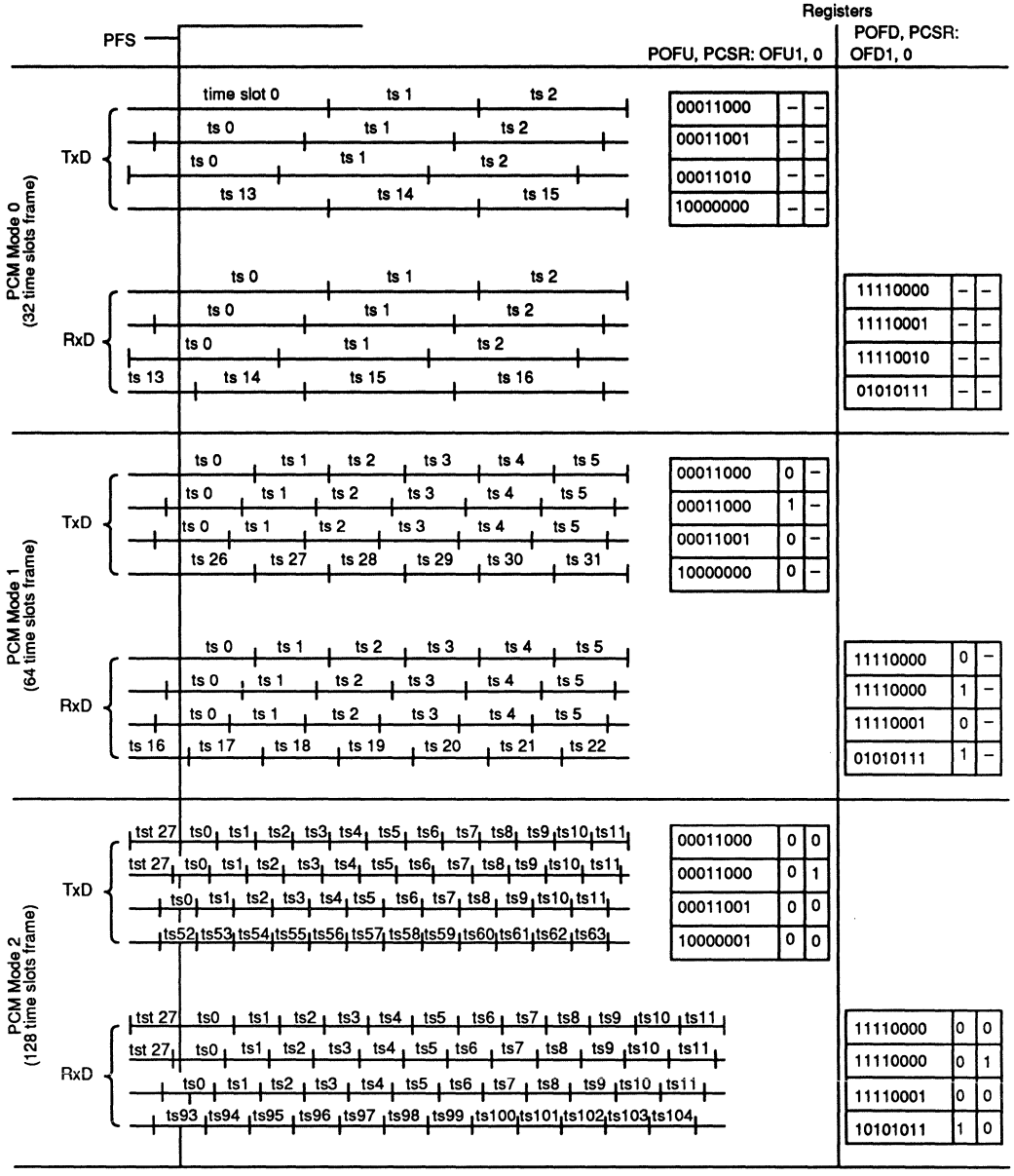


Figure 3. PCM Interface Framing Offset

Reset value: 00_H

bit 7 **bit 0**

0	OFD1	OFD0	DRE	0	OFU1	OFU0	URE

OFD1–OFD0 Offset downstream bits 1–0

See POFD register. In PCM Mode 0, both bits, and in PCM Mode 1, OFD0 needs to be fixed to logical "0."

DRE Downstream rising edge

The PCM data is sampled with the rising edge (DRE = 1) or the falling edge (DRE = 0) of DCL, as shown in Figure 2.

OFU1–OFU0 Offset upstream bits 1–0

See POFU register. In PCM Mode 0, both bits, and in PCM Mode 1, OFU0 needs to be fixed to logical "0."

URE Upstream rising edge

The PCM data is transmitted with the rising edge (URE = 1) or the falling edge (URE = 0) of DCL, as shown in Figure 2.

PCM Input Comparison Mismatch (PICM)

Access in the multiplexed microprocessor interface mode: Read, address: $2A_H$

Access in a demultiplexed microprocessor interface mode: Read or write, address 5_H, OMDR:RBS = 1

bit 7 **bit 0**

IPN	TSN6	TSN5	TSN4	TSN3	TSN2	TSN1	TSN0

IPN Input Pair Number

This bit denotes the pair of ports where a bit mismatch of the inputs occurred. A logical "0" indicates a mismatch between Ports 0 and 1, a logical "1" between Ports 2 and 3.

TSN6–TSN0

These bits contain information as to when mismatches occur (see Table 4).

Table 4. Identification of the Differing PCM Data

PCM Mode	Time Slot Identification	Bit Identification
2	TSN–TSN0	
1	TSN6–TSN1	TSN0 = 1: bits 0–3 TSN0 = 0: bits 4–7
0	TSN6–TSN2	TSN1–TSN0 = 11: bits 0–1 TSN1–TSN0 = 10: bits 2–3 TSN1–TSN0 = 01: bits 4–5 TSN1–TSN0 = 00: bits 6–7

Registers for the Control of the Configurable Interface (CFI)

Configurable Interface Mode Register 1 (CMD1)

Access in the multiplexed microprocessor interface mode: Read or write address: $2C_H$

Access in a demultiplexed microprocessor interface mode: Read or write, address 6_H, OMDR:RBS = 1

Reset value: 00_H

bit 7 **bit 0**

CSS	CSM	CSP1	CSP0	CMD1	CMD0	CIS1	CIS0

CSS Clock Source Selection

A logical "0" selects PCL as clock and PFS as framing source for the configurable interface. Clock and framing signals derived from these sources are output at the DCL and FSC pins (OMDR : CSB = 1). A logical "1" selects DCL as clock and FSC as framing signal source for the configurable interface.

CSM Configurable Interface Synchronization Mode

The positive FSC transition synchronizes the CFI frame. The FSC signal is evaluated at every positive clock slope (logical "1") or at every negative clock slope (logical "0"). CSM is only effective for CSS = 1. Also refer to Figure 4.

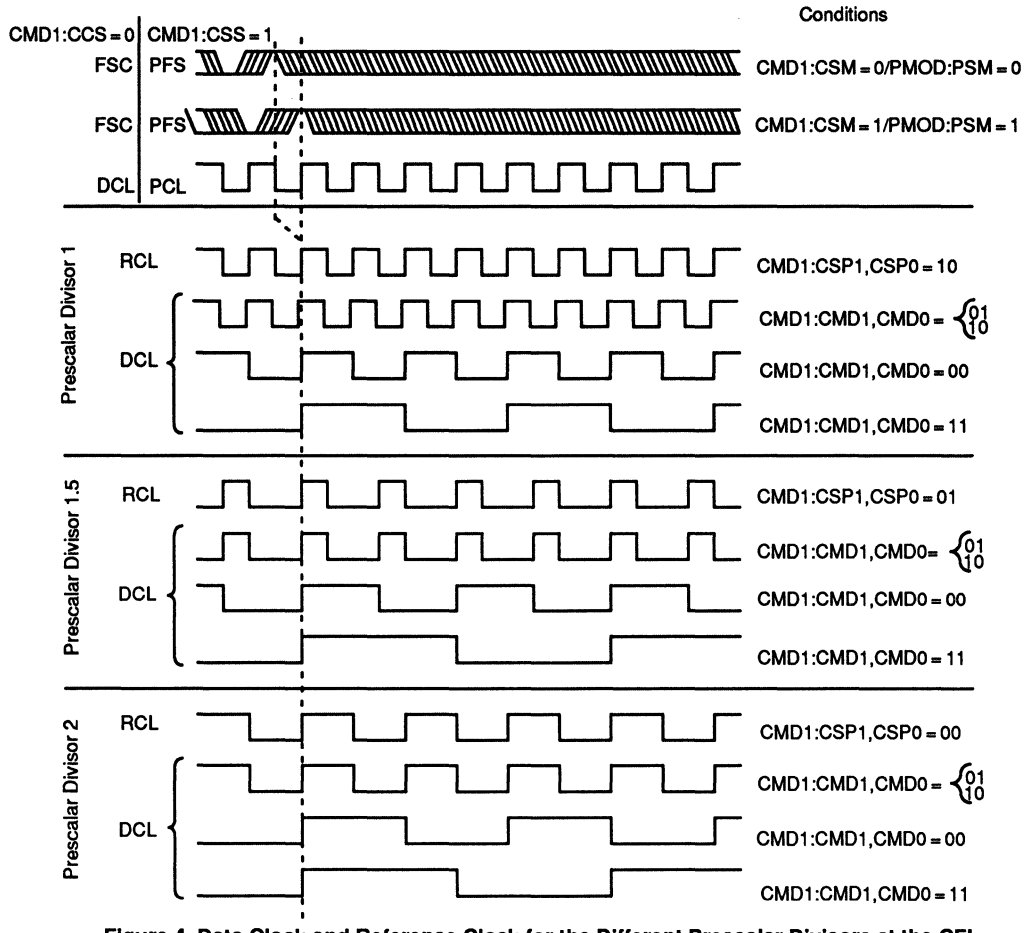


Figure 4. Data Clock and Reference Clock for the Different Prescalar Divisors at the CFI

CSP1–CSP0 Clock Source Prescalar 1,0

The clock source frequency is divided by the values listed in Table 5 to obtain the device reference clock.

Table 5. Prescalar Divisors

CSP1	CSP0	Prescalar Divisor
0	0	2
0	1	1.5
1	0	1
1	1	Not Allowed

CMD1,0 CFI Mode 1,0

These bits define the CFI mode according to Table 6.

C1S1,0 CFI Alternative Input Selection

These bits determine the relationship between the physical pins and the logical port numbers used to program the switching function of the EPIC, as shown in Table 7.

Configurable Interface Mode Register 2 (CMD2)

Access in the multiplexed microprocessor interface mode: Read or write, address: $2E_H$

Access in a demultiplexed microprocessor interface mode: Read or write, address 7_H, OMDR:RBS = 1

Reset value: 00_H

bit 7 **bit 0**

FC2	FC1	FC0	COC	CXF	CRR	CBN9	CBN8

Table 6. Configurable Interface Modes

CMD1	CMD0	CFI Mode	Port Count/ device setup	Max. Data Rate (DR) kb/s	Necessary Reference Clock (RCL)	Possible FSC Output Frequencies
1	1	3	8 bidir	1024	4xDR	DR, 2xDR
0	0	0	4 duplex	2048	2xDR	DR, 2xDR
0	1	1	2 duplex	4096	DR	DR
1	0	2	1 duplex	8192	DR	DR

Table 7. CFI Pin Configuration

CFI Mode	DO0	DU0	DI1	DU1	DD2	DU2	DD3	DU3
0	Out0	In0	Out1	In1	Out	In2	Out3	In3
1	Out0	In0 (CIS0 = 0)	Out1	In1 (CIS1 = 0)	tristate	In0 (CIS0 = 1)	tristate	In1 (CIS1 = 1)
2	Out	InCIS0 = 0)	tristate	not active	tristate	In (CIS0 = 1)	tristate	not active
3	I/O0	I/O4	I/O1	I/O5	I/O2	I/O6	I/O3	I/O7

FC2–FC0 Framing Output Control

For CMD1:CSS = 0 and OMDR:CSB = 0, these bits determine the type of generated CFI framing signal according to Figure 5.

The FSC signal in FC Mode 7 for software-timed multiplexed applications is derived from the FC Modes 3 and 6. Setting CMDR:ST1 starts a cyclic multiplexing process. Its period is defined by the content of TIMR. After each of these periods, the EPIC issues one FSC pulse of FC Mode 3; in all other frames the framing pulse of FC Mode 6 is issued. Figure 6 shows this behavior, assuming TIMR:TVAL6–TVAL0 = 0000100 and CFI Mode 0.

Table 8. Applications of the Framing Control Modes

FC2	FC1	FC0	FC Mode	Main Applications
0	0	0	0	IOM Rev. 1 MUX mode
0	0	1	1	IOM Rev. 1 MUX mode
0	1	0	2	IOM Rev. 2
0	1	1	3	IOM Rev. 2
1	0	0	4	2 Am2085s per SLD Port
1	0	1	5	Reserved
1	1	0	6	IOM Rev. 2 or bidirectional applications
1	1	1	7	Software-timed multiplexed applications

COC Clock Output Control for CMD1 : CSS = 0

A logical "1" selects the EPIC to output a DCL clock signal with a frequency of twice the CFI data rate (double rate clock, CFI Modes 0 and 3 only); a logical "0" produces a single rate clock. Also see Figure 4.

CXF CFI Transmit on Falling Edge

CFI data is transmitted with the rising edge (CXF = 0) or falling edge (CXF = 1) of the reference clock, as shown in Figure 4.

CRR CFI Receive on Rising Edge

CFI data is sampled with the falling (CRR = 0) or rising edge (CRR = 1) of the reference clock (see Figure 7). In CFI Mode 3, CRR has to be set to logical "0."

CBN9–CBN8 CFI Bit Number 9–8

These bits together with CBNR:CBN7–0 hold the number of bits per CFI frame (see CBNR).

Configurable Interface Bit Number Register (CBNR)

Access in the multiplexed microprocessor interface mode: Read or write, address: 30_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 8_H, OMDR : RBS = 1

Reset value: FF_H

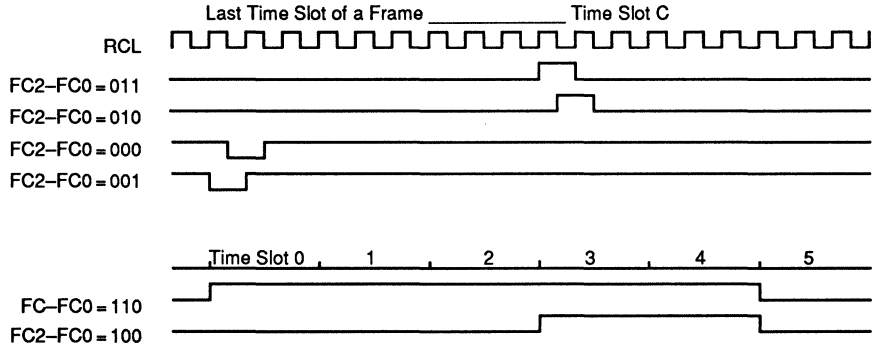


Figure 5. FSC Framing Signal Generation

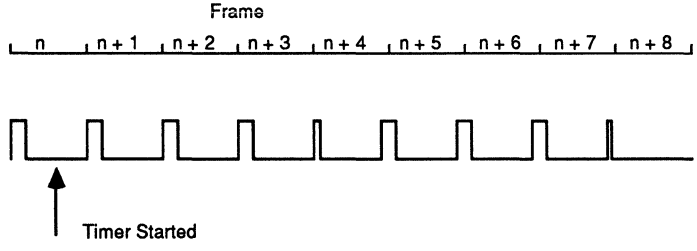


Figure 6. FSC Output Signal in FSC Mode 7

bit 7							bit 0
CBN7	CBN6	CBN5	CBN4	CBN3	CBN2	CBN1	CBN0

CBN7–CBN0 CFI Bit Number 7–0

Together with CMD2:CBN9,CBN8, these bits hold the bit count of a CFI frame minus 1 bit. CBN9 is the most, CBN0 the least significant bit. For example, setting CBN9–CBN0 to 3FFH, the CFI frame consists of 1024 bits.

Configurable Interface Time Slot Adjustment Register (CTAR)

Access in the multiplexed microprocessor interface mode: Read or write, address: 32_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 9_H, OMDR:RBS = 1

Reset value: 00_H

bit 7							bit 0
0	TSN6	TSN5	TSN4	TSN3	TSN2	TSN1	TSN0

TSN6–TSN0 Time Slot Number

The framing signal at the configurable interface marks the downstream time slot numbered according to the formula below:

Time Slot Number Downstream = $(TSN6–TSN0–10)_B$

Also refer to Figure 7.

Configurable Interface Bit Shift Register (CBSR)

Access in the multiplexed microprocessor interface mode: Read or write, address: 34_H

Access in a demultiplexed microprocessor interface mode: Read or write, address A_H, OMDR : RBS = 1

Reset value: 00_H

bit 7							bit 0
0	CDS2	CDS1	CDS0	CUS3	CUS2	CUS1	CUS0

CDS2–CDS0 CFI Downstream Bit Shift 2–0

The framing signal at the configurable interface marks the bit numbered according to the following formula:

Bit Number Downstream = $(111–CDS2–CDS0)_B$

This behavior can also be seen in Figure 7.

CUS3–CUS0 CFI Upstream Bit Shift 3–0

Using this bit, the upstream frame may be shifted by up to 15 clock steps relative to the downstream frame. Figure 8 outlines this function.

Configurable Interface Subchannel Register (CSCR)

Access in the multiplexed microprocessor interface mode: Read or write, address: 36_H

Access in a demultiplexed microprocessor interface mode: Read or write, address B_H, OMDR : RBS = 1

Reset value: 00_H

bit 7							bit 0
SC31	SC30	SC21	SC20	SC11	SC10	SC01	SC00

SC#1–SC#0 Subchannel Control for the Logical Port Number

These bits select the sub-time slot position of the CFI Port Number to be handled according to the CM entries.

Only one channel may be mapped to one CFI time slot.

For example: programming SC01, SCO0 to 00, 16 kb/s channels consisting of the bits 6 and 7 of an 8-bit time slot, 32-kb/s time slots consisting of the bits 4 and 7 of an 8-bit time slot, or a full 64-kb/s time slot may be handled at the CFI Port 0.

For the CFI Modes 2 and 1, the following restrictions apply:

CFI Mode 2: SC31 = SC21 = SC11 = SC01;
SC30 = SC20 = SC10 = SC00;

CFI Mode 1: SC31 = SC11;
SC30 = SC10;
SC21 = SC01;
SC20 = SC00.

In CFI Mode 3, SC31 and SC30 control Ports 3 and 7, SC21 and SC20 Ports 2 and 6, SC11 and SC10 Ports 1 and 5, and SC00 and SC01 Ports 0 and 4.

Memory Access Registers

The memory access registers are used to
–directly access the data memory
–directly access the control memory
–program the switching function
–select the signaling application

One of these specific functions is selected by programming the memory access control register.

The address of the memory encodes the time slot and logical port numbers, the data memory addresses (Table 17) for the channels at the PCM interface, and the control memory addresses (Table 18) for the channels of CFI.

The C device performs a memory access specified by the actual content of the three memory access registers

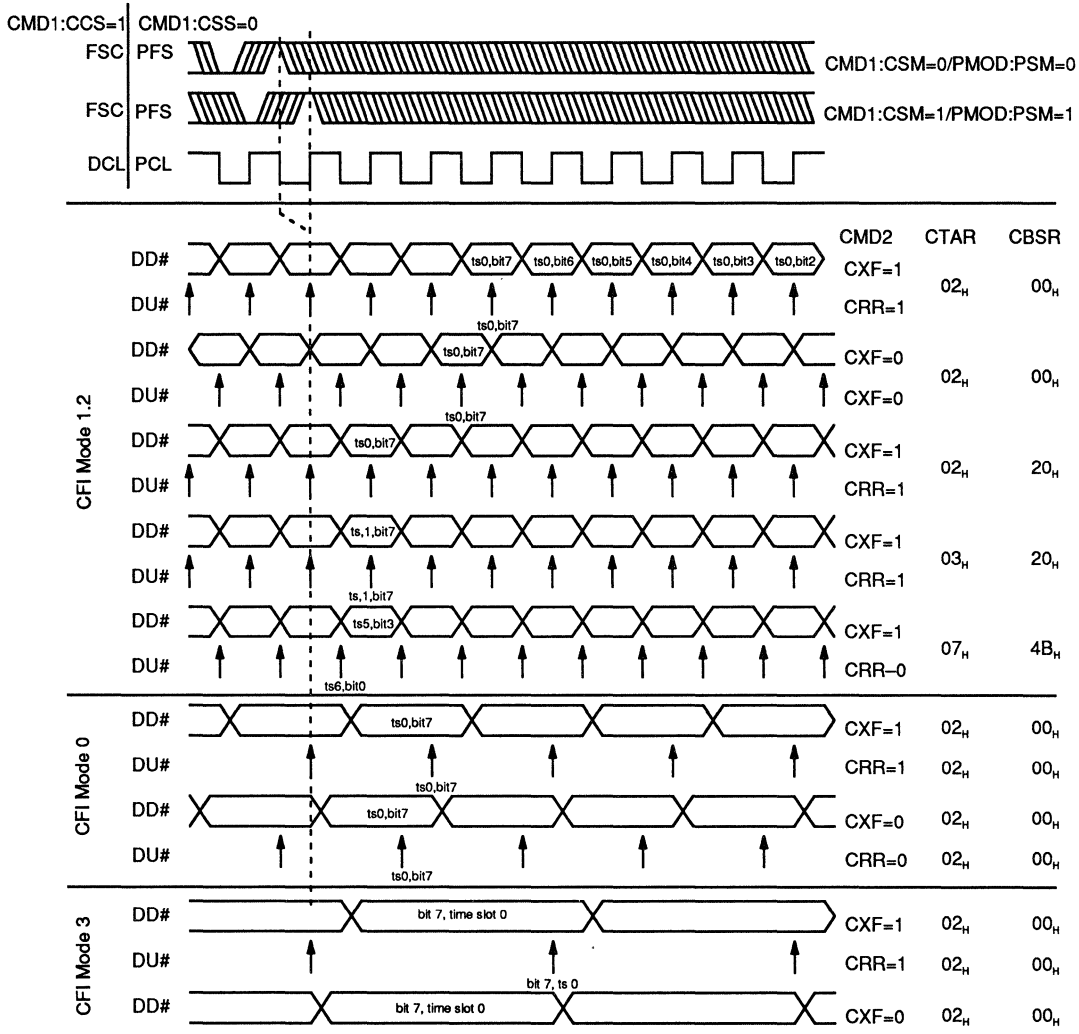


Figure 7. Configurable Interface Bit Timing

following every write operation at MACR. This access takes at most 9.5 reference clock cycles.

Memory Access Control Register (MACR)

Access in the multiplexed µP interface mode: Read and write, address: 00_H

Access in a demultiplexed µP interface mode: Read or write, address 0_H, OMDR:RBS = 0. A write access to this register initiates the memory access.

bit 7 **bit 0**

RWS	MOC3	MOC2	MOC1	MOC0/ CMC3	CMC2	CMC1	CMC3

RWS Read/Write Select

A logical "1" selects the EPIC for a read, a logical "0" for a write operation on the control or data memories.

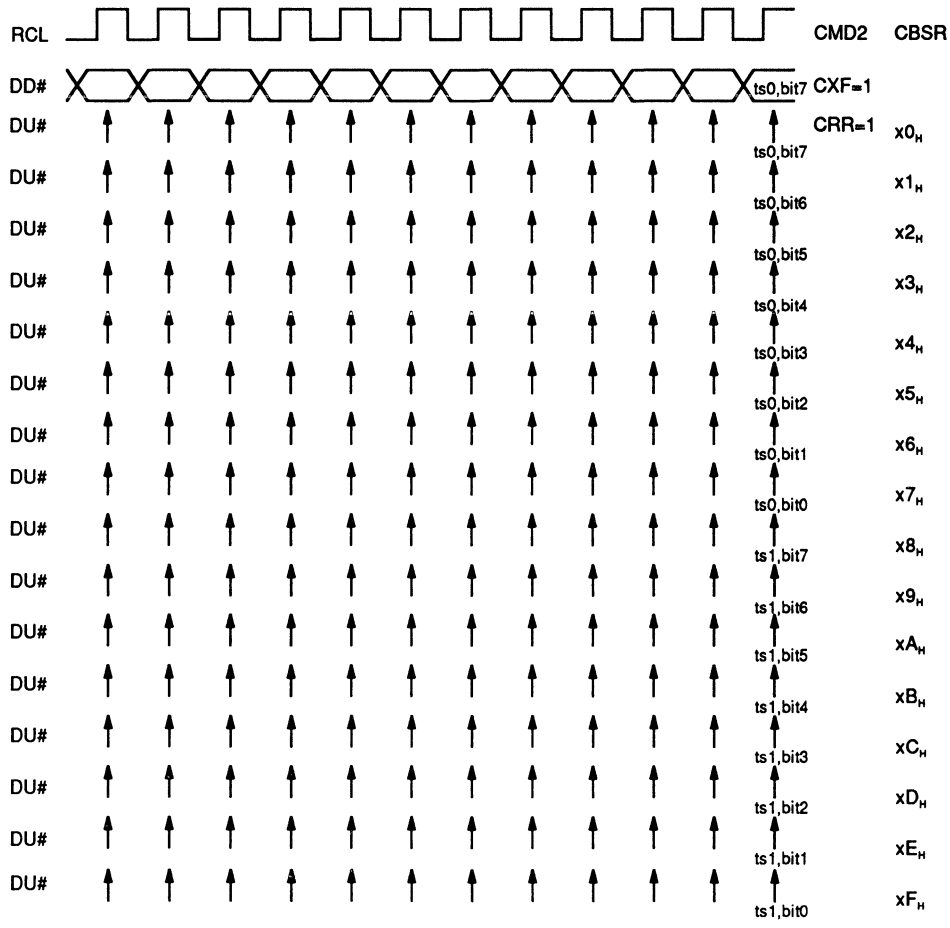


Figure 8. CFI Upstream Bit Shifting

Table 9. CFI Subchannel Assignments

SC#1	SC#0	Data Rate 64 kb/s	32 kb/s	16 kb/s
1	1	0–7	0–3	0–1
1	0	0–7	4–7	2–3
0	1	0–7	0–3	4–5
0	0	0–7	4–7	6–7

MOC3–MOC0 Memory Operation Code

Identifies type and destination of the memory operation according to the following tables.

PCM Interface Data Rate and Subchannel Selection

Table 10. Codes for the Selection of the PCM Interface and Subchannel

Code Bits MOC				Transferred Bits	Channel Bandwidth
0	0	0	0	Loopable unassigned	
0	0	0	1	Bits 7–0	64 kb/s
0	0	1	0	Bits 3–0	32 kb/s
0	0	1	1	Bits 7–4	32 kb/s
0	1	0	0	Bits 1–0	16 kb/s
0	1	0	1	Bits 3-2	16 kb/s
0	1	1	0	Bits 5-4	16 kb/s
0	1	1	1	Bits 7-6	16 kb/s

2

Using these codes in direct write accesses, the MADR content is transferred to the data memory subject to the subchannel selection (for example, PCM idle code programming). The code 0010 transfers MADR:MD7–MD4 to the data memory. These bits will then be output at the PCM interface in the bit 7–bit 4 positions of the chosen time slot (MAAR).

In direct read accesses, the complete data memory location content will be copied to MADR for all these codes.

Tristate Codes

Programming one of these codes, MADR:MDR3–MD0 is copied to a selected position (1100) or all positions (1101) of the upstream data memory tristate field. MD3 controls the PCM interface tristate function of the bits 7–6, MD2 of bits 5–4, MD1 of bits 3–2, and MD0 of bits 1–0 (high impedance = 0, low impedance = 1).

Resetting the complete DM tristate field takes 1035 reference clock periods.

The tristate codes are always used as direct codes (MOC bit position).

Table 11. Tristate Codes

Code Bits MOC	Function
1 1 0 0	Single Channel Tristate Control
1 1 0 1	Tristate Control Reset

CMC3–CMC0 Control Memory Code

Data to be entered to the CM code field in a write operation (MOC3–MOC1 = 111). Otherwise, CMCE2–0 have to be set to logical "0." For a read operation with MOC3–MOC1 = 111, CMCE3–0 must be fixed to logical "0."

If MADR contains data to be entered to a memory, the code values of the tables below have to be used in the MOC positions (direct access). Thus, for example, CFI idle codes may be programmed.

If MADR contains a pointer, the code values are programmed to the CMC positions of MACR (indirect access). This function is necessary to establish a switched connection. Then the MOC3–MOC1 bits need to be fixed to 111.

Table 12. CM Codes

Code Bits MOC	Transferred Bits	Function
1 0 0 1	MADR:MD7–MD0	Control Memory Access
1 1 1 X	MADR:MD7–MD0, MACR:CMC3–CMC0	Indirect Access

CM Codes

These codes are used to enter or get information to or from a CM location. Either the 8 bits of MADR (1001) or these 8 bits plus the 4 bits of MACR:CMC3–CMC0 (111X) are written to the CM.

The code 1001 may be programmed directly in the MOC bit positions (for example, change of a connection with unchanged sub-time slot selection) or indirectly in the CMC bit positions to enter a CFI idle code. Sixty-four kb/s is the only possible CFI idle code bandwidth.

The code 111X is applicable in indirect accesses (CMC bit positions) mainly to program the source channel, the target channel, and the channel bandwidth of a connection with one memory access. (For selecting the proper tristate functionality at the PCM highway of the device, a second memory access is necessary.) In read accesses either the 8 bits of the CM data field (1101) or the 4 bits of the CM control field are read and stored in MADR:MD3–MD0 (1110).

The selected code is transferred to a CM code field position selected by MAAR. During the switching operation, the code controls the bandwith and subtime slot position at the PCM interface on a per-channel basis. Thus, more than one CFI subtime slot can be mapped to one PCM time slot.

Programming the loopable unassigned code (0000) for an upstream connection, the corresponding upstream data memory location is not written during switching operation (unassigned channel). In downstream connections, however, the location is read, but not output at the CFI and, thus, can be used to establish loops.

Table 13. Codes for the Selection of the PCM Interface Subchannels

CMC3	CMC2	CMC1	CMC0	Transferred Bits	Channel Bandwidth
0	0	0	0	—	—
0	0	0	1	Bits 7–0	64 kb/s
0	0	1	1	Bits 7–4	32 kb/s
0	0	1	0	Bits 3–0	32 kb/s
0	1	1	1	Bits 7–6	16 kb/s
0	1	1	0	Bits 5–4	16 kb/s
0	1	0	1	Bits 3–2	16 kb/s
0	1	0	0	Bits 1–0	16 kb/s

Table 14. CFI Codes

CMC3	CMC2	CMC1	CMC0	Transferred Bits
1	0	1	1	CS Channel
1	0	1	0	CS Channel
1	0	0	0	CS Channel
1	0	0	1	CS Channel, µP channel setup

For a description of the CS Channel Function, please refer to the following pages.

The CMC3–CMC0 = 1001 choice is also used for accessing a 64-kb/s CFI Channel. In this case the CM data field entry is exchanged with the CFI port and time slot coded by the CM locations address according to Table 18. Thus, a CFI idle channel may be set up, or CFI data may be read via the µP interface.

CS Codes

The CS codes control the CS channel (Control channel in IOM applications, Signaling channel in SLD applications). They need to be transferred to the CM code field, hence they are used in the CMC bit positions. To activate the EPIC's signaling handling function, two CFI codes per SLA (Subscriber Line Access) are programmed to two consecutive CM code field positions starting with the even address. These two positions map to the preprocessed time slots, for example, the Monitor and Feature Control channels in IOM configuration or the Control and Signaling channels in SLD configuration. The signaling schemes defined by these codes are explained in Figures 15 and 16 for the upstream and downstream directions.

The signaling schemes may be programmed to the SLAs independently from one another.

Memory Access Address Register (MAAR)

Access in the multiplexed microprocessor interface mode: Read or write, address: 02_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 1_H, OMDR : RB8 = 0

bit 7 **bit 0**

U/D	MA6	MA5	MA4	MA3	MA2	MA1	MA0

This register buffers the microprocessor address to the CM or DM. In DM accesses, the address encodes the time slot and port of the PCM interface; in CM accesses, the CFI interface.

U/D

For both CM and DM accesses, the U/D bit being programmed as logical "0" selects the downstream direction in the specific memory; a logical "1" selects the upstream direction.

Table 15. Programming the CM for Downstream Signaling Handling

Application Scheme	MAAR	CM Entries CMC3–0	CM Entries MADR	Output at the Configurable Interface Preprocessed Channels	
Decentral D-Channel Handling	Even	1 0 0 0	1 1 C/I 1 1	mmmmmmmm	COS C/I mm
	Odd	1 0 1 1	0 0 0 0 0 0 0 0	Monitor Channel	Control Channel
Central D-Channel Handling	Even	1 0 1 0	1 1 C/I 1 1	mmmmmmmm	P P C/I mm
	Odd	PCM code for a 2-bit sub-time slot	PCM channel	Monitor Channel	Control Channel
6-Bit Signaling (e.g., analog IOM)	Even	1 0 1 0	S G 1 1	mmmmmmmm	S G mm
	Odd	1 0 1 1	0 0 0 0 0 0 0 0	Monitor Channel	Control Channel
8-Bit Signaling (e.g., SLD)	Even	1 0 1 0	S G	mmmmmmmm	S G
	Odd	1 0 1 1	0 0 0 0 0 0 0 0	Feature Control Channel	Signaling Channel

Table 16. Programming the CM for Upstream Signaling Handling

Application	CM Entries			Input from the Configurable Interface	
Scheme	**Address**	**Code Bits**	**Data Bits to Read**	**Preprocessed Channels**	
Handling	Even	1 0 0 0	x x C/I x x	m m m m m m m m	COS C/I m m
	Odd	0 0 0 0	x x x x x x x x	Monitor Channel	Control Channel
Central D-Channel Handling	Even	1 0 0 0	x x C/I x x	m m m m m m m m	P P C/I m m
	Odd	PCM code for a 2-bit sub-time slot	PCM channel	Monitor Channel	Control Channel
6-Bit Signaling (e.g., analog IOM)	Even	1 0 1 0	Actual Value x x	m m m m m m m m	Value m m
	Odd	1 0 1 0	Stable Value x x	Monitor Channel	Control Channel
Bidirectional 8-Bit Signaling (e.g., SLD)	Even	1 0 1 1	Actual Value	m m m m m m m m	Value
	Odd	1 0 1 1	Stable Value	Feature Control Channel	Signaling Channel

Legend

m These bits are treated by the monitor handler.

COS The output resistance of these bits is determined by OMDR:COS.

C/I These bits are exchanged between the CFI in/output and the CM. A change of the C/I bits in upstream direction causes an interrupt (ISTA : SFI). The location of the change is stored in CSFIFO.

P This D-channel information is switched to and from the PCM interface. The PCM time slot and sub-time slot positions are defined by the pointer and the code field entry residing in the odd CM entry of the respective SLA.

SIG These bits are included in the CFI data stream from the CM.

Value These bits are extracted from the CFI upstream data. The value that occurred in the last frame is stored in the actual value bits of the even address CM location, the stable value in the CM location of the odd address. The stable value is found implementing the double last look algorithm: a new value must reoccur after the double last look period programmed in TIMR to become a stable value. A new stable value causes an interrupt (ISTA:SFI, location in SFIFO).

MA6–MA0

The time slot to be handled is programmed to the EPIC via the time slot number bits according to Tables 17 and 18. In all modes, the MA6 bit is the most significant and the MA0 bit is the least significant time slot number bit.

The logical port numbers used for programming may be looked up in Tables 3 and 7 for the PCM ports and for the IOM ports. The bidirectional pin and logical port numbers match.

Table 17. Programming PCM Ports and Time Slots for the Data Memory

Data Memory Address		
PCM Mode 2	Bit U/D Bit MA6 to MA0	Direction Selection Time Slot Number
PCM Mode 1	Bit U/D Bit MA6 to MA3, MA1, MA0 Bit MA2	Direction Selection Time Slot Number Logical PCM Port Number
PCM Mode 0	Bit U/D Bit MA6 to MA3, MA0 Bit MA2 to MA1	Direction Selection Time Slot Number Logical PCM Port Number

Table 18. Programming CFI Ports and Time Slots for the Control Memory

Control Memory Address		
CFI Mode 2	Bit U/D Bit MA6 to MA0	Direction Selection Time Slot Number
CFI Mode 1	Bit U/D Bit MA6 to MA3, MA1, MA0 Bit MA1	Direction Selection Time Slot Number Logical IOM Port Number
CFI Mode 0	Bit U/D Bit MA6 to MA3, MA0 Bit MA2 to MA1	Direction Selection Time Slot Number Logical IOM Port Number
CFI Mode 3	Bit U/D Bit MA6 to MA4, MA0 Bit MA3 to MA1	Direction Selection Time Slot Number Logical Bidirectional Port Number

Memory Access Data Register (MADR)

Access in the multiplexed microprocessor interface mode: Read or write, address: 04_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 2_H, OMDR:RBS = 0

bit 7							bit 0
MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0

This register buffers the microprocessor data to the data and control memories.

In DM accesses, real data are transferred, whereas in CM accesses, when setting up a connection, this register may also contain a pointer to a DM address.

Synchron Transfer Registers

The synchron transfer period between the maskable synchron transfer overflow interrupt (ISTA:SOV, MASK:SOV) and the maskable synchron transfer interrupt (ISTA:SIN, MASK:SIN) is 16 RCL periods per active synchron transfer channel.

Synchron Transfer Data Register A (STDA)

Access in the multiplexed microprocessor interface mode: Read or write, address: 06_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 3_H, OMDR:RBS = 0

bit 7							bit 0
MTDA7	MTDA6	MTDA5	MTDA4	MTDA3	MTDA2	MTDA1	MTDA0

MTDA7–MTDA0 Microprocessor Transfer Data for Channel A

These bits are extracted by the EPIC from the source data stream as specified in SARA, and included in the destination data stream at the location indicated by the SAXA content.

Synchron Transfer Data Register B (STDB)

Access in the multiplexed microprocessor interface mode: Write or read, address: 08_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 4_H, OMDR:RBS = 0

bit 7							bit 0
MTDB7	MTDB6	MTDB5	MTDB4	MTDB3	MTDB2	MTDB1	MTDB0

MTDB7–MTDB0 Microprocessor Transfer Data for Channel B

These bits are extracted from the source data stream as specified in SARB, and included into the destination data stream at the location indicated by the SAXB content.

Synchron Transfer Receive Address Receive A (SARA)

Access in the multiplexed microprocessor interface mode: Read or write, address: $0A_H$

Access in a demultiplexed microprocessor interface mode: Read or write, address 5_H, OMDR:RBS = 0

bit 7							bit 0
ISRA	MTRA6	MTRA5	MTRA4	MTRA3	MTRA2	MTRA1	MTRA0

ISRA Interface Select Receive for Channel A

Selects the PCM interface (logical "0") or the configurable interface (logical "1") as source interface for the synchron transfer Channel A.

MTRA6–MTRA0 Microprocessor Transfer Receive for Channel A

Identifies the source port and time slot as stated in table dmadd and cmadd for the PCM and CFG interfaces being selected by ISRA, respectively.

Synchron Transfer Receive Address Register B (SARB)

Access in the multiplexed microprocessor interface mode: Read or write, address: $0C_H$

Access in a demultiplexed microprocessor interface mode: Read or write, address 9_H, OMDR:RBS = 0

bit 7 bit 0

ISRB	MTRB6	MTRB5	MTRB4	MTRB3	MTRB2	MTRB1	MTRB0

ISRB Interface Select Receive for Channel B

Selects the PCM interface (logical "0") or the configurable interface (logical "1") as source interface for the synchron transfer Channel B.

MTRB6–MTRB0 Microprocessor Transfer Receive for Channel B

Identifies the source port and time slot as stated in Tables 17 and 18 for the PCM and CFG interfaces being selected by ISRB, respectively.

Synchron Transfer Transmit Address Register A (SAXA)

Access in the multiplexed microprocessor interface mode: Read or write, address: $0E_H$

bit 7 bit 0

ISXA	MTXA6	MTXA5	MTXA4	MTXA3	MTXA2	MTXA1	MTXA0

ISXA Interface Select Transmit for Channel A

Selects the PCM interface (logical "0") or the configurable interface (logical "1") as target interface for the synchron transfer Channel A.

MTXA6–MTXA0 Microprocessor Transfer Transmit for Channel A

Identifies the destination port and time slot as stated in Tables 17 and 18, in case the PCM and configurable interfaces are selected, respectively.

Synchron Transfer Transmit Address Register B (SAXB)

Access in the multiplexed microprocessor Interface mode: Read or write, address: 10_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 8_H, OMDR:RBS = 0

bit 7 bit 0

ISXB	MTXB6	MTXB5	MTXB4	MTXB3	MTXB2	MTXB1	MTXB0

ISXB Interface Select Transmit for Channel B

Selects the PCM interface (logical "0") or the configurable interface (logical "1") as target interface for the synchron transfer Channel B.

MTXB6–MTXB0 Microprocessor Transfer Transmit for Channel B

Identifies the destination port and time slot as stated in Table 17 if the PCM interface is selected, and Table 18 in case the configurable interface is programmed.

Synchron Transfer Control Register (STCR)

Access in the multiplexed microprocessor interface mode: Read or write, address: 12_H

Access in a demultiplexed microprocessor interface mode: Read or write, address 9_H, OMDR:RBS = 0

Reset value: 00_H

bit 7 bit 0

TBE	TAE	CTB2	CTB1	CTB0	CTA2	CTA1	CTA0

TAE, TBE Transfer Channel A (B) Enable

A logical "1" enables the microprocessor transfer; a logical "0" disables the transfer of the corresponding channel.

CTA2–CTA0, CTB2–CTB0 Channel Type A (B)

These bits determine the bandwidth of the channel and the position of the relevant bits in a time slot according to Table 19.

Table 19. Synchron Transfer Channel Type

CT#2	CT#1	CT#0	Bandwidth	Relevant Bits
0	0	0	Not Allowed	—
0	0	1	64 kb/s	bits 7–0
0	1	0	32 kb/s	bits 3–0
0	1	1	32 kb/s	bits 7–4
1	0	0	16 kb/s	bits 1–0
1	0	1	16 kb/s	bits 3–2
1	1	0	16 kb/s	bits 5–4
1	1	1	16 kb/s	bits 7–6

The Monitor/Feature Control Registers

MF Channel Active Indication Register (MFAIR)

Access in the multiplexed microprocessor interface mode: Read, address: 14_H

Access in a demultiplexed microprocessor interface mode: Read or write, address A_H, OMDR:RBS = 0

Reset value: undefined

bit 7							bit 0
0	SO	SAD5	SAD4	SAD3	SAD2	SAD1	SAD0

SO MF Channel Search On

This bit indicates if the EPIC is still busy looking for an active MF channel (logical "1") or not (logical "0").

SAD5–SAD0 Subscriber Address

With OMDR:MFPS and CMDR:MFSO set to logical "1" (search for Command/Acknowledgment within the handshake procedure) after an interrupt (ISTA, MASK:MFFE), these bits point to the port and time slot where an active MF channel was found. For the coding of time slot and port numbers, refer to MFSAR: SAD5–SAD0.

MF Channel Subscriber Address Register (MFSAR)

Access in the multiplexed microprocessor interface mode: Write, address: 14_H

Access in a demultiplexed microprocessor interface mode: Read or write, address A_H, OMDR:RBS = 0

Reset value: undefined

bit 7							bit 0
MFTC1	MFTC0	SAD5	SAD4	SAD3	SAD2	SAD1	SAD0

MFTC 1–0 MF Channel Transfer Control 1,0

These bits, in addition to CMDR:MFT1,0 and OMDR:MFPS, control the MF Channel transfer as detailed in the following tables.

Transmit Selection (CMDR:MTF1,MTF0 = 01)

Table 20. Transmit Selection

MFTC1,0		Recipient
0	0	As programmed in MFSAR:SAD5–SAD0
0	1	Broadcast: all MF channels
1	0	Test operation
1	1	Reserved

For broadcast transmission with active handshake protocol, instantaneous acknowledgment by all recipients is assumed and arriving acknowledgments are ignored. The MF bytes are transmitted at full speed, that is, one byte per frame. In test operation (MFT1,0 = 10), the data in MFFIFO is not transmitted and can be read instantaneously.

Selections Transmit + Receive and Transmit and Interrupt

(CMDR:MTF1,MTF0 = 10 or 11)

Table 21. Receive Byte Count

MFTC1,0		Number of expected receive bytes
0	0	1
0	1	2 (not allowed for active handshake protocol)
1	0	8 (not allowed for active handshake protocol)
1	1	16 (not allowed for active handshake protocol)

Table 21 states the number of bytes written into MFFIFO with the handshake protocol not active. When the handshake protocol is active, the selection MFTC1,0 = 00 is the only allowed choice.

SAD5–SAD0

These bits define the addressed subscriber. To achieve a definite time slot identification according to Table 18, an additional least significant bit set to logical "0" and an additional most significant bit, depending on the transfer direction (CMDR:MFT1,MFT0), has to be assumed.

MF Channel FIFO (MFFIFO)

Access in the multiplexed microprocessor interface mode: Read or write, address: 16_H, 16 bytes deep

Access in a demultiplexed microprocessor interface mode: Read or write, address B_H, OMDR:RBS = 0

Reset value: the MFFIFO is empty

bit 7							bit 0
MFD7	MFD6	MFD5	MFD4	MFD3	MFD2	MFD1	MFD0

The contents of this register are transferred byte by byte in the monitor (IOM configuration) or feature control (SLD configuration) channels as specified by MFSAR: SAD5–SAD0. MFFIFO is reset by setting CMDR: MFFR or by selecting OMDR:OMD0 = 0.

The Status/Control Registers

Signaling FIFO (C/I FIFO)

Access in the multiplexed microprocessor interface mode: Read, address: 18_H, 9 bytes deep

Access in a demultiplexed microprocessor interface mode: Read or write, address C_H, OMDR:RB8 = 0

Reset value: 0XXX XXXX

bit 7							bit 0
SBV	SAD6	SAD5	SAD4	SAD3	SAD2	SAD1	SAD0

SBV Signaling Byte Valid

A logical "1" indicates SAD6–SAD0 to be valid. This bit is set to logical "0" if the corresponding SAD6–SAD0 is invalid.

The SBV bits of all C/I FIFO bytes are reset by selecting OMDR:OMS0 = 0 or by instructing the EPIC to reset the C/I FIFO (CMDR:MFFR = 1)

SAD6–SAD0 Subscriber Address

This address points to the location in the CM where the received C/I channel has changed. If SBV signals a valid pointer, all 8 bits of the signaling FIFO may be used as a CM address, since SBV being logical "1" identifies the upstream block of the CM. If SAD0 is logical "0" (even address), a C/I channel of a digital subscriber is addressed. If it is logical "1" (odd address) the stable value of a signaling channel (analog subscriber) is addressed.

Timer Register (TIMR)

Access in the multiplexed microprocessor interface mode: Write, address: 18_H

Access in a demultiplexed microprocessor interface mode: Read or write, address C_H, OMDR:RBS = 0

Reset value: 00_H

bit 7							bit 0
SSR	TVAL6	TVAL5	TVAL4	TVAL3	TVAL2	TVAL1	TVAL0

SSR Signaling Channel Sample Rate

The double last look period is fixed to 125 microseconds (logical "0") or determined by TVAL6–TVAL0 (logical "1").

TVAL6–0 Timer Value

The timer period is given by these bits in intervals of 250 msec and is used in three functions of the EPIC:

–double last look algorithm (controlled by SSR)

–interrupt generation (CMDR:ST1)

–CFI multiframe generation (CMD2:FC2–FC0)

The timer is started as soon as CMDR:ST1 is set to logical "1" and stopped by programming the timer register or by selecting OMDR:OMS0 = 0.

Status Register (STAR)

Access in the multiplexed microprocessor interface mode: Read, address: $1A_H$

Access in a demultiplexed microprocessor interface mode: Read or write, address D_H, OMDR:RBS = 0

Reset value: 05_H

bit 7							bit 0
MAC	TAC	PSS	MFTO	MFAB	MFAE	MFRW	MFFE

MAC Memory Access

A memory access is in operation if this bit is logical "1." Hence, the memory access registers may not be used.

TAC Timer Active

If this bit is set, the timer runs (TIMR).

PSS PCM Synchronization Status

The PCM interface is synchronized (logical "1") or not synchronized (logical "0") (ISTA:PFI, MASK:PFI, PBNR).

MFTO MF Channel Transfer in Operation

The MF channel transfer is still in operation (logical "1") or completed (logical "0").

MFAB MF Channel Transfer Aborted

A logical "1" indicates that the remote transmitter aborted a handshaked message transfer.

MFAE MF FIFO Access Enable

The MFFIFO may be either read or written (logical "1") or may not be accessed (logical "0").

MFRW MF FIFO Read/Write

If MFAE = 1, the MFFIFO may be read (logical "1") or is ready to be written (logical "0").

MFFE MF FIFO Empty

The MFFIFO is empty (logical "1") or not empty (logical "0").

Command Register (CMDR)

Access in the multiplexed microprocessor interface mode: Write, address: $1A_H$

Access in a demultiplexed microprocessor interface mode: Read or write, address D_H, OMDR:RBS = 0

Reset value: 00_H

bit 7							bit 0
0	ST	TIG	CFR	MFT1	MFT0	MFSO	MFFR

ST Start Timer

Setting this bit to logical "1" starts the timer running cyclically from 0 to the value programmed in TIMR: TVAL6–0. A logical "0" does not affect the timer.

TIG Timer Interrupt Generation

Setting this bit together with ST to logical "1" causes the C device to generate a periodic interrupt whenever the timer runs out. (See TIMR.TVAL6–0 and ISTA:TIN.) A logical "0" disables the interrupt.

CFR C/I FIFO Reset

Setting this bit resets the signaling FIFO, which takes 2 RCL periods.

MFT1, MFT0 MF Channel Transfer

These bits start an MF Channel transfer according to the following table:

Table 22. MF Channel Transfer Direction Control

MFT1	MFT0	Function
0	0	Inactive
0	1	Transmit
1	0	Transmit + Receive
1	1	Transmit + Interrupt

The contents of MFFIFO are exchanged in the channel programmed to MFSAR:SAD5–SAD0 controlled by MFSAR:MFTC1–0. By programming MFT1, MFT0 = 01, the MFFIFO contents are transmitted. By selecting MFT1, MFT0 = 10, they are transmitted and an answer is expected. MFT1, MFT0 = 11 is chosen if more than 16 bytes are to be transmitted, the first 16 of which reside in MFFIFO. After transmitting the MFFIFO contents or receiving the answer, the EPIC generates a maskable interrupt (ISTA:MFFI, MASK:MFFI). The actual state of the transfer can be evaluated by reading STAR:MFTO–MFTE.

MFSO MF Channel Search On

If this bit is set to logical "1," the EPIC starts to search for active MF channels. If an active channel is found, the channel address is stored in MFSAR and an interrupt is generated (ISTA:MAC). The search is stopped when an active MF channel has been found or when OMDR:OMS0 = 0

If the MFFIFO is empty and Transmit + Receive is selected, the EPIC waits for received bytes.

With the handshake protocol active (OMDR:MFPS = 1), the EPIC expects received bytes in the time slot it also sent MFSAR:SAD5–SAD0). However, for the Transmit + Interrupt selection only, and only when the handshake protocol is not active, it expects received bytes at the same line, but 4 time slots later than programmed to MFSAR:SAD5–SAD0 (SLD configuration).

MFFR MF FIFO Reset

Setting this bit resets the MFFIFO within 2 RCL periods, after which MFFR is reset again.

Interrupt Status Register (ISTA)

Access in the multiplexed microprocessor interface mode: Read, address: 1 C_H

Access in a demultiplexed microprocessor interface mode: Read or write, address E_H, OMDR:RBS = 0

Reset value: 00_H

bit 7							bit 0
TIN	SFI	MFFI	MAC	PFI	PIM	SIN	SOV

TIN Timer Interrupt

If this bit is set to logical "1," a timer interrupt (CMDR:ST0, TIMR) has occurred. The bit is reset by reading ISTA.

SFI Signaling FIFO Interrupt

A logical "1" indicates a change in an upstream C/I or signaling channel. The bit is reset when FIFO is empty.

MFFI MF FIFO Interrupt

If this bit is set, the last MF channel command (issued by CMDR:MFT1, MFT0) has been executed and the EPIC is ready to accept the next command (STAR:MFTO–MFFE). This bit is reset by the ISTA access.

MAC Monitor Channel Active Interrupt

A logical "1" indicates an active monitor channel, for the case CMDR:MFSO = 1. MFAIR contains the address of this channel. This bit is cleared by the ISTA access.

PFI PCM Framing Interrupt

This bit being logical "1" indicates lost or regained synchronization at the PCM interface. Synchronization is considered lost by the EPIC if the PFS signal is not repeated with a period determined by PBUP. Synchronization is considered regained if two consecutive PFS pulses with the correct period are received. PFI is reset by reading ISTA.

PIM PCM Input Mismatch

A logical "1" indicates a PCM input mismatch (PICM,PMOD: AIS0,1); PIM is reset by reading ISTA.

SIN Synchron Transfer Interrupt

The synchron transfer registers can be read and written if this bit is logical "1"; this bit is reset by reading ISTA.

SOV Synchron Transfer Overflow

A logical "1" indicates that the data of one of the active synchron transfer channels (STCR:TAE,TBE) has not been accessed within the microprocessor access period of the synchron transfer facility. This bit is reset by reading ISTA.

Mask Register (MASK)

Access in the multiplexed microprocessor interface mode: Write, address: 1 C_H

Access in a demultiplexed microprocessor interface mode: Read or write, address E_H, OMDR:RBS = 0

Reset value: 00_H

bit 7							bit 0
TIN	SFI	MFFI	MFAC	PFI	PIM	SIN	SOV

A logical "1" disables the corresponding interrupt as described in ISTA.

A masked interrupt is stored internally and reported in ISTA immediately, if the mask is released.

A SFI interrupt, however, is also reported in ISTA, if it is masked. In this case an interrupt is not generated.

Operation Mode Register (OMDR)

Access in the multiplexed microprocessor interface mode: Read or write. 1 E_H or $3E_H$

Access in a demultiplexed microprocessor interface mode: Read or write, address F_H, OMDR:RB8 = X

Reset value: 00_H

bit 7							bit 0
OMS1	OMS0	PSB	PTL	COS	MFPS	CSB	RBS

0MS1–0MS0 Operation Mode Select

These bits determine the operation mode the EPIC is working in according to Table 23.

Table 23. Operation Modes

OMS1	OMS0	Function
0	0	CM reset mode: In this mode the EPIC does not operate normally. Instead, a CM Access via the memory access registers accesses all CM positions. Only accesses with MACR:MOC3 = 1 are performed. A typical application is resetting the CM (MACR = 70_H, MADR:XX_H). Such a complete CM access is finished within 256 RCL periods.
1	0	CM initialization mode: In this selection the EPIC does not operate normally. This selection allows for a fast programming of the CM (for example, for monitor and C/ channel selection), reducing the memory access busy time to maximally 2.5 RCL cycles.
1	1	Normal operation mode: Memory access period 9.5 RCL cycles maximally.
0	1	Test mode: In this mode the EPIC sustains normal operation. However, a memory access does not only affect the one location of MAAR, but all positions in the specific memory.

Setting OMS0 to logical "0" stops the software timer (CMDR:ST1; TIMR) and resets MFFIFO as well as C/I FIFO.

PSB PCM Standby

A logical "0" switches the PCM interface outputs to high impedance.

COS CFI Output Driver Selection

Programming a logical "1," the CFI output drivers are open drain drivers for unassigned channels; programming a logical "0," they are tristate.

MFPS MF Channel Protocol Selection

A logical "1" enables the handshake facility of the EPIC; with a logical "0" it is disabled.

PTL PCM Test Loop

The PCM interface output and input lines are internally connected if PTL is logical "1." In this case, the PCM data are still output according to the DM tristate field, but the PCM input is disabled.

CSB CFI Standby

A logical "0" switches the CFI outputs to high impedance.

RBS Register Bank Selection

In the demultiplexed microprocessor access mode, this bit switches between the registers mainly used for initialization (logical "1") and those mainly used in operation (logical "0").

Version Number Status Register (VNSR)

Access in the multiplexed microprocessor interface mode: Read, address: $3A_H$

Access in a demultiplexed microprocessor interface mode: Read or write, address D_H, OMDR:RBS = 1

Reset value: $0X_H$

bit 7							bit 0
IR	0	0	0	VN3	VN2	VN1	VN0

These bits contain the version number of the device according to the following table:

VN3	VN2	VN1	VN0	Device Versions
0	0	0	0	A1

IR Initialization Request

The code memory has to be reprogrammed due to loss of data (IR = 1). The IR bit is set after power failure or in appropriate clocking and reset when a code memory initialization (CMDR:CMS1,0 = 10) is finished.

APPLICATIONS

Communication Multiplexers

The non-blocking switching capability for various bandwidths implemented in the EPIC makes the circuit suitable for use in communication multiplexers. Due to the data rate programmability of the configurable and PCM interfaces, data rate adaption (for example, between 1544 and 2048 kb/s systems) can be accomplished.

Concentrators

Due to the high data rates of up to 8192 kb/s, the EPIC can be used in concentrator applications.

Central Switches

The EPIC is a non-blocking switch for up to 128 channels per direction. The channel bandwidth can be programmed to 16, 32, or 64 kb/s. The PCM and configurable interfaces are programmable for a wide variety of data rates from 8 to 8192 kb/s. PCM and configurable interfaces can be operated with different clock frequencies. Thus, the EPIC can be used in central switches and for data rate adaption.

Line Cards

The EPIC is designed to operate in three different digital or analog line card architectures. For a schematic summary of these possible line card configurations, refer to Figure 9.

With its configurable interface being programmed as an SLD interface, the EPIC can communicate with SLD-compatible devices (for example, ISDN Subscriber Access Controller ISAC-S, Am2085). The EPIC provides the signals for up to 16 ISAC-Ss, supporting up to 16 S Interfaces.

Alternatively, the configurable interface may be selected as IOM Interface, which is compatible to both the multiplexed IOM1 and the IOM Rev. 2 Interface.

In the case of an IOM Rev. 2 interface, the EPIC supports up to 32 ISDN or 64 voice subscribers. They are connected via the SBCX (Am2081), Am2096, or IEC (Am20901 and Am20902) and a digital loop. In both cases, either the Am2070 or the IDEC (Am2075) may perform the D-channel handling.

Digital Line Cards

On digital line cards, the EPIC performs the switching function for up to 32 ISDN subscribers between the PCM system highways and the IOM interfaces. Moreover, it has the Layer 1 controlling capability of buffering the C/I and monitor channels of the IOM interface.

The EPIC can be operated in tandem (one device is active, another one is a backup device). The backup device can instantaneously take over for the active

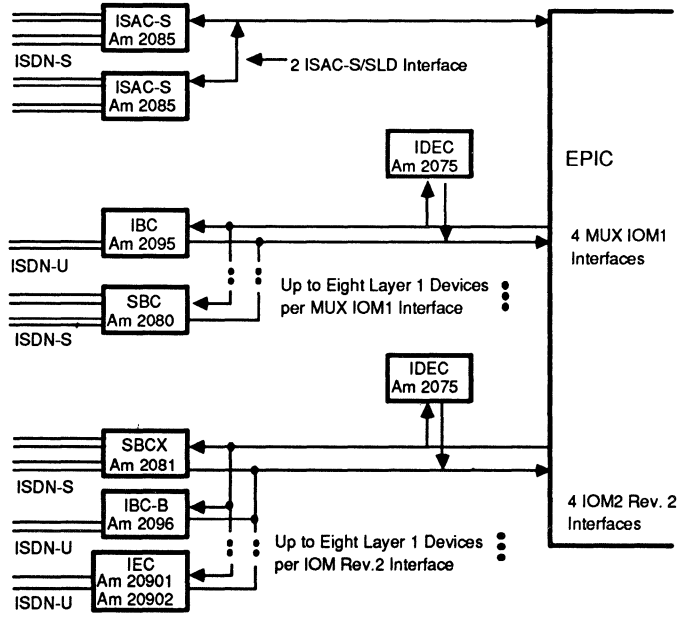


Figure 9. Schematic Summary of the Line Card

device when it fails. Due to this tandem operation capability and the high number of ISDN subscribers that can be connected to one EPIC, the use of single line cards is feasible.

Several architectures are possible.

In completely decentralized D-channel processing architectures (Figure 10), the processing capacity of a line card is usually dimensioned to avoid blocking situations even under maximum conceivable D-channel traffic conditions. In such an architecture, the EPIC switches the B channels and performs C/I and monitor channel control. The IDECs handle the Layer 2 functions for signaling and data packets in the D channel and transfer the extracted data via the µP and an HDLC controller (for example, the HSCX Enhanced High Level Serial Communication Controller Am82525) to the system. One of the channels of the HSCX may be used, for example, for the signaling information, the other for data packets. The HSCX may access either a time slot of programmable bandwidth on one of the system highways (Figure 10) or a separate signaling highway (Figure 11). In both cases, the highway capacity used for packet traffic can be shared among several line cards due to the statistical multiplexing capabilities of the HSCX.

In an architecture with completely central D-channel handling (Figure 11), the EPIC switches the B and D channels and performs the C/I and monitor channel control functions.

The line card microcontroller programs the EPIC and is connected to the group control via a signaling highway and an HSCX. Moreover, the EPIC controls the Layer 1 protocol on the IOM interface, buffering the C/I and monitor channels to the microprocessor.

A third possibility is a mixed architecture with central packet data and decentralized signal handling. This is a very flexible architecture that reduces the dynamic load of central processing units by evaluating the signaling information on the line card. For this case, any increase of packet data traffic does not necessitate any change in the architecture because the line cards do not

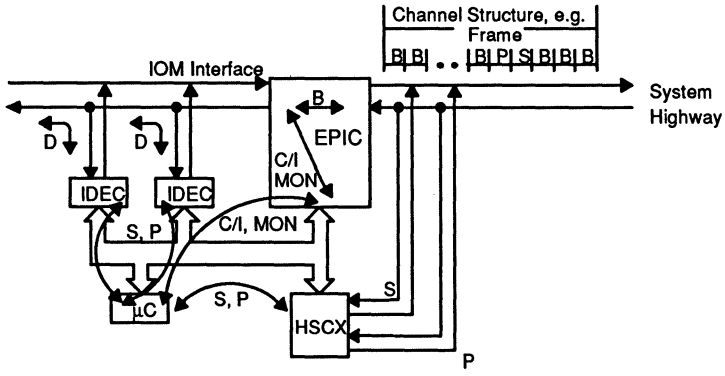


Figure 10. Completely Decentralized Packet Switching Digital Line Card Architecture

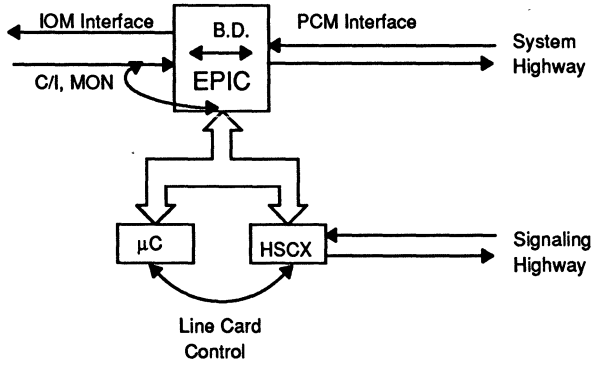


Figure 11. Digital Line Card Architecture with Completely Central D-Channel Handling

have to be modified. The central packet handling unit can simply be expanded.

For such an architecture, the EPIC performs B- and D-channel switching in addition to C/I and monitor channel control. The IDECs handle the signaling data of the D channel. These messages are transferred to the group controller via the microprocessor and an HDLC controller. The packet data of the D channel are switched to the system highways and processed by the central packet unit.

In this architecture, the EPIC switches the B channels from IOM Port A (Figure 12) to the PCM interface. The IDEC works in a master/slave configuration. Therefore, an additional collision resolution line is needed. The IDEC separates signaling from data packets. The signaling messages are transferred to the microcontroller, which in turn hands them over to the group controller using the HSCX.

The packet data are processed differently. Together with the collision resolution line they are handled by the IDEC at another IOM port (Port B). The EPIC switches the channels of these ports to the PCM interface as shown in Figure 12.

In such a configuration, the "p" packets and the collision resolution signal occupy one of the IOM ports available at the configurable interface. This reduces the total switching capability of the EPIC to 24 ISDN subscribers.

Alternately, the packet data and the collision line can be directly exchanged between the IDEC and the PCM highway. The EPIC then simply switches the B channels (see Figure 13). The packet data are separated by the IDEC and placed on the PCM highway. Thus, the full 32-subscriber switching capability of the EPIC is retained.

Packet Handlers

The EPIC is an important building block for networks based on either central, decentralized, or mixed signaling and packet data handling architectures. Its flexibility allows for the modification of the packet handling architecture according to changing needs.

Thus, it may be useful to add central packet handling groups to a network originally based on decentralized signaling packet handling. This may be the case if growing data packet traffic exceeds the initial capacity of the network. The result is a mixed architecture.

On the other hand, increasing packet handling demand on a few dedicated subscriber lines calls for solutions that back up the capacity at these few decentralized line cards.

In both of these cases and several other applications, the EPIC is a powerful device for solving the problem of packet handling. In most applications it is used together with the IDEC (ISDN D-Channel Exchange Controller).

Decentralized and mixed packet handling has already been covered in the line card chapter. In the following, the centralized signaling/data packet handlers built up with the EPIC will be described.

Central packet handling is used if many subscribers with a generally low demand for packet switching are to be connected to a system. Concentrating the packet servers for multiple users eliminates the need to provide a packet server channel for every user. The overall number of packet server channels can thus be reduced.

In such a central packet handling group, the EPIC performs the switching and concentrator function. It connects a variable number of PCM highways to the packet handler internal highway. HDLC controllers are

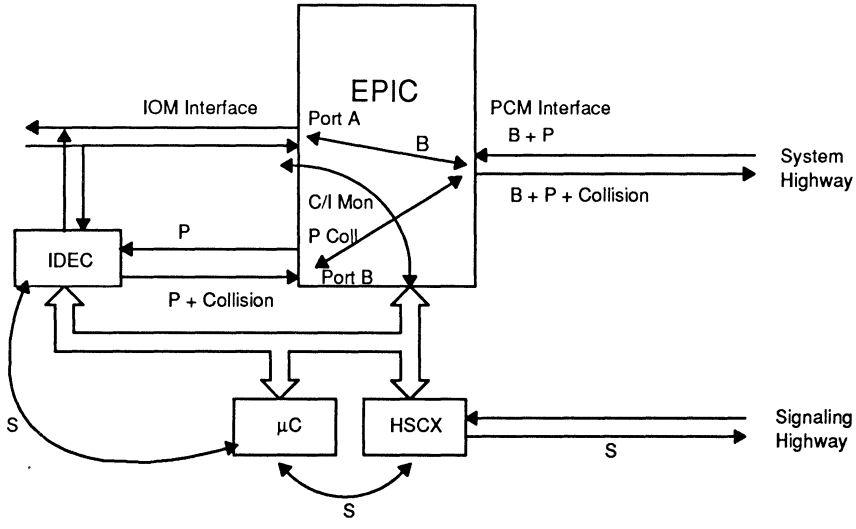


Figure 12. Line Card Architecture for Mixed Packet Handling Two IOM Ports of the EPIC

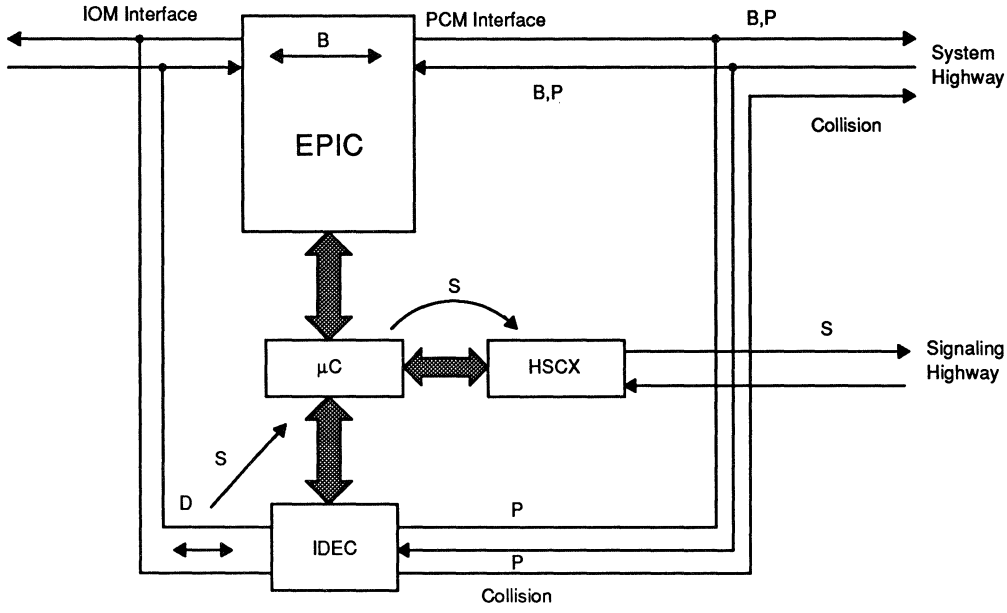


Figure 13. Digital Line Card Architecture for Mixed Packet Handling Using a Collision Highway

also connected to this internal highway as illustrated in Figure 14.

This figure shows one EPIC connecting four PCM highways to one packet handler internal highway. These highways are accessed by the IDECs (ISDN D-Channel Exchange Controller) which are four-channel HDLC controllers that handle the packets. If more than four PCM highways are connected to the centralized packet

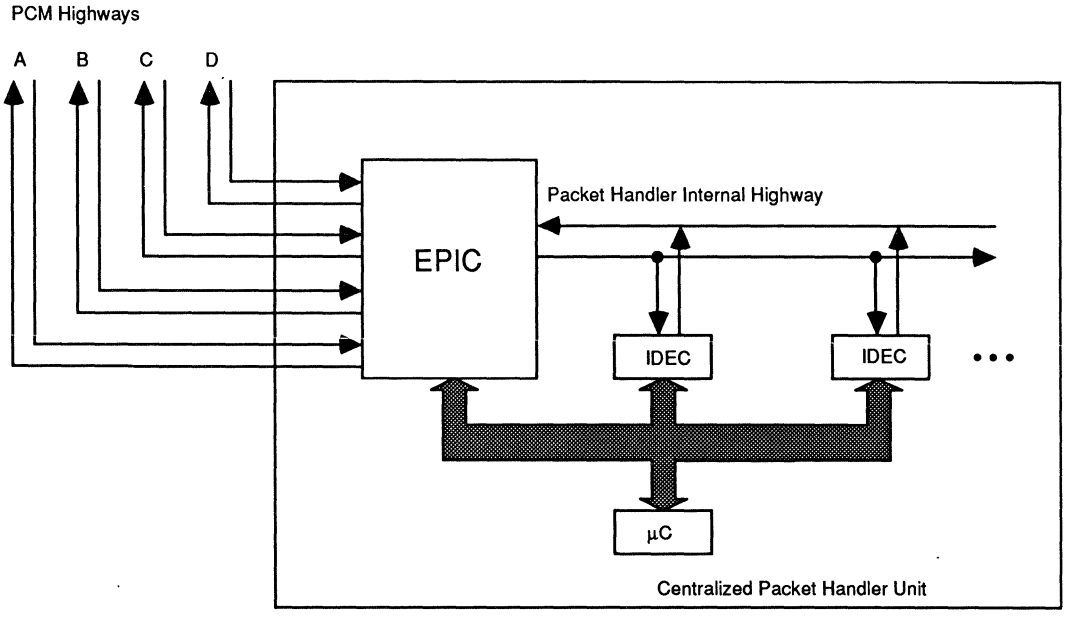


Figure 14. Centralized Packet Handler with a Single Internal Highway Connected to Four PCM Highways

handler, further EPICs are necessary. Such a situation is shown in Figure 15, where eight highways are switched to one packet handler internal highway. In this case, the two EPICs are connected in parallel at the packet handler internal side.

The data rate of the packet handler internal highway can be up to 4096 kb/s. If this capacity is not sufficient, other packet handler internal highways may be added as shown in Figure 16.

In some applications an additional collision resolution signal is required for the HDLC controllers. This information can be demultiplexed from the PCM highways to a third line for each packet handler internal highway (see Figure 17).

The applications illustrated apply equally to centralized signaling as well as to data packet handlers.

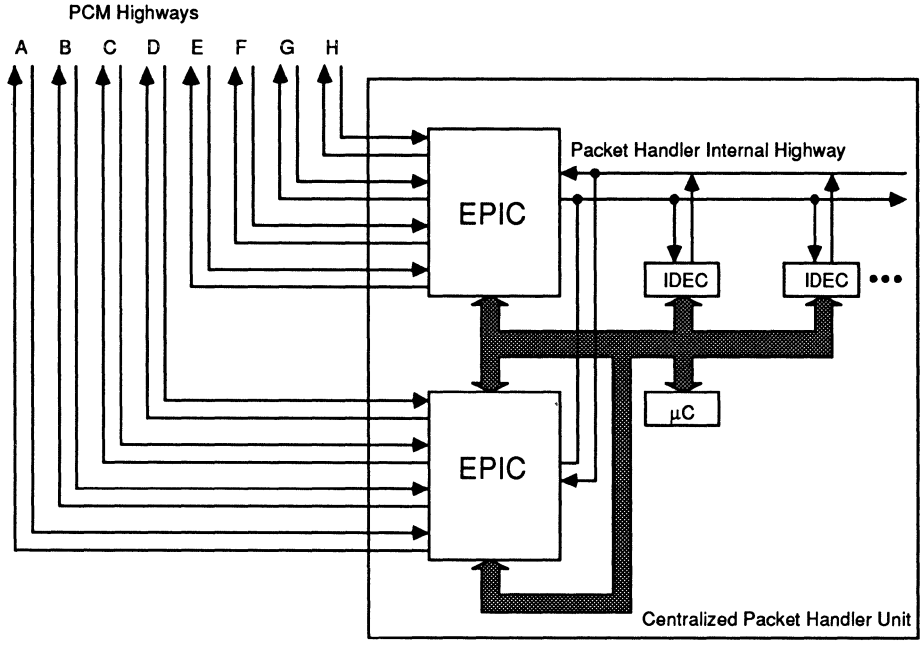


Figure 15. Centralized Packet Handler with One Internal Highway Connected to Eight PCM Highways

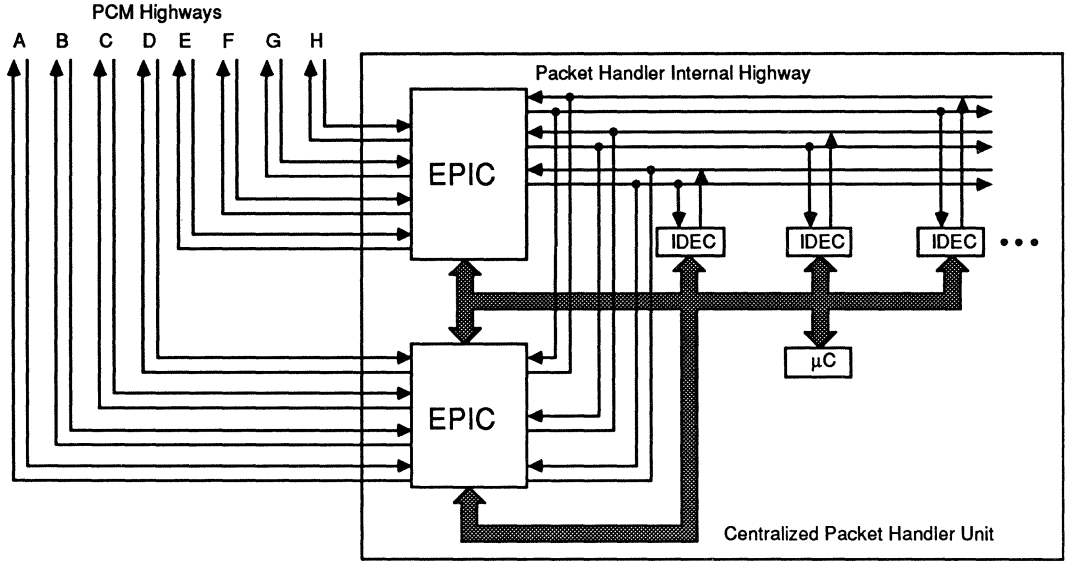


Figure 16. Centralized Packet Handler with Three Internal Highways

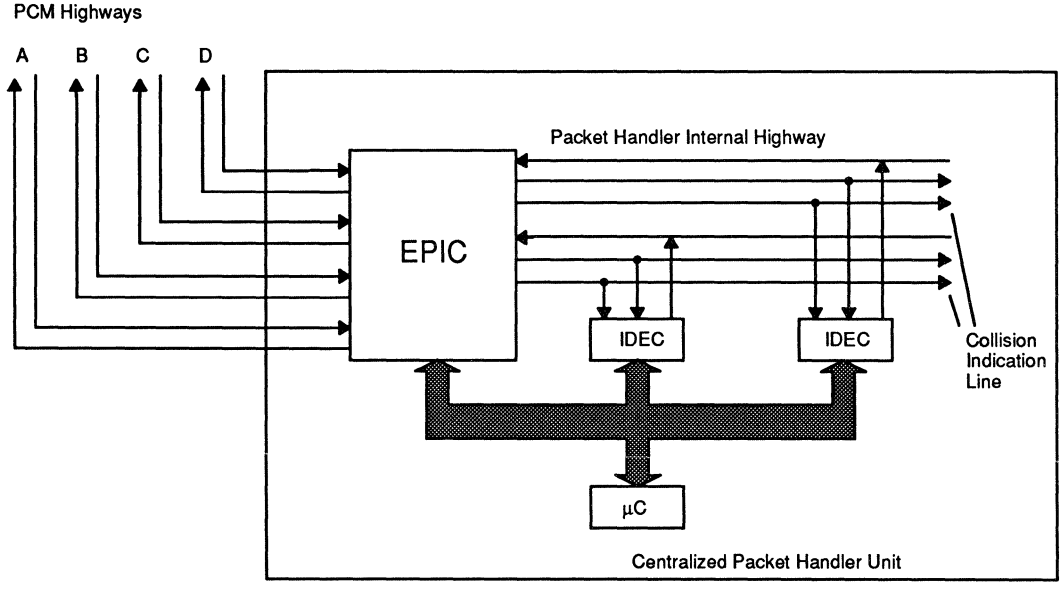


Figure 17. Centralized Packet Handler with Internal Collision line

ABSOLUTE MAXIMUM RATINGS

Ambient Temperature under bias	0 to 70°C
Storage Temperature	–65 to 125°C

Stresses above those listed may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods of time may affect device reliability.

OPERATING RANGES

Ambient Temperature (T_A)	0 to +70°C
Supply Voltage with respect to V_{SS}	V_{DD} = 5 V ±5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{IL}	Input LOW Voltage		–0.4	0.8	V
V_{IH}	Input HIGH Voltage		2.0	V_{DD}+ 0.4	V
V_{OL}	Output LOW Voltage	I_{OL} = 2 mA		0.45	V
V_{OH}	Output HIGH Voltage	I_{OH} = –400 µA	2.4		V
V_{OH}	Output HIGH Voltage	I_{OH} = –100 µA	V_{DD}–0.5		V
I_{CC}	Operational Power Supply Current	V_{DD} = 5 V, Inputs at 0 V or V_{DD}, no output loads.			mA
		clock frequency > 4096 kHz		9.5	
		clock frequency ≤ 4096 kHz		6.5	
I_{LI}	Input Leakage Current	0 V < V_{IN} < V_{DD} to 0 V		10	µA
I_{LO}	Output Leakage Current	0 V < V_{OUT} < V_{DD} to 0 V		10	µA

CAPACITANCE

T_A = 25°C; V_{DD} = 5 V ±5%, V_{SS} = 0 V.

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
C_{IN}	Input Capacitance			10	pF
C_{IH}	I/O Capacitance			20	pF
C_{OUT}	Output Capacitance			15	pF

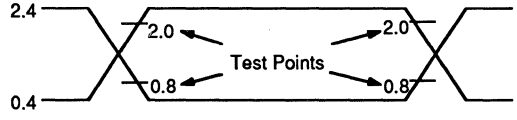
SWITCHING CHARACTERISTICS over operating range

Timing of Microprocessor Interface

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
t_{AA}	ALE pulse width		30		nsec
t_{AL}	Address setup time to ALE		10		nsec
t_{LA}	Address hold time from ALE		20		nsec
t_{ALS}	Address latch setup time to $\overline{WR}$, $\overline{RD}$		0		nsec
t_{AS}	Address setup time to $\overline{WR}$, $\overline{RD}$		10		nsec
t_{AH}	Address hold time from $\overline{WR}$, $\overline{RD}$		25		nsec
t_{DSD}	$\overline{RD}$ delay after $\overline{WR}$ setup		0		nsec
t_{RR}	$\overline{RD}$ pulse width		120		nsec
t_{RD}	Data output delay from $\overline{RD}$			100	nsec
t_{DF}	Data float from $\overline{RD}$			25	nsec
t_{RI}	$\overline{RD}$ control interval		70		nsec
t_{WW}	$\overline{WR}$ pulse width		60		nsec
t_{DW}	Data setup time to $\overline{WR}$+$\overline{CS}$		30		nsec
t_{WD}	Data hold time from $\overline{WR}$+$\overline{CS}$		10		nsec
t_{WI}	$\overline{WR}$ control interval		70		nsec

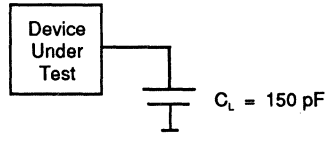
Inputs are driven at 2.4 V for a logical "1" and at 0.4 for a logical "0." Timing measurements are made at 2.0 V for a logical "1" and at 0.8 V for a logical "0." The AC testing input/output waveforms are shown below.

SWITCHING TEST WAVEFORM



I/O Waveform for AC Tests

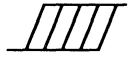
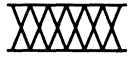
SWITCHING TEST CIRCUIT



Switching Test Load Circuit

SWITCHING WAVEFORMS

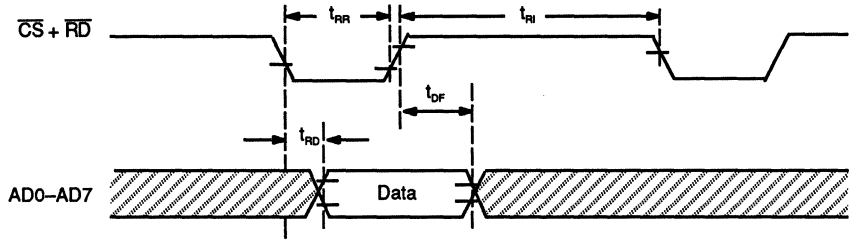
KEY TO SWITCHING WAVEFORMS

Waveform	Inputs	Outputs
———	Must Be Steady	Will Be Steady
———		
	May Change From H to L	Will Be Changing From H to L
	May Change From L to H	Will Be Changing From L to H
	Don't Care, Any Change Permitted	Changing, State Unknown

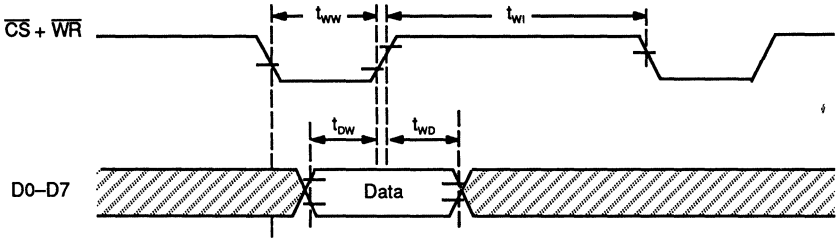
KS000010

SWITCHING WAVEFORMS

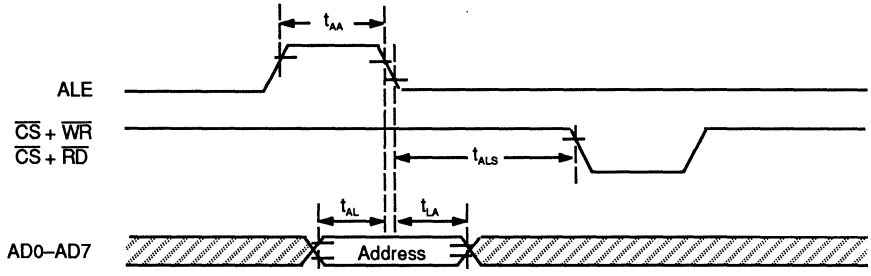
Intel Bus Mode



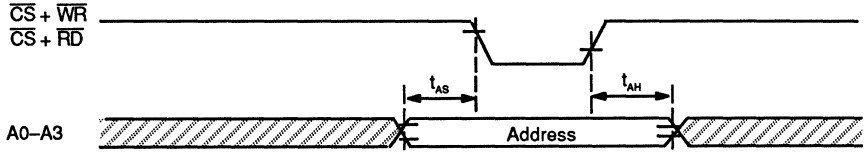
µP Read Cycle



µP Write Cycle



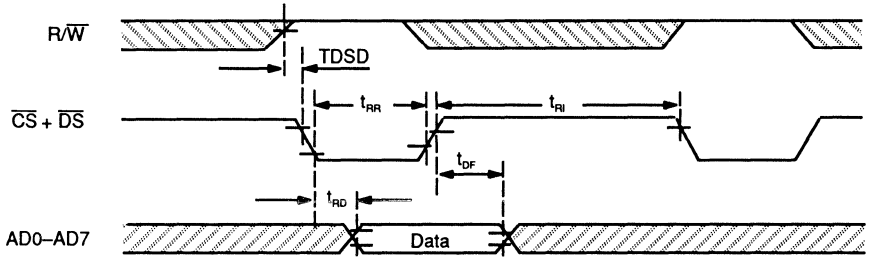
Multiplexed Address Timing



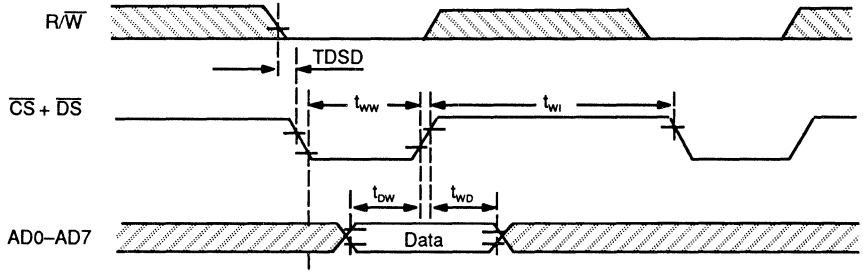
Demultiplexed Address Timing

SWITCHING WAVEFORMS (continued)

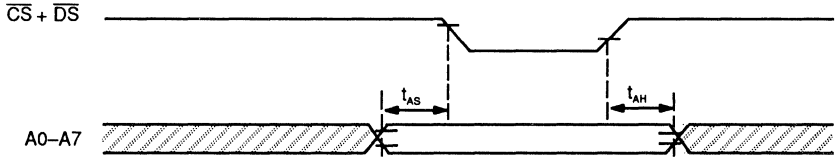
Motorola Bus Mode



µP Read Cycle



µP Write Cycle



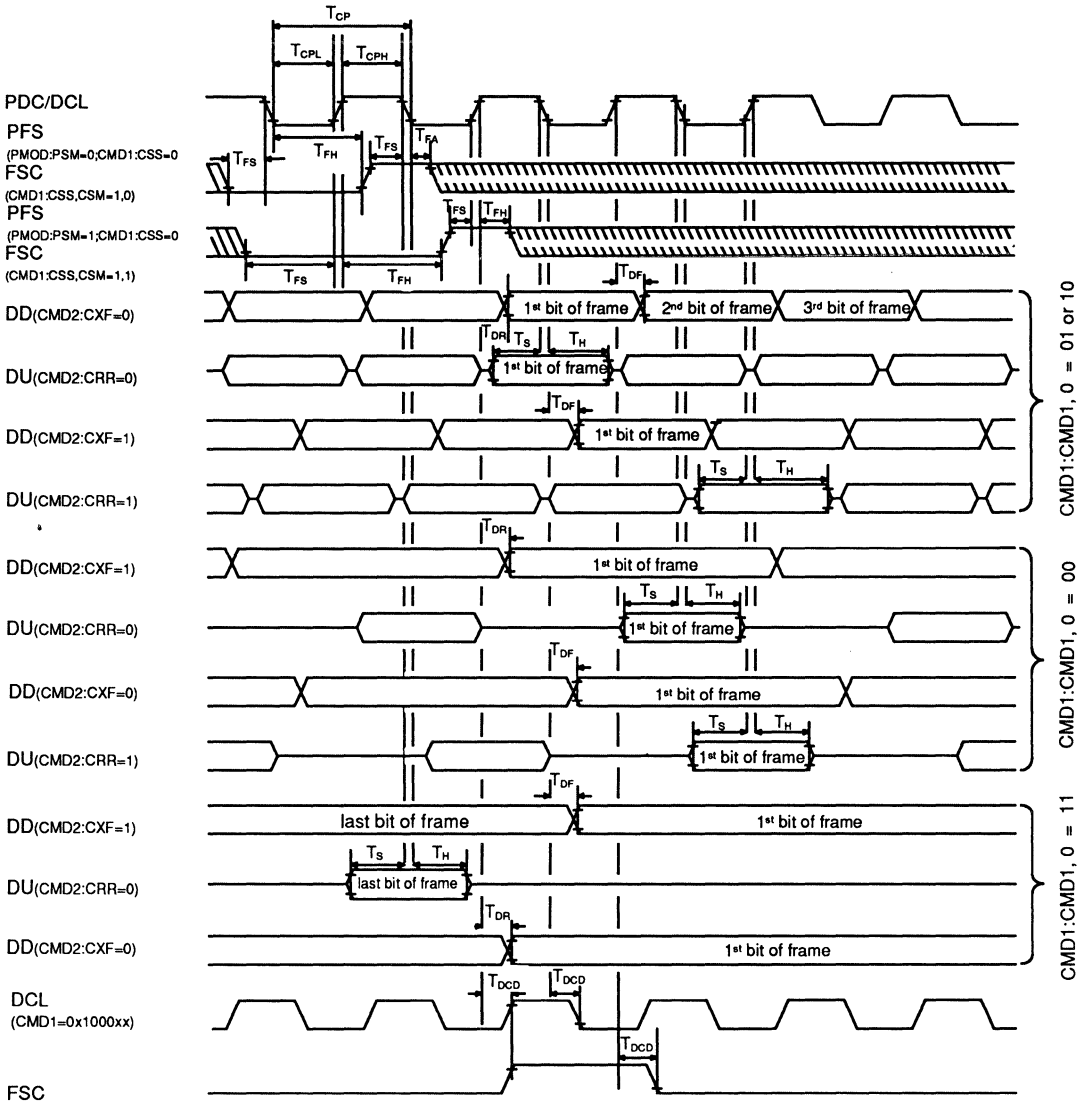
Address Timing

SWITCHING CHARACTERISTICS over operating range (continued)

Timing of PCM and Configurable Interfaces

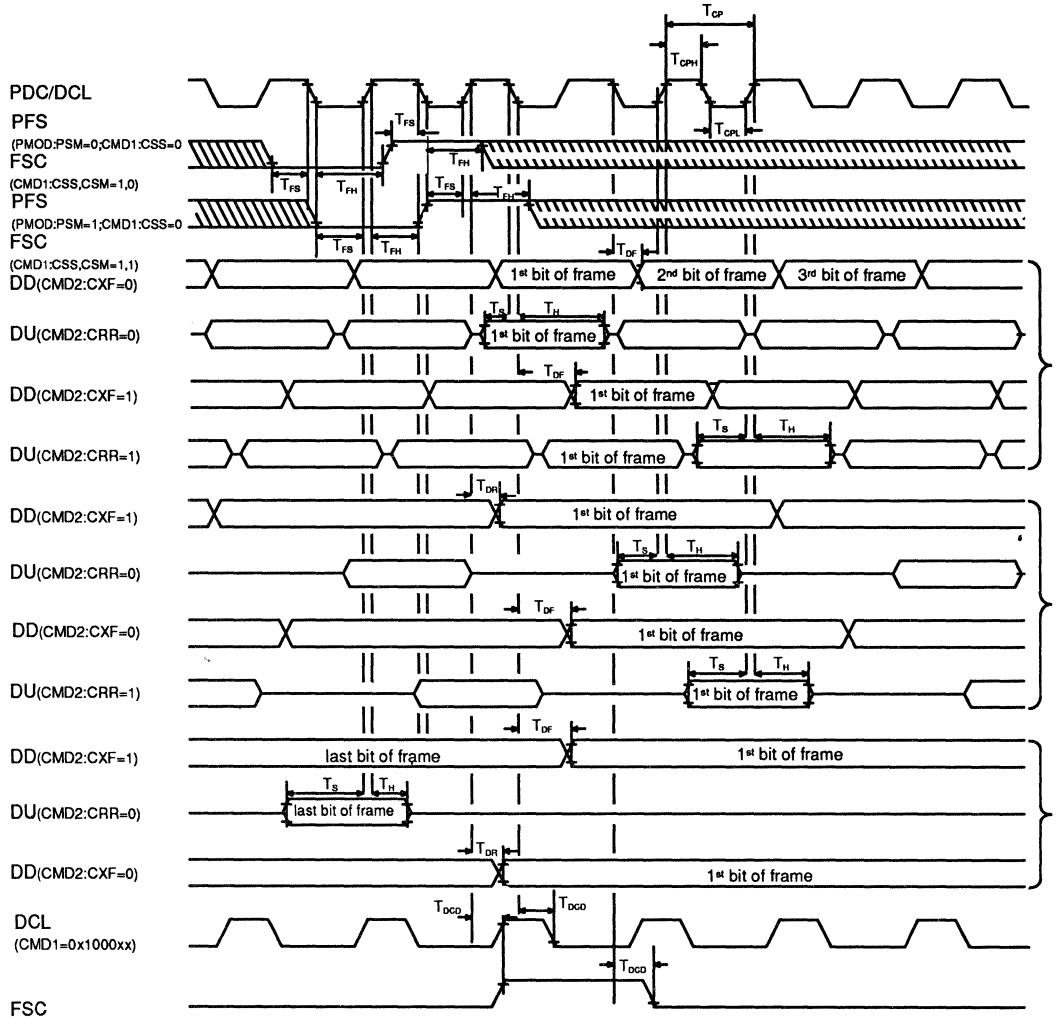
Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
t_{CP}	Clock period	clock frequency ≤ 4096 kHz	240		nsec
t_{CPL}	Clock period LOW	clock frequency ≤ 4096 kHz	80		nsec
t_{CPH}	Clock period HIGH	clock frequency ≤ 4096 kHz	100		nsec
t_{CP}	Clock period	clock frequency > 4096 kHz	120		nsec
t_{CPL}	Clock period LOW	clock frequency > 4096 kHz	50		nsec
t_{CPH}	Clock period HIGH	clock frequency > 4096 kHz	50		nsec
t_{FS}	Frame setup time		15		nsec
t_{FH}	Frame hold time		50		nsec
t_{DCD}	Data clock delay time			125	nsec
t_S	Serial data input setup time	PCM input data frequency > 4096 kb/s	7		nsec
t_H	Serial data input hold time	PCM input data frequency > 4096 kb/s	35		nsec
t_S	Serial data input setup time	PCM input data frequency ≤ 4096 kb/s	15		nsec
t_H	Serial data input hold time	PCM input data frequency ≤ 4096 kb/s	50		nsec
t_S	Serial data input setup time	CFI input data frequency > 4096 kb/s	15		nsec
t_H	Serial data input hold time	CFI input data frequency > 4096 kb/s	50		nsec
t_S	Serial data input setup time	CFI input data frequency ≤ 4096 kb/s	0		nsec
t_H	Serial data input hold time	CFI input data frequency ≤ 4096 kb/s	75		nsec
t_D	PCM Serial data output delay time			55	nsec
t_T	Tristate control delay			60	nsec
t_{DF}	CFI Serial data output delay time (falling clock edge)			60	nsec
t_{DR}	CFI Serial data output delay time (rising clock edge)			80	nsec

SWITCHING WAVEFORMS (continued)



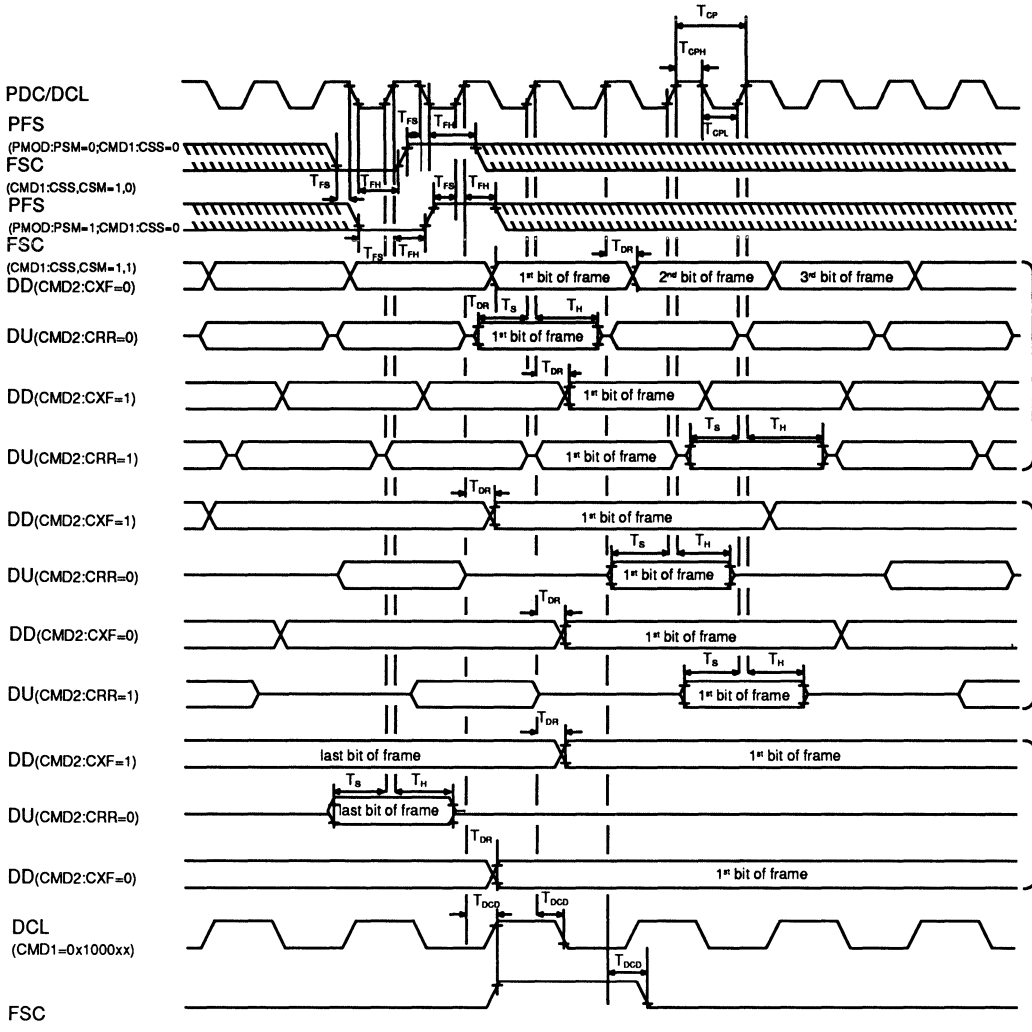
Switching Characteristics at the CFI with CMD: CSP1, 0 = 10 (Prescalar Divisor = 1)

SWITCHING WAVEFORMS (continued)



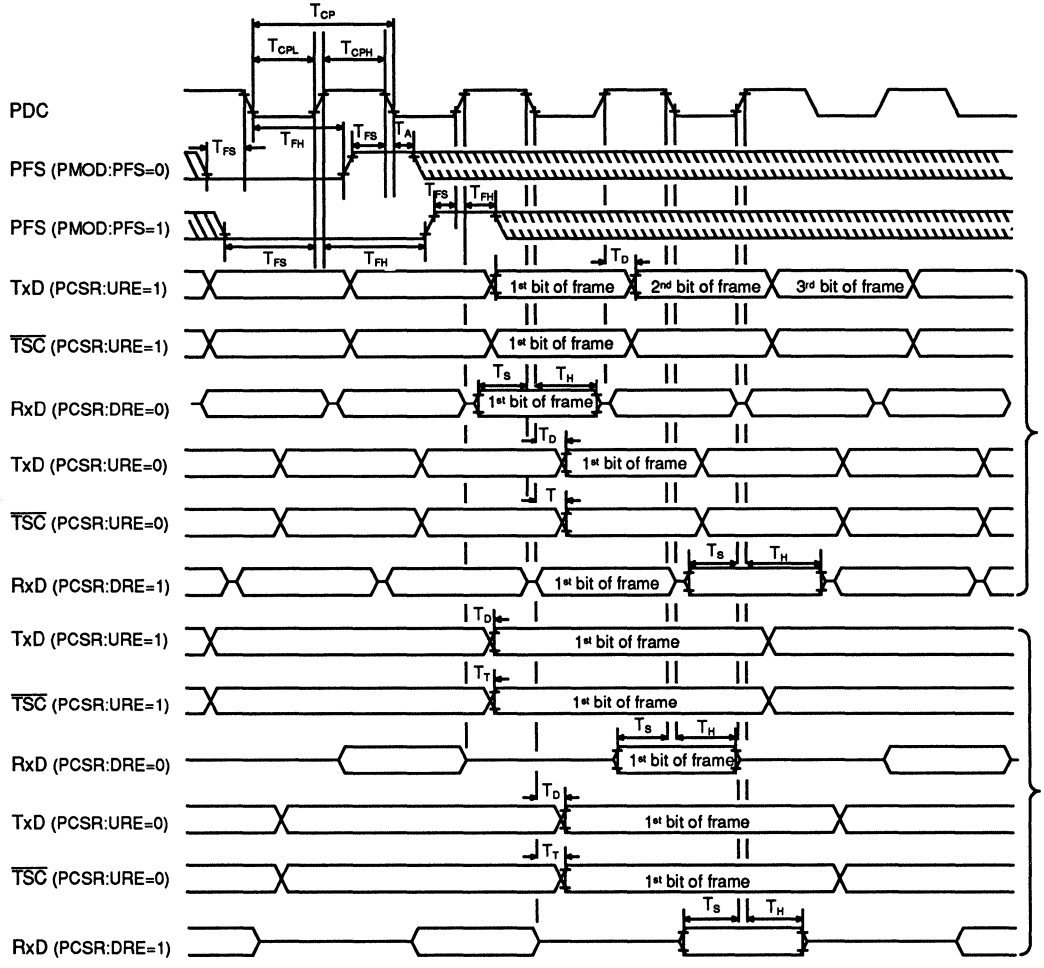
Switching Characteristics at the CFI with CMD1: CSP = 01 (Prescalar Divisor = 1.5)

SWITCHING WAVEFORMS (continued)



Switching Characteristics at the CFI with CMD: CSP1, 0 = 00 (Prescalar Divisor = 2)

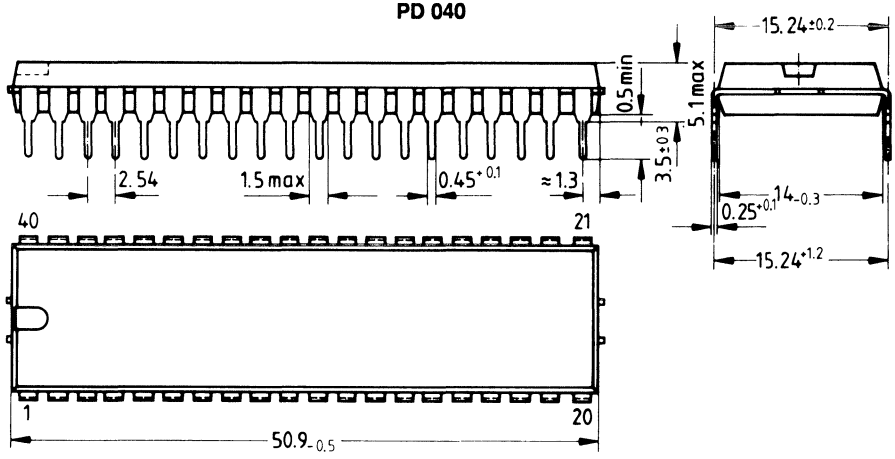
SWITCHING WAVEFORMS (continued)



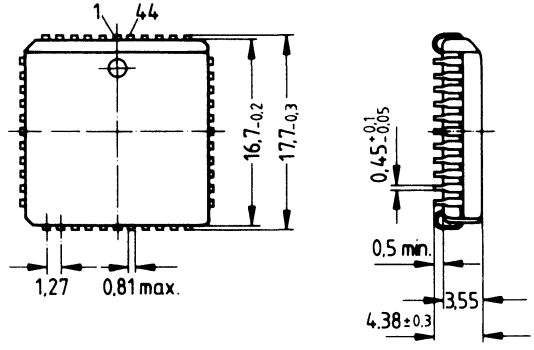
Switching Characteristics at the PCM Interface

PHYSICAL DIMENSIONS

PD 040



PLCC 044



Note: Physical Dimensions are given in mm.

Preliminary

Am2075

ISDN Digital Exchange Controller (IDEC)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Four independent HDLC channels**
- **64 byte FIFO storage per channel and direction**
- **Handling of basic HDLC functions**
  - flag detection/generation
  - zero deletion/insertion
  - CRC checking/generation
  - check for abort
- **Single connection and quad connection modes**
- **IOM™ interface or PCM interface**
- **Programmable time slots and channel data rates (up to 4 Mbs)**
- **Different methods of contention resolution**
- **8-bit parallel microcontroller interface with vectored interrupt**
- **Advanced CMOS technology**
- **Power consumption less than 50 mW**

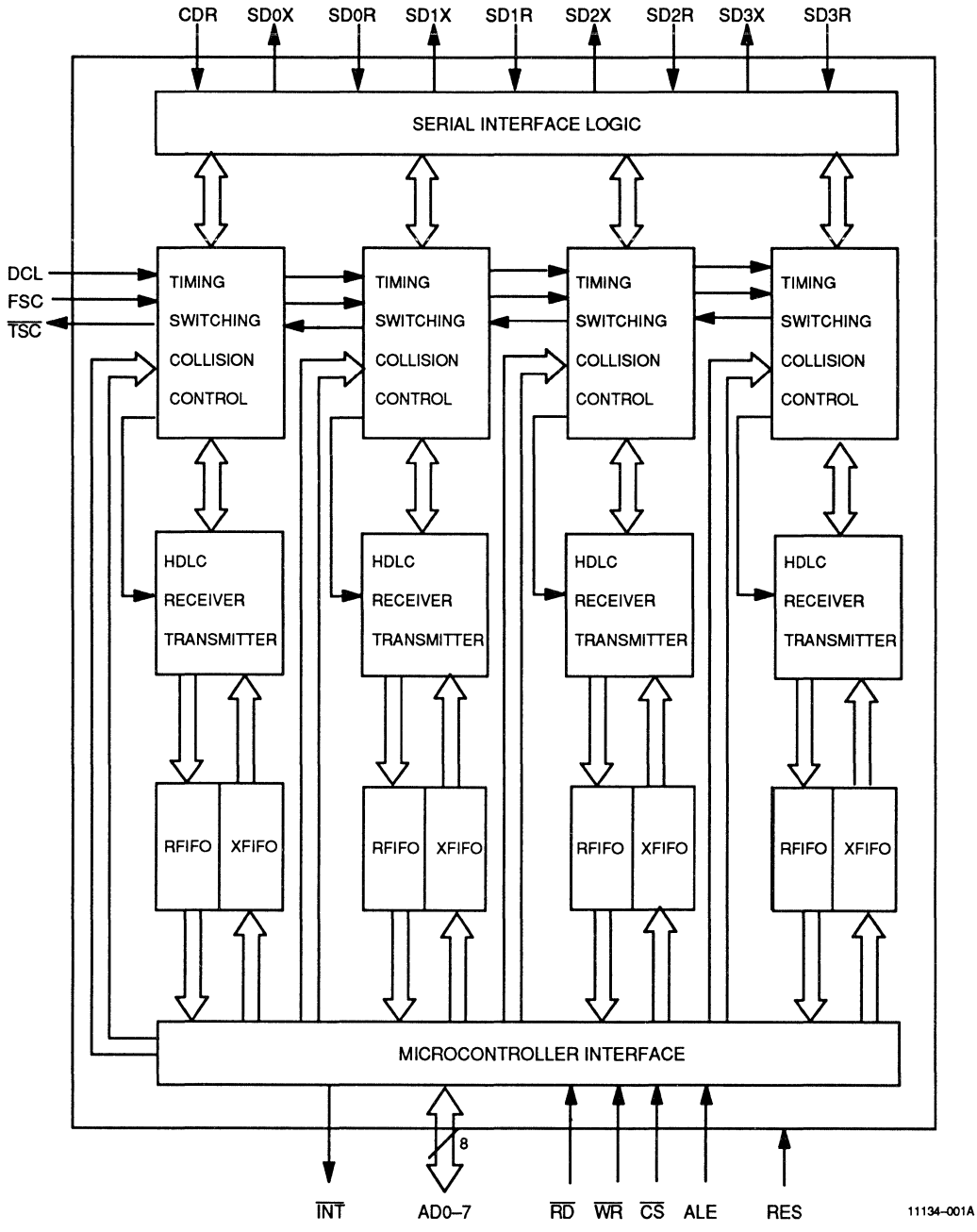
GENERAL DESCRIPTION

The Am2075, ISDN Digital Exchange Controller (IDEC), is a serial HDLC data communication circuit with four independent channels. Its telecommunication-specific features make it especially suited for use in variable data rate PCM systems. In addition, the device contains sophisticated switching functions and implements automatic contention resolution between packet data from different sources.

Its applications include communication multiplexers, peripheral ISDN line cards, packet handlers, and X.25 packet switching devices. The IDEC is a fundamental building block for networks with either centralized, decentralized, or mixed signaling packet data handling architectures.

Publication # 11134 Rev. B Amendment /0
Issue Date: April 1989

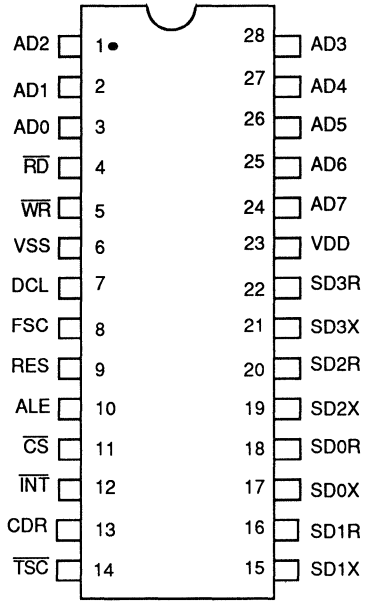
BLOCK DIAGRAM



2

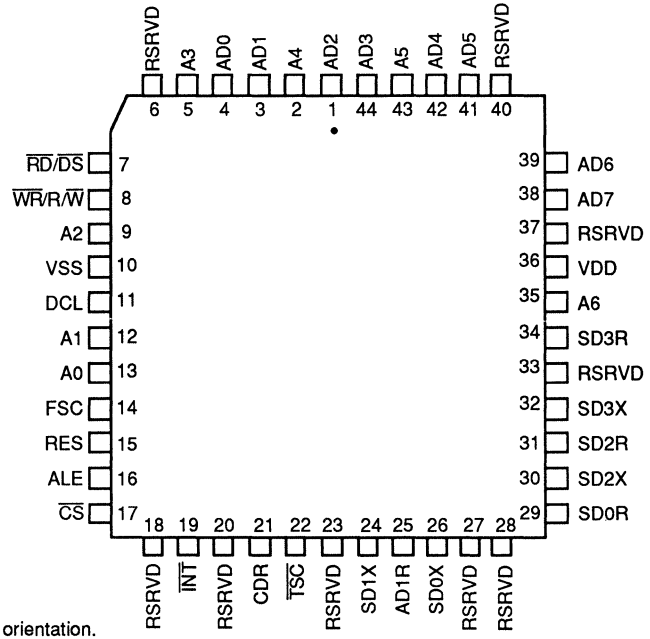
CONNECTION DIAGRAMS
Top View

28-Pin DIP



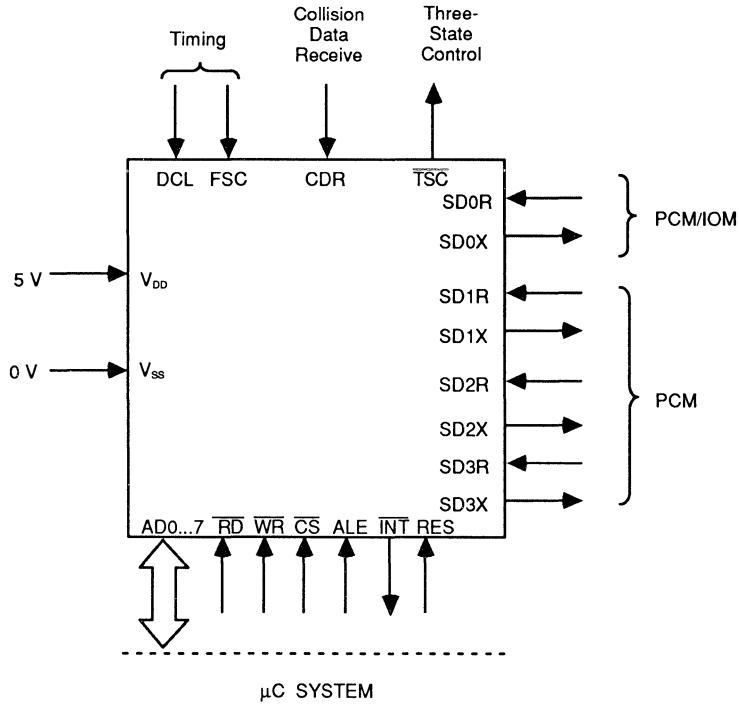
44-Pin PLCC

11134-002B



Note: Pin 1 is marked for orientation.

LOGIC SYMBOL



11134-003B

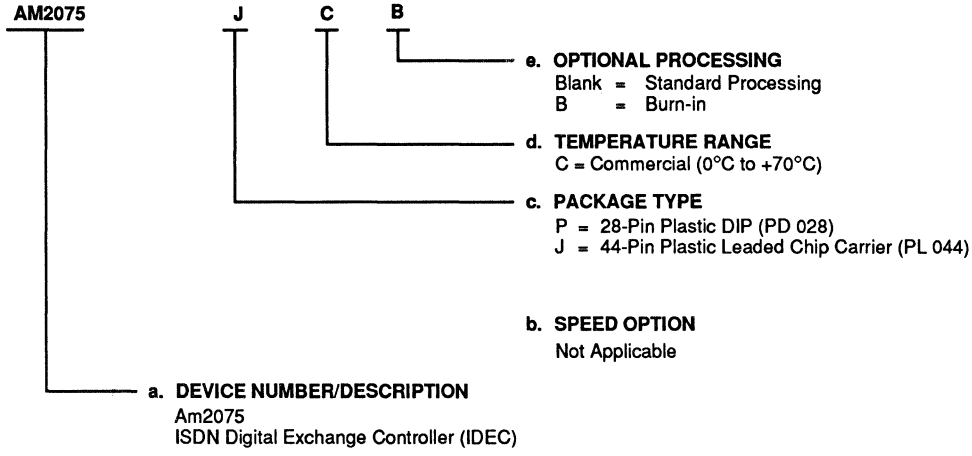
2

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM2075	PC, JC, PCB, JCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

AD0–AD7

Address-Data Bus (Input/Output)

The multiplexed Address Data bus transfers data and commands between the µP system and the IDEC.

$\overline{CS}$

Chip Select (Input)

A LOW on this line selects the IDEC for a read/write operation.

$\overline{WR}$

Write (Input)

A LOW on this line indicates a write operation.

$\overline{RD}$

Read (Input)

A LOW on this line indicates a read operation.

$\overline{INT}$

Interrupt Request (Output)

This line is activated when the IDEC requests an interrupt. It is and open drain output.

ALE

Address Latch Enable (Input)

A HIGH on this line indicates an address on the external address-data bus, selecting one of the internal sources or destinations.

SD0R–SD3R

Serial Data Receive (Input)

SD0X–SD3X

Serial Data Transmit (Output)

SD2X is a collision output in Master Mode.

DCL

Data Clock (Input)

Supplies a clock signal either equal to or twice the data rate.

FSC

Frame Synchronization (Input)

Data strobe signal.

$\overline{TSC}$

Time Slot Control (Output)

Supplies a control signal for an external driver.

CDR

Collision Data Receive (Input)

RES

Reset (Input)

V_{SS}

Ground (Input)

V_{DD}

+5 V Supply Voltage (Input)

FUNCTIONAL DESCRIPTION

General Functions and Device Architecture

The IDEC is an HDLC controller which handles four HDLC communication channels, each channel fully independent and programmable by its own register set. The circuit performs the following functions:

- Extraction (reception) and insertion (transmission) of the HDLC data packets in a time-division multiplex-bit stream.
- Implementation of the basic HDLC functions of the layer 2 protocol.
- Interfacing of the data packets to the microprocessor bus. For the temporary storage of data packets, overlapping FIFO structures are used per channel and direction.
- Switching of data between serial interfaces.
- Implementation of different types of collision resolution.
- Test functions.

Operating Modes

Each HDLC controller of the IDEC is assigned to one time channel governed either by time slot assignment or by an external strobe signal. Two basic configurations are distinguished (Figure 1).

- In the Quad connection configuration, the four HDLC controllers (A–D) are connected to individual time-multiplexed communication lines.

In the Quad connection configuration, two modes are distinguished as follows:

- Each connection is a time-slotted highway; the lengths and positions of the time slots are programmable (quad-connection time slot mode).
- Each connection is a communication line; the time channels are marked by an external strobe signal (quad-connection common control mode).
- In the Single connection configuration, the four HDLC channels are all connected to one time-multiplexed communication line.

Two modes are distinguished in turn for the single connection configuration as follows:

- The connection is a standard IOM interface with predefined channel positions (single connection IOM mode).
- The connection is a time-slotted highway (single-connection time slot mode).

For simplicity, a time-slotted highway will sometimes be referred to as a "PCM highway," or PCM for short.

Table 1. Four Basic Operating Modes of the IDEC

MDS1	MDS0	Mode Description
0	0	Single connection time slot mode
0	1	Quad connection common control mode
1	0	Single connection IOM mode
1	1	Quad connection time slot mode

The four modes of operation are illustrated in Figure 2. Using channel-by-channel programming, one of a number of collision detection modes may be selected in each of the basic modes of operation. For future reference, they are also depicted in Figure 2.

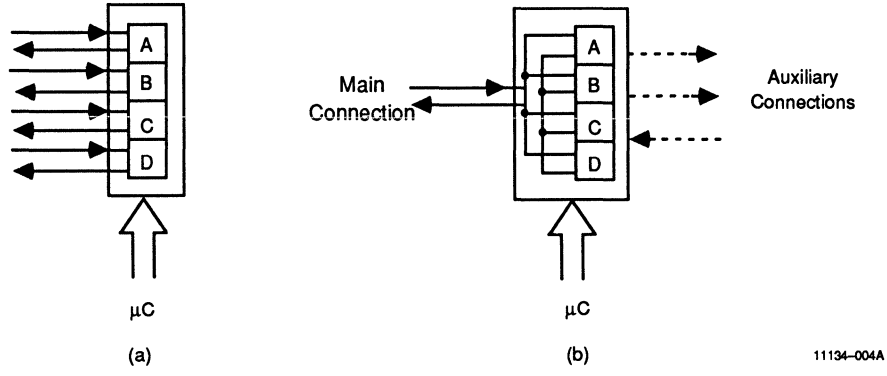
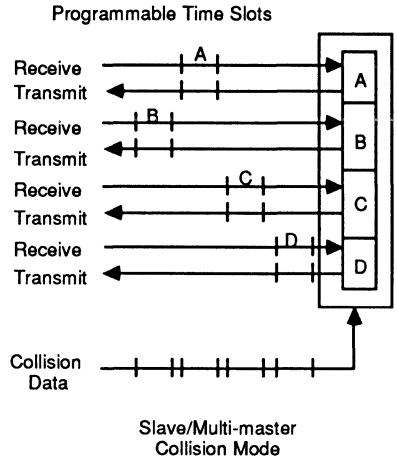
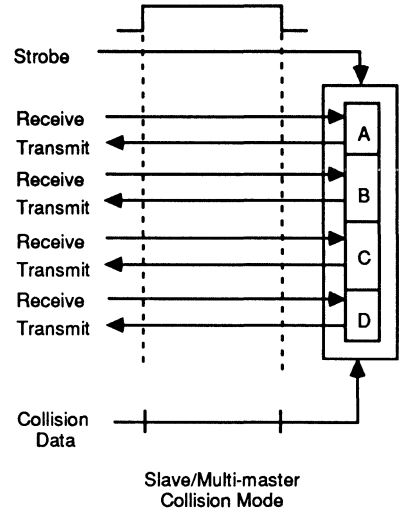


Figure 1. (a) Quad Connection; (b) Single Connection Configurations

a. Quad Connection TS Mode

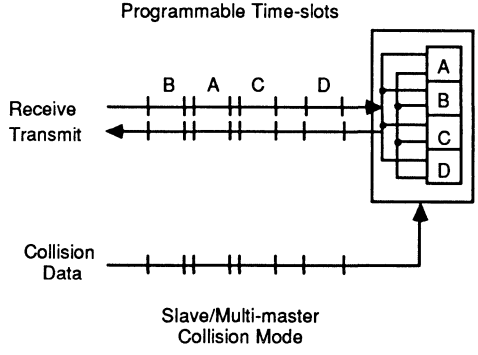


b. Quad Connection Common Control Mode

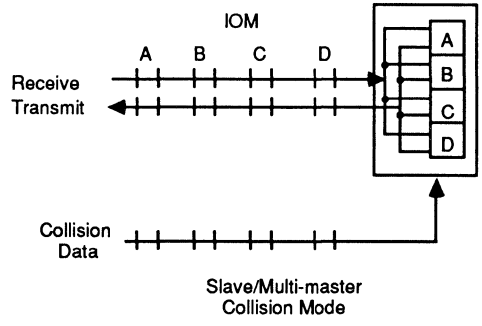


2

c. Single Connection TS Mode



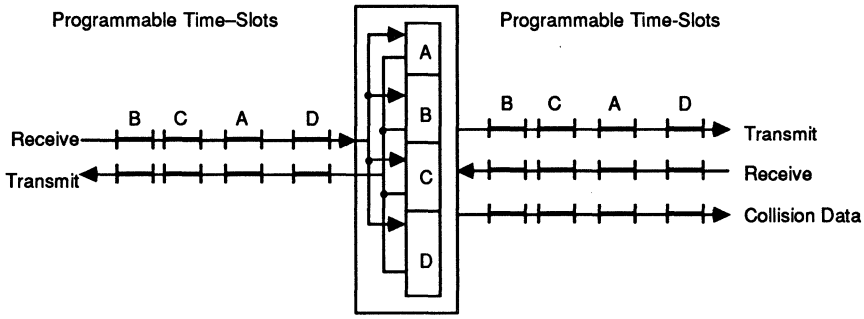
d. Single Connection IOM Mode



11134-005₁A

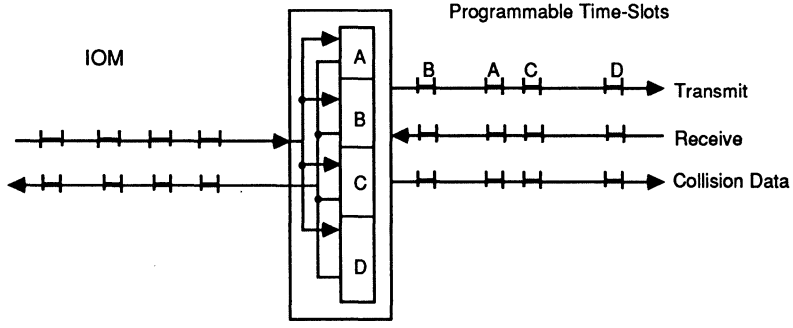
Figure 2. Operating Modes of the IDEC

e. Single Connection TS Mode



Master Collision Mode

f. Single Connection IOM Mode



Master Collision Mode

11134-005$_2$A

Figure 2. Operating Modes of the IDEC (continued)

INTERFACES

Microcontroller Interface

The IDEC is programmable over an 8-bit parallel microcontroller interface. Easy and fast microprocessor access is provided by 8-bit address decoding the on chip. The interface consists of 13 lines and is directly compatible with processors of the multiplexed address/data bus type.

In addition to 8-bit processors, the IDEC supports a direct connection to 16-bit processors. Thus, through an internal address transformation, it is possible to access all IDEC registers using either even microprocessor addresses only or odd microprocessor addresses only.

Serial Interfaces

Depending on the selected mode, the IDEC supports four physically separate, full-duplex serial interfaces, or one full-duplex serial interface.

In addition to the data input and data output lines, the serial interface requires a common data clock (input DCL) and a frame synchronization signal (input FSC). Input data is latched on the falling edge of DCL and output data is clocked on the rising edge of DCL. The IDEC may be programmed so that the data clock rate is either equal to the data rate, or twice the data rate.

Table 2. Microcontroller Interface Signals of the IDEC

Symbol	Type	Name and Function
AD0 AD1 AD2 AD3 AD4 AD5 AD6 AD7	I/O I/O I/O I/O I/O I/O I/O I/O	Address-Data bus. The multiplexed address/data bus transfers data and commands between the µC system and the IDEC.
$\overline{CS}$	I	Chip Select. A LOW on this signal selects the IDEC for a read/write operation.
$\overline{WR}$	I	Write. This signal indicates a write operation.
$\overline{RD}$	I	Read. This signal indicates a read operation.
$\overline{INT}$	OD	Interrupt Request. The signal is activated when the IDEC requests an interrupt. It is an open drain output.
ALE	I	Address Latch Enable. A HIGH on this line indicates an address on the external address/data bus.

INDIVIDUAL FUNCTIONS

Channel Access

The four HDLC controllers of the IDEC are connected to the serial interfaces as shown in Table 3. The table indicates the selection of the data channel, the selectable time-slot widths, the output driver type, and the function of the active-LOW Tri-State Control ($\overline{TSC}$) output in each of the operating modes.

The data output is set in a high impedance state outside the time channel where data is transmitted.

Quad connection time slot mode. Channel selection is performed via the Time Slot Select registers (TSR). For each HDLC channel, the 8-bit TSR register gives the position of a time slot with a 2-bit resolution. The length of the time slot, either 1, 2, 7 or 8 bits, can be selected using the MODE register (CCS1, 0). These parameters are common to the receive and the transmit channel. In the case where the number of bits in a PCM frame is 256 or 512, the frame synchronization signal FSC need not be provided at every PCM frame beginning because bit counters are automatically reset at frame end. When the PCM frame length is not equal to either 256 or 512 bits, the frame synchronization signal has to be provided at the beginning of every PCM frame.

The tri-state control output line $\overline{TSC}$ marks the time slot when data is transmitted/received by the HDLC controller B.

The position of a time slot with respect to FSC, as a function of the TSR register contents, is shown in Figure 3.

Quad connection common control mode. Channel selection is performed by an active HIGH strobe signal provided through the FSC input. The strobe signal is common to all four HDLC channels. The $\overline{TSC}$ output is active when the FSC strobe is active.

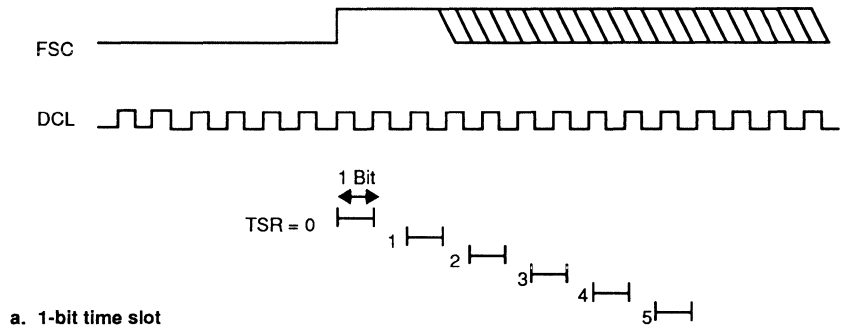
Single connection TS mode. The time slots selected by the TSR registers all pertain to the same PCM highway. The programming of a channel otherwise proceeds exactly as explained above. The tri-state control output line $\overline{TSC}$ marks the time slots when data is transmitted/received by any of the four controllers.

Single connection IOM mode. The IOM is an interface where a frame is composed of n IOM channels ($n \geq 1$; $n = 8$ in Figure 4a). Each IOM channel has a unique structure. It consists of two 8-bit bytes, corresponding to the ISDN B-channels, a monitor byte, and a control byte of which the first two bits are allocated to the ISDN D-channel. In the single connection IOM mode, the serial interface has an IOM frame structure and the four HDLC channels are assigned to the D bits of four consecutive IOM channels. The choice whether the four HDLC controllers are assigned to IOM channels 0–3 or 4–7 is governed by the microcontroller bit VIS (Common Configuration Register). See Figure 4b.

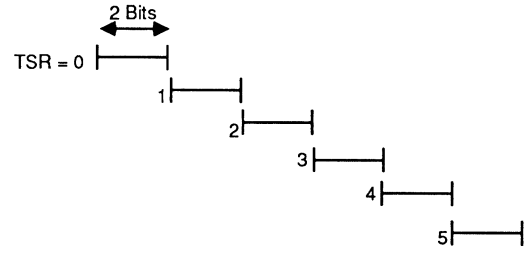
Table 3. HDLC Controller Channel Selection

Mode		Channel Input				Channel Output				Output	Channel	Tri-State Channel	Control	
MDS1	MDS0	A	B	C	D	A	B	C	D	Driver	Select	Width	$\pi(\overline{TSC})$	Description
0	0	SD0R	SD0R	SD0R	SD0R	SD0X	SD0X	SD0X	SD0X	PP or 0D	TSR A–D registers	1,2,7,8	TSR A-D	Single connection TS mode
0	0	SD0R	SD1R	SD2R	SD3R	SD0X	SD1X	SD2X	SD3X	PP or 0D	FSC strobe	Arbitrary	FSC inverted	Quad connection mode
1	0	SD0R	SD0R	SD0R	SD0R	SD0X	SD0X	SD0X	SD0X	OD	Fixed 2-bit TSs	2	Fixed 2-bit TSs A–D	Single connection IOM mode
1	1	SD0R	SD1R	SD2R	SD3R	SD0X	SD1X	SD2X	SD3X	PP or OD	TSR A–D registers	1,2,7,8	TSR B	Quad connection TS mode

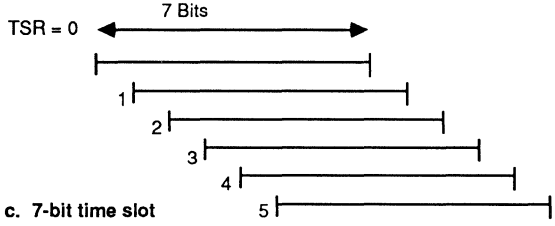
OD = Open-drain driver
PP = Push-pull driver
The output driver type refers to the SD0X (or SD0X, SD1X, SD2X, SD3X) outputs.
$\overline{TSC}$ is a push-pull signal.



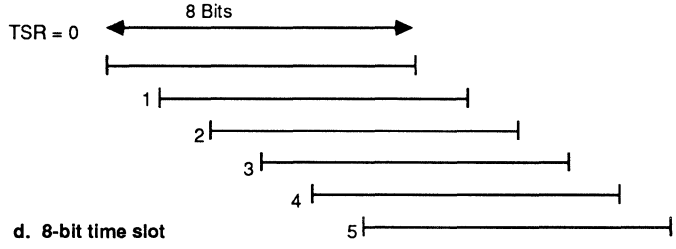
a. 1-bit time slot



b. 2-bit time slot



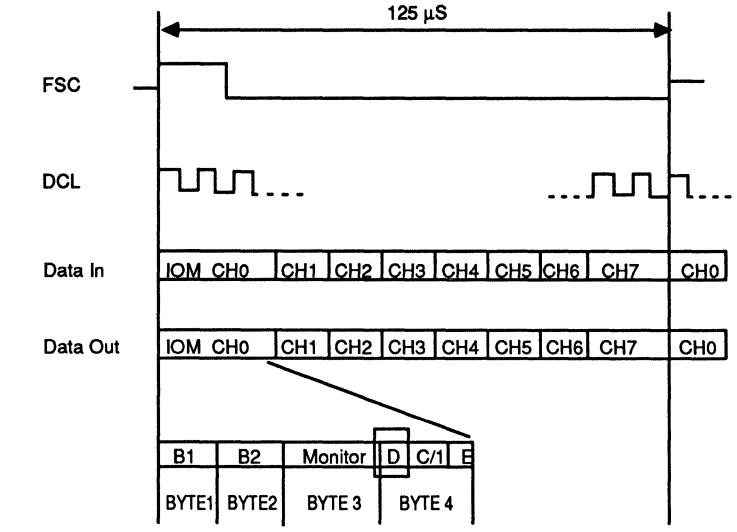
c. 7-bit time slot



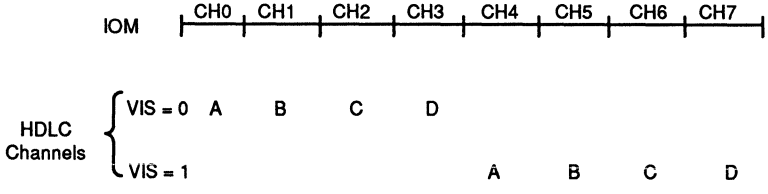
d. 8-bit time slot

11134-006A

Figure 3. Position of Time Slot for Different Channel Widths as a Function of TSR Register Contents



(a)



(b)

11134-007A

Figure 4. a) Multiplexed Frame Structure of the IOM Interface (n = 8)
b) Assignment of HDLC Channels in IOM Mode

HDLC Communication Functions

Basic HDLC Functions

Each of the four controller channels handles the following basic HDLC functions.

Receive direction:

- Flag detection

  A zero followed by six consecutive ones and another zero is recognized as a flag.

- Zero delete

  A zero after five consecutive ones within an HDLC frame is deleted.

- CRC checking

  The CRC field of an HDLC frame is checked according to the generator polynomial $X^{16} + X^{12} + X^5 + 1$.

- Check for abort

  Seven or more consecutive ones are interpreted as an abort sequence.

- Check for idle

  Fifteen or more consecutive ones are interpreted as "idle," and reported to the processor via a status bit.

- Minimum length checking

  Reception of frames with less than three bytes between opening and closing flag is not reported to the microcontroller.

Transmit direction:

- Flag generation

  A flag is generated at the beginning and at the end of every frame.

- Zero insert

  A zero is inserted after five consecutive ones within an HDLC frame.

- CRC generation

  The CRC field of the transmitted frame is generated according to the generator polynomial $X^{16} + X^{12} + X^5 + 1$.

- Abort sequence generation

  An HDLC frame may be terminated with an abort sequence under software control or due to a FIFO underrun condition.

- Inter-frame time fill

  As inter-frame time fill, either flags or idle (continuous ones) may be transmitted.

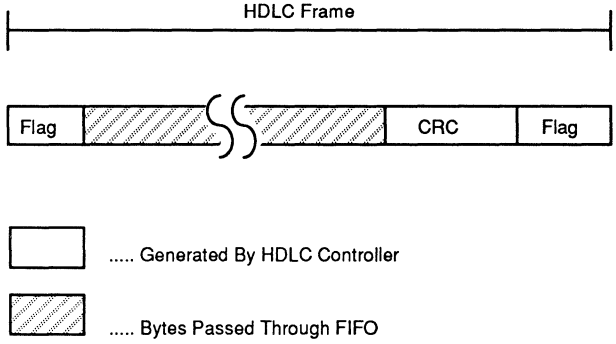


Figure 5. HDLC Frame Structure

Reception and Transmission Functions

FIFO Structure

Each HDLC controller uses a 64-byte FIFO per direction for the intermediate storage of data packets. All data bytes between the opening flag and the CRC field of an HDLC frame are passed through the FIFO.

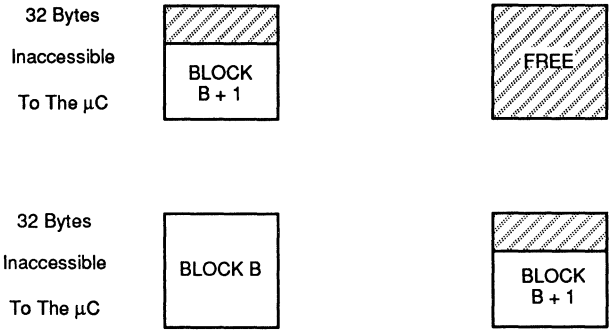
The Receive and Transmit FIFOs are both divided in two blocks of 32 bytes each; one accessible to the microcontroller and one inaccessible to the microcontroller. While the microcontroller is reading (Receive FIFO) or writing (Transmit FIFO) data in one 32-byte block, the other block is filled (Receive FIFO) or emptied (Transmit FIFO) by the IDEC. Therefore, the length of the received or transmitted frame is not limited by the FIFO size.

Reception of Frames

In the case of a frame up to 64 bytes long, the whole frame may be stored in the Receive FIFO. After the first 32 bytes have been received, the device prompts the microcontroller to read data from the FIFO. Having done this, the microcontroller releases the FIFO. This is effected by a software command, after which the rest of the frame, when ready, is made available to the microcontroller (Figure 6).

In the case of frames longer than 64 bytes, the microcontroller will repeatedly be prompted by interrupt to read out the FIFO by blocks of 32 bytes (except possibly the end of the frame if the total length is not a multiple of 32 bytes). Again, after reading a block, the microcontroller acknowledges the data by a software command and thus releases the FIFO. If this is not done before an additional 32 data bytes are received, the next data byte will lead to a "data overflow" condition.

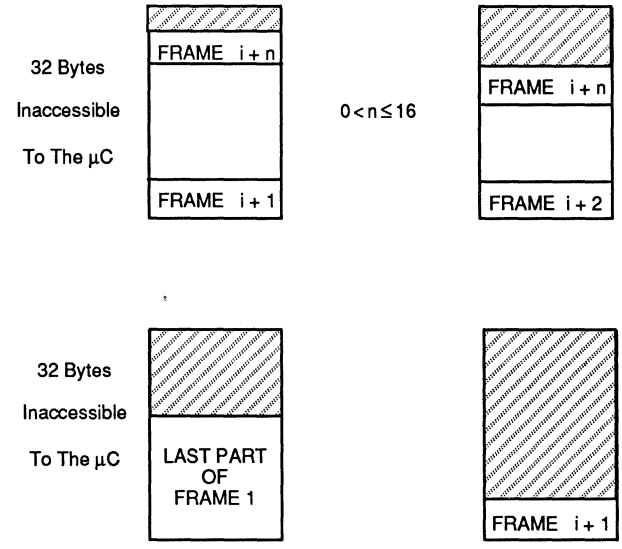
In the case of several shorter frames, up to seventeen may be stored inside the HDLC controller. After an interrupt, one frame is available in the FIFO for the microcontroller to read. Up to sixteen other frames may be stored in the meanwhile in the upper half of the FIFO (Figure 7a). When the microcontroller releases the current data block from the FIFO by a software command, the next frame becomes available and the corresponding space is freed in the upper half for a subsequent frame (Figure 7b).



a. Prior to µC Acknowledgment **b. After µC Acknowledgment**

11134-009A

Figure 6. Receive FIFO in the Case of a Frame no Longer than 64 Bytes



a. Prior to µC Acknowledgment **b. After µC Acknowledgment**

11134-010A

Figure 7. Receive FIFO in the Case of Short Frames

The interrupts accumulating in the process are incorporated into a queue and transferred one by one to the microcontroller along with additional information about the frame. In particular, the frame length is stored in a register. Information such as "frame aborted yes/no," "CRC error yes/no," "data overflow yes/no," is included in an extra byte in the FIFO.

Every interrupt has to be acknowledged by the microcontroller. A full FIFO at the beginning of a frame will lead to a frame overflow condition.

If the microcontroller does not wish to preserve an incoming frame, the possibility exists to ignore it. When the corresponding command is issued, the part of the frame stored is deleted and the rest of the entire frame will be ignored.

Transmission of Frames

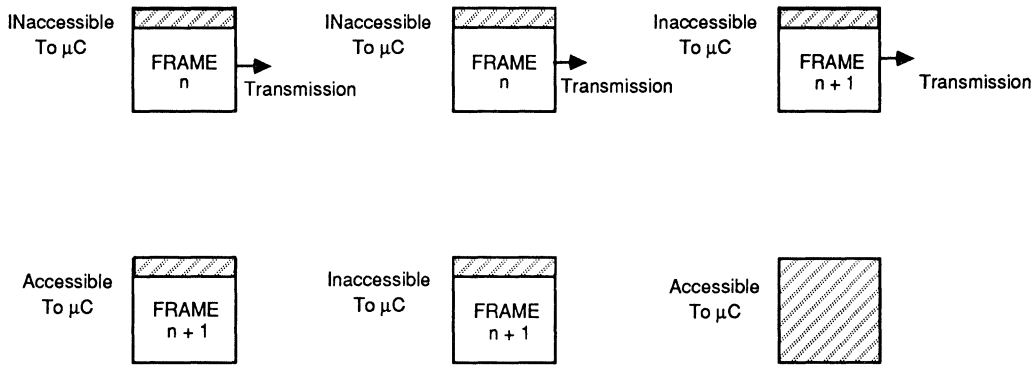
2 • 32 bytes of intermediate storage are provided per HDLC controller in the transmit direction. After up to 32 bytes have been written to the FIFO, transmission is started by a software command. If the previous transmission is still underway when a new transmission command is issued, microcontroller access to the FIFO will be blocked until the first transmission is completed (Figure 8). This means that at most one complete frame may be written to the FIFO before a transmission has to be initiated. If a transmission request does not include a "frame end" indicator, the HDLC controller will request the next data block via an interrupt if the FIFO contains not more than 32 bytes. This procedure will be repeated until the microcontroller indicates that the frame is to be closed.

In the case when this indication is not given and there is no more data ready for transmission, the frame is terminated with an abort sequence and the microcontroller is notified via an interrupt. The frame may also be aborted per software command. The completed transmission of an HDLC frame is reported by interrupt.

The IDEC possesses flexible collision control capabilities which are totally transparent to the microcontroller. The collision control modes allow using the circuit in statistical multiplexing applications or in centralized or decentralized packet switches. Each of the four HDLC controllers is individually programmed in one of four modes by its own register bits CMS1–0 (Collision Mode Select).

Table 4 lists the four collision modes that can be selected, along with the auxiliary I/O lines used in each case. The outputs SD1X and SD2X can be selected to be of the open-drain or of the push-pull type.

2



a. Prior to transmission command for frame n + 1

b. After transmission command for frame n + 1

c. After interrupt "Transmit pool ready"

11134-011A

Figure 8. Transmit FIFO

Table 4. Collision Modes of the IDEC

CMS1	CMS0	Description	Auxiliary I/O			
			Data In	Data Out	Coll. In	Coll Out
0	0	Unconditional transmission				
0	1	Slave mode			CDR	
1	0	Multi-master			CDR	
1	1	Master mode	CDR	SD1X		SD2X

Unconditional Transmission Mode

The HDLC controller transmits frames without collision detection on the transmit line (time channel).

Slave Mode

The input CDR (Collision Data Receive) is used to control transmission of frames. This input is common to all HDLC controllers which are programmed in the slave mode.

Transmission is inhibited by a LOW on the CDR input. If CDR becomes LOW during the transmission of a frame, the frame is aborted by the HDLC controller, and the data output is set to high impedance. (Refer to Figure 9.)

The state of CDR is evaluated by the HDLC controller only in the time channel used for transmission by that controller.

When CDR is switched HIGH, interframe time fill is marked in the transmit-time channel if no transmission request is pending; otherwise, transmission starts at the first available instant. Transmission of a previously aborted frame is automatically restarted by the HDLC controller if the beginning of the frame is still available in the transmit FIFO. Otherwise, an interrupt to the microcontroller indicates that the transmission has failed.

The slave mode is applicable in all of the basic operation modes, in both single connection and in quad connection applications. However, there is only one CDR line. This could be a restriction in the following cases:

- The IDEC is configured in the quad connection common control mode and more than one HDLC controller is operated in the slave mode;
- When a time slot is used by more than one HDLC controller in the slave mode.

In both cases, more than one controller is evaluating the CDR line during the same time interval, and when CDR goes LOW, they all stop transmitting.

Multi-Master Mode

In the multi-master mode the controllers perform a bus access procedure and collision detection in their assigned time channel(s). As a result, any number of IDECs can be assigned to one physical channel, where they perform statistical multiplexing.

Collisions are detected by automatic comparison of each transmitted bit with the bit received over the CDR input. For this purpose, a logical "and" of the bits transmitted by parallel controllers is formed and connected to the input CDR. This may be implemented most simply by defining the output line driver to be of the open drain type (ODS = 1). Consequently, the logical "and" of the outputs is formed by simply tying them together ("wired or"). The result is returned to the CDR input of all parallel circuits.

The multi-master mode is applicable in all operating modes, in both single connection and quad connection applications. In the quad connection mode, those output lines (SD0X. . .SD3X) for which this collision mode is selected may be connected to CDR. The four HDLC controllers may either be programmed to transmit in separate time channels or in the same time channel. A prerequisite for the multi-master mode is that the interframe time fill used is "idle."

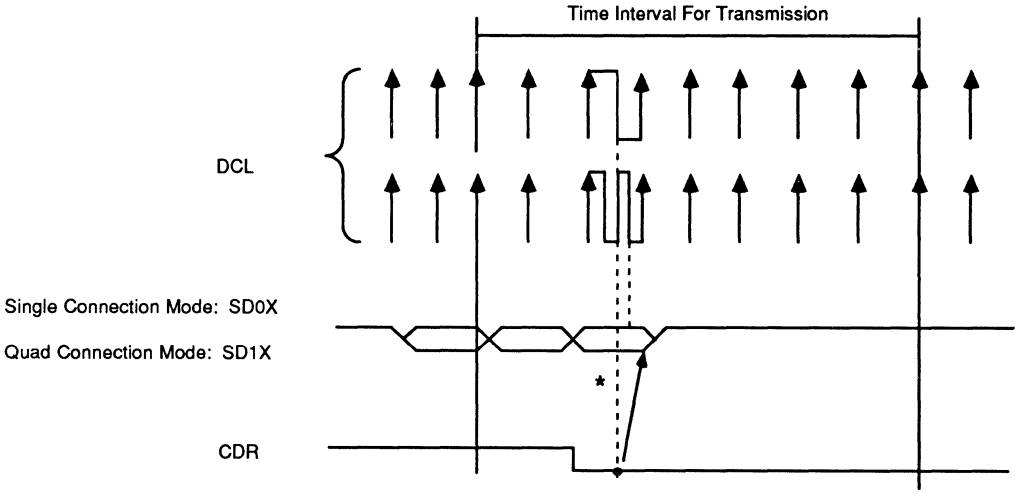
When a mismatch between a transmitted bit and the bit on CDR is detected, the HDLC controller stops sending further data and its output is set to high impedance. (Refer to Figure 10.) As soon as it detects the transmit bus to be "idle" again, the controller automatically attempts to re-transmit its frame. By definition, the bus is assumed idle when x consecutive ones are detected in the transmit channel. Normally x is equal to 8.

An automatic priority adjustment is implemented in the multi-master mode. Thus, when a complete frame is successfully transmitted, x is increased to 10, and its value is restored to 8 when a row of ten "1"s is detected on the bus. Furthermore, transmission of a new frame may be started by the HDLC controller after the tenth "1". This multi-master, deterministic-priority management ensures an equal right of access of every HDLC controller to the transmission medium, thereby avoiding blocking situations.

Master Mode

The master mode requires three auxiliary connections; data input CDR, data output SD1X, and collision data out SD2X. This mode is applicable only in single connection operation. In the master mode (refer to Figure 11), the controller performs two functions:

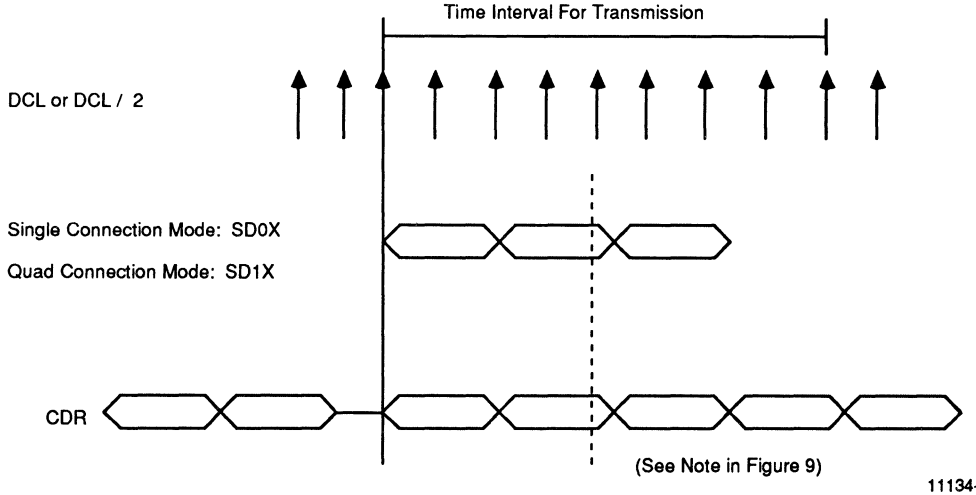
- Switching of data packets between the main connection SD0X, SD0R and the auxiliary input and output (CDR, SD1X)
- Resolution of collisions between data from the auxiliary connection and HDLC frames from the local microcontroller



Note: The CDR input is evaluated:
- at the falling edge of DCL, for a DCL rate equal to the data rate
- at the falling edge of DCL immediately preceding the rising edge, used for transmission for a DCL rate twice the data rate

11134-012A

Figure 9. Transmission Control in the Slave Mode (Example)



11134-013A

Figure 10. Collision Detection in the Multi-Master Mode (Example)

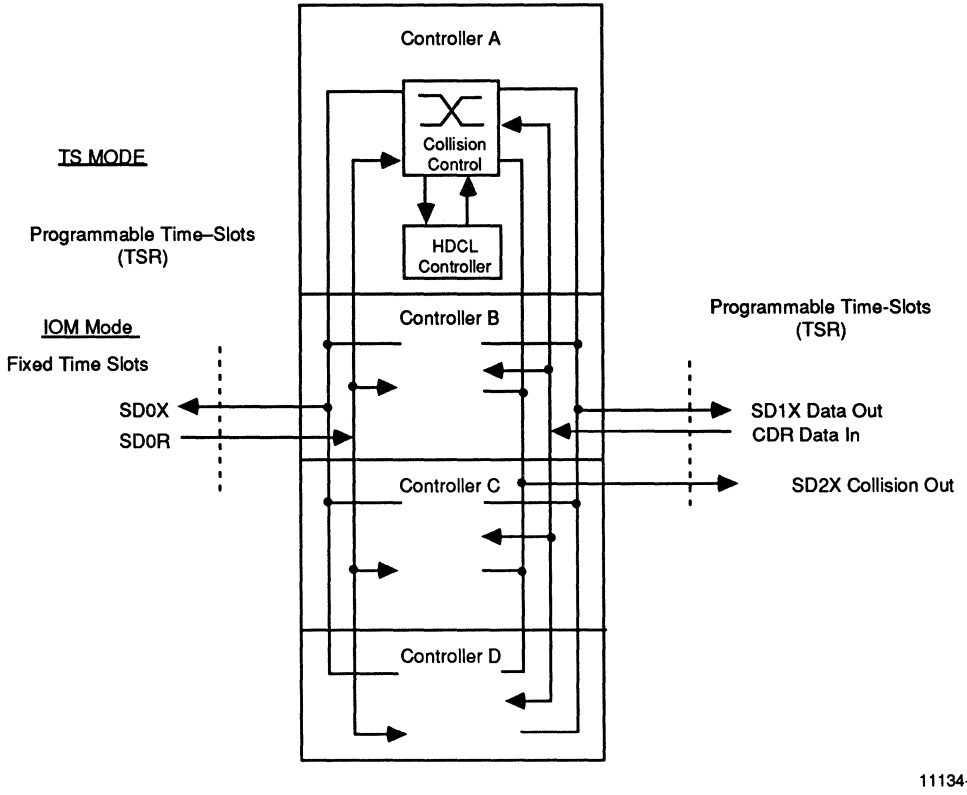


Figure 11. I/O Connections in the Master Mode

In the TS mode, the time slot programmed via the Time Slot Select Register (TSR) applies simultaneously to SD0X/SD0R and to the auxiliary lines CDR, SD1X, SD2X. In the IOM mode, the TSR register selects a time channel on the auxiliary connections CDR, SD1X and SD2X only (however, the channel width selected should be two bits, as on the IOM interface, to ensure a correct data throughput).

The switching of data from SD0R to SD1X is transparent. The switching of data from CDR to SD0X depends on the state of the HDLC controller (transmit/no transmit) and on selected priorities, as follows.

When no transmission command is issued to the HDLC controller, data is transparently switched through from CDR to SD0X. When a transmit request is issued, but the Force HDLC Frame (FHF) bit is not set to 1, the data currently being received (if any) on CDR is given priority. The HDLC controller starts transmitting its frame on SD0X only after CDR is detected to be idle; in other words, when a row of eight "1"s is observed on CDR. Simultaneously, SD2X is set LOW to indicate that no data will be accepted on CDR input data line.

Figure 12a shows the time relation between CDR (data in) and SD2X (collision out) as well as the logical relation between SD2X and SD0X (data out). The figures are simplified in that the grouping of bits into time slots on SD0X, and on SD2X/CDR is not depicted.

When a transmit command is issued and the Force HDLC Frame (FHF) bit is set to "1", the frame currently being received on CDR is aborted. Seven "1"s are appended to the last bit of the aborted frame on SD0X, after which the HDLC controller starts transmitting its frame (Figure 12b).

In both cases, SD2X is set HIGH again after a delay of eight bit-times following the last "0" of the closing flag, to indicate that data is accepted on the CDR input data line. However, if a new transmit command is issued before that time, SD2X remains LOW and transmission of the new frame starts immediately after the eighth "1".

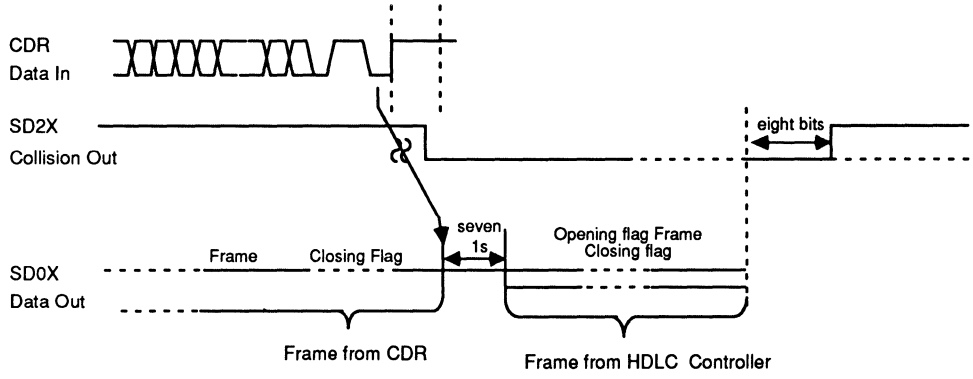
Note on data delay in Master Mode. The data bits are switched from SD0R to SD1X and from CDR to SD0X with a minimum delay as shown in Figure 13.

Two cases are distinguished:

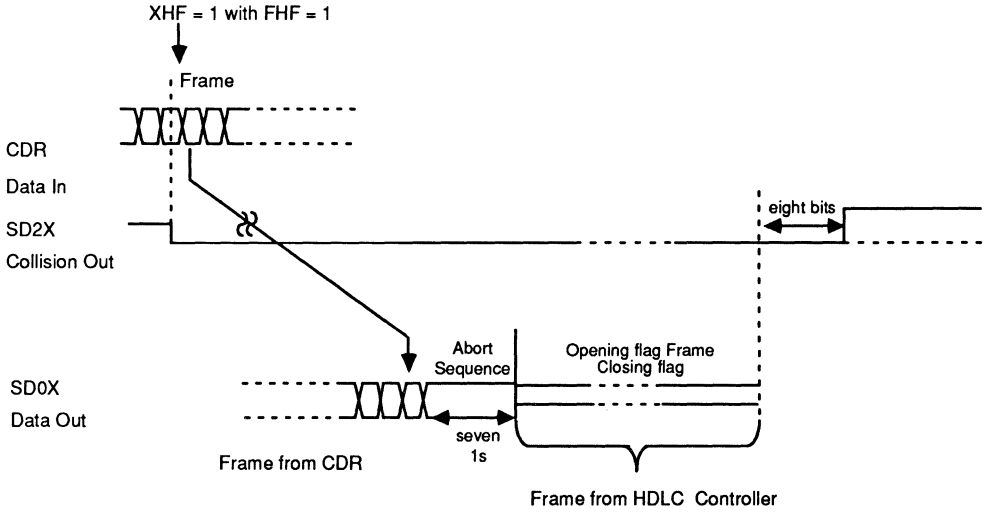
a. TS mode.
In this case the time slots on SD0R/SD0X and on CDR/SD1X are identical. The data delay from CDR to SD0X is one bit, whereas the delay from SD0R to SD1X is two bit times.

b. IOM mode with identical channel (time slot) on SD0R/SD0X and CDR/SD1X. This case is identical to the previous one.

c. IOM mode with a time slot on CDR/SD1X which does not coincide with the IOM channel bits on SD0R/SD0X. In this case, the data bits undergo (in addition to the inherent delay due to the different bit positions) a delay of one bit time from CDR to SD0X, whereas no additional bit delay is introduced when going from SD0R to SD1X.



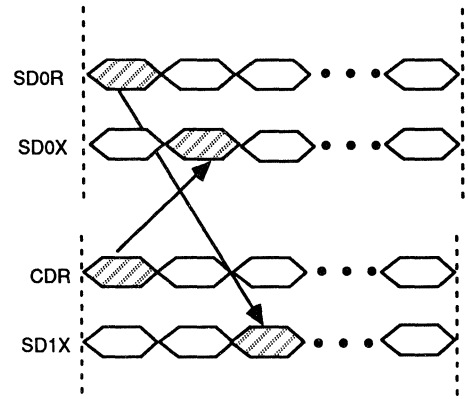
a. XHF Command with FHF = 0



b. XHF command with FHF = 1

11134-015A

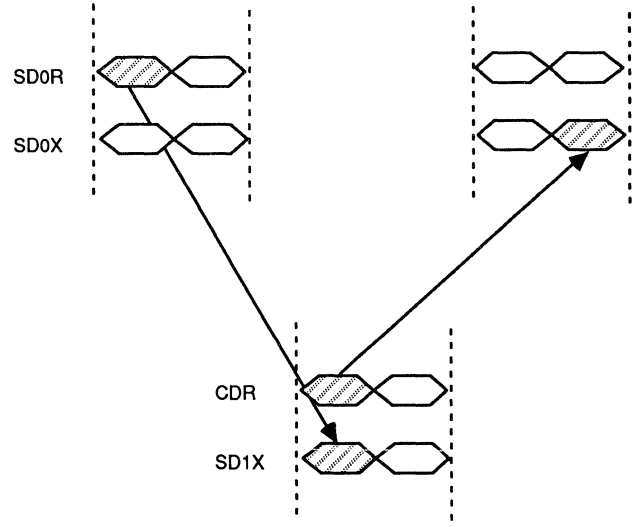
Figure 12. Collision Resolution in the Master Mode with Programmable Priority (FHF)



a. Bit delay for coinciding channel/time slot position on SD0R/SD0X and on CDR/SD1X

11134-016A

Figure 13. Bit Delay from SD0R/CDR to SD1X/SD0X (Minimum Delay)



b. Bit delay for non-identical channel/time slot position on SD0R/SD0X and on CDR/SD1X (possible only when SD0R/SD0X is an IOM interface)

11134-017A

Figure 14. Bit Delay from SD0R/CDR to SD1X/SD0X

Test Functions

A test loop is provided in each of the four HDLC controllers of the IDEC. When the test loop is activated, the input and the output of the HDLC channel are connected together. The test loop control is independent for each HDLC channel (bit TLP).

The test loop is either transparent (forward data is outputted on the line) or non-transparent (forward data is not outputted on the line), depending on the selected model. In the quad connection common control mode and in the single connection IOM mode, the loops are transparent. In the other cases they are non-transparent. During a non-transparent loop, the data output is high impedance inside the assigned time channel.

Communication multiplexers. The four independent serial HDLC communication channels implemented in the IDEC make the circuit suitable for use in communication multiplexers. The collision detection/resolution capability of the circuit allows statistical multiplexing of packets in one or several physical data communication channels; for example, in DMI (mode 3) applications.

Centralized signaling/data packet handlers. The IDEC can be used in central packet handlers of ISDN networks to process signaling or packet data of four ISDN subscribers. In this application, it may be used with or without the Am2055 Extended PCM Interface Controller (EPIC™).

The IDEC can be connected to the IOM interface of the EPIC that is connected to the PCM system highway. The EPIC implements concentration and time slot assignment functions. As an alternative, the IDEC may be directly connected to PCM highways (Figure 15).

The size (from 1 to 8 bits) and the position of the time slot associated with each HDLC controller is software programmable. In addition to the receive and transmit data highways, the IDEC accepts a third input connection for collision detection purposes. The mode of collision detection is programmable. A "collision highway" (or time slot) can be used for remote collision control, as a "clear to send" lead, or for local contention resolution among several IDECs.

Line cards in decentralized or mixed signaling/data packet handling architectures. The IDEC can be used on peripheral line cards to process D-channel packets for ISDN subscribers. An Am2055 Extended PCM Interface Controller (EPIC) has the layer 1 controlling capacity and a B-channel switching capacity for a total of 32 subscribers. The B- and D-channels and the control information for eight subscribers is carried over one IOM interface. Thus a line card dimensioned for 32 ISDN subscribers may employ up to eight IDECs, two for each IOM connection (Figure 16). The Am82520 High Level Serial Communication Controller (HSCC) with two HDLC channels, or another IDEC, may be used to transmit and receive signaling over the system highway in a common channel. Again, such a common channel may be shared among several line cards, due to the statistical multiplexing capability of these controllers.

In completely decentralized D-channel processing architectures, the processing capacity of a line card is usually dimensioned to avoid blocking situations even under maximum conceivable D-channel traffic conditions. It may sometimes be more advantageous to perform p-packet handling in a centralized manner while keeping s-packet handling on the line cards. A statistical increase in p-packet traffic then has no effect on the line card and can be easily dealt with by one of the modular architectures for a central packet handler shown in the previous section. A more effective sharing of the total p-packet handling capacity is the result, especially in a situation where p-packet traffic patterns vary widely from one subscriber group to another.

The use of IDEC in the mixed D-channel processing architecture is illustrated in Figure 17. The additional "transparent data" connections supported by the IDEC enable a merging of p- and s-packets into one D-channel. Possible collision situations are dealt with by the IDEC which uses either the additional collision detect line (Figure 17a), or a time slot on the system highway (Figure 17b) from the line card to the central packet handler.

Reset

After a hardware reset (pin RES), the configuration/command register bits are zeroed. No interrupts are active and all outputs are in a high impedance state. Table 5 sums up the state of the IDEC immediately after a hardware reset has been applied.

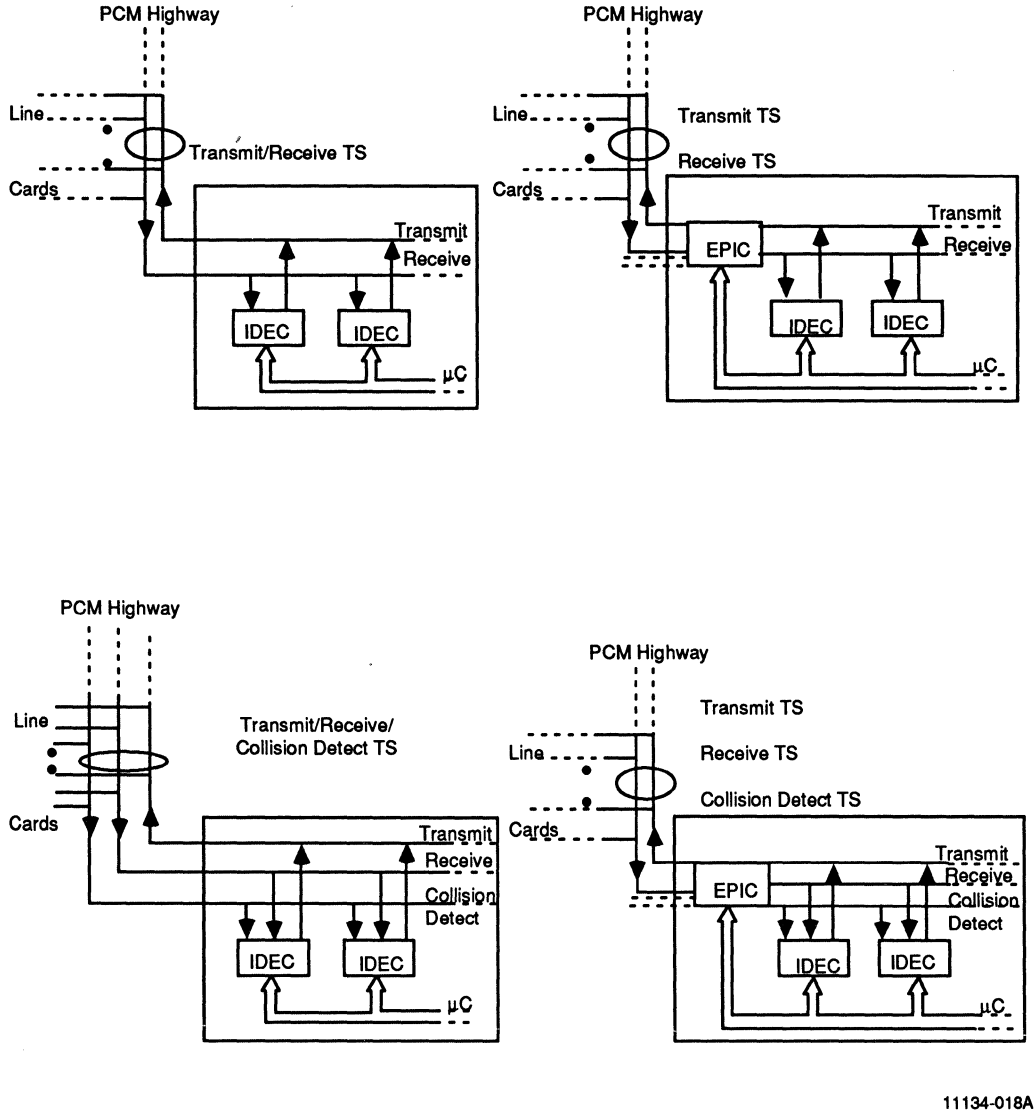
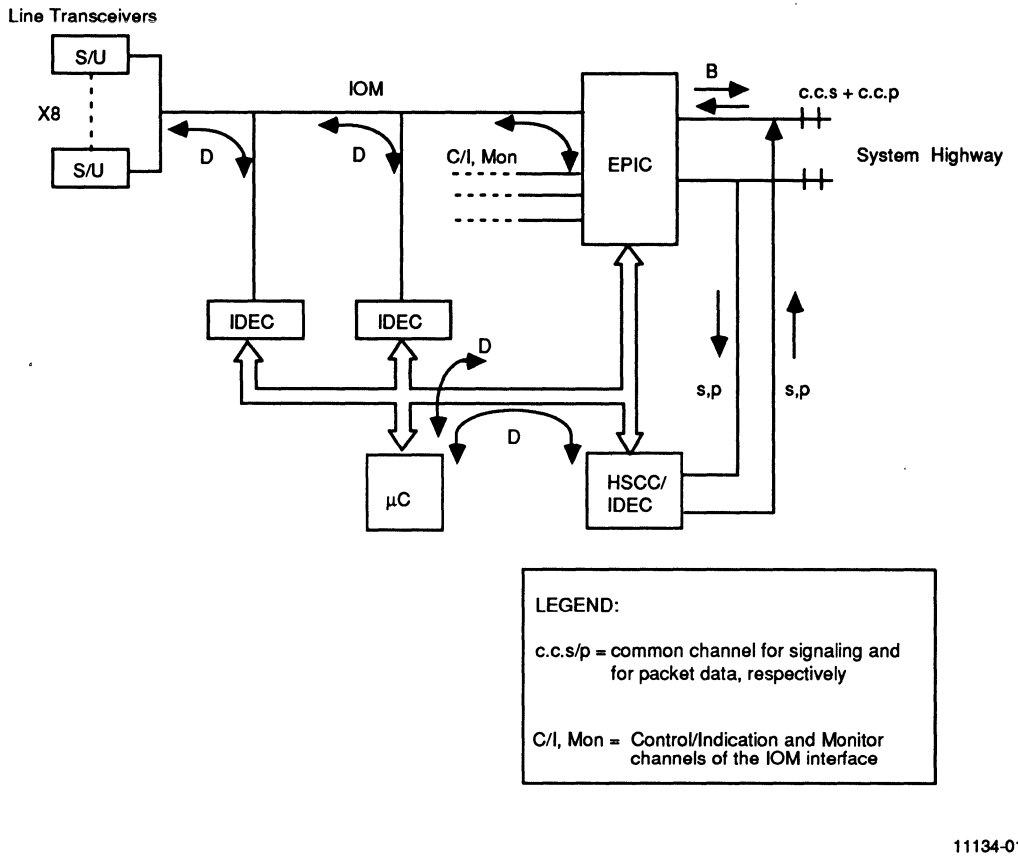
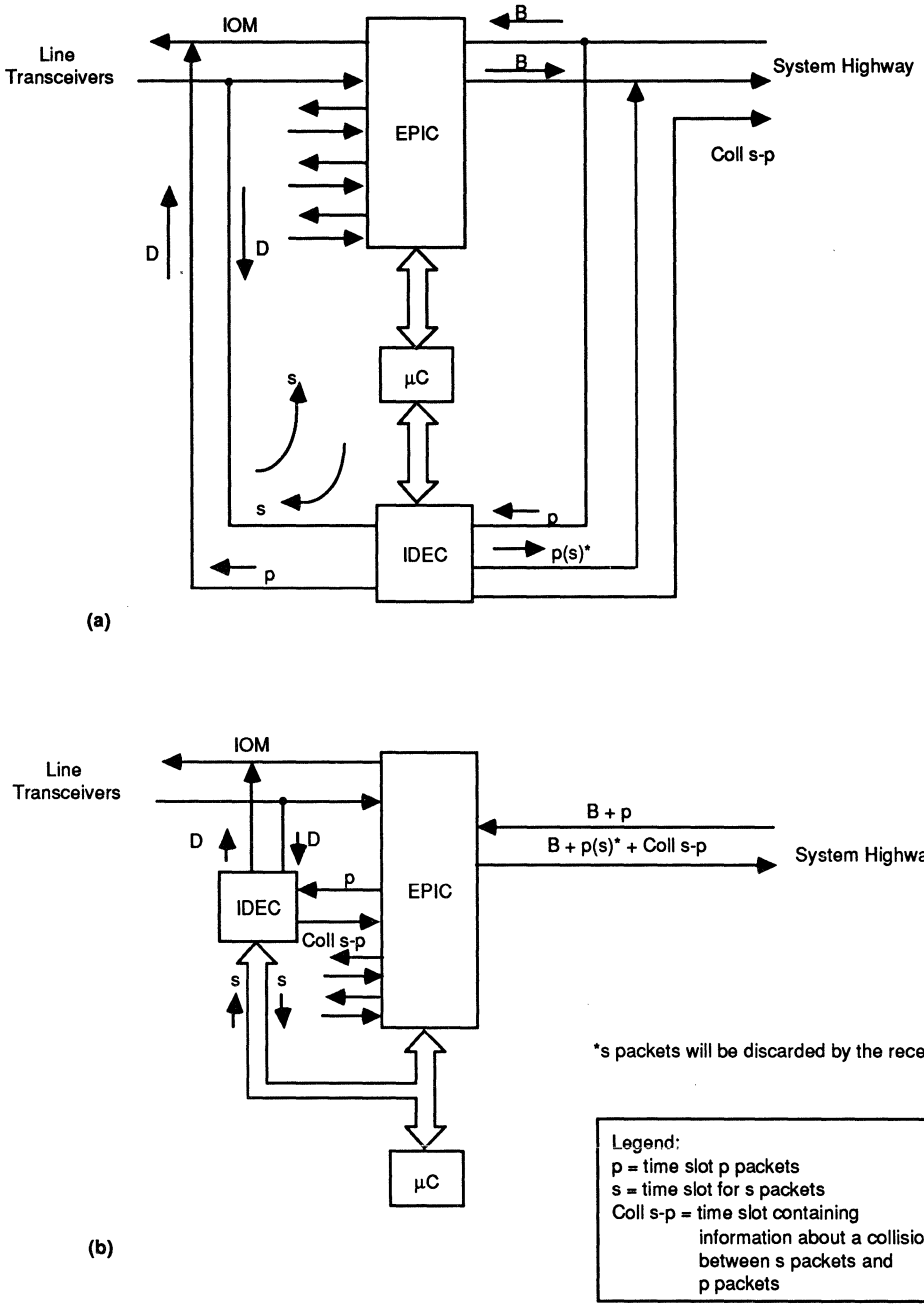


Figure 15. Use of IDEC in Central Signaling/Data Packet Handlers



11134-019A

Figure 16. Line Card in a Decentralized D-Channel Handling Architecture



11134-020A

Figure 17. IDEC on a Line Card in a Mixed D-Channel Processing Architecture

Table 5. State of IDEC after a Hardware Reset

Register Name	Value after Hardware Reset (hex)	Meaning
Common Register		
CCR	00	Single connection TS mode interrupt vector may be read on AD bus bits 0–3. Bits per frame: 257 to 512. Bit rate is equal to clock rate. Output drivers are of the push-pull type.
VISR	00	No interrupt from any IDEC channel.
VISM	00	All channel interrupts are enabled.
Individual registers I = A,B, C,D,		
ISTA	00	No interrupts from channel i.
ISM	00	All channel i interrupts enabled.
STAR	50	Transmit FIFO is ready to be written to. Receive line is idle.
CMDR	00	No commands
MODE	00	Test loop not active. No collisions will be detected (unconditional transmission). Interframe time fill = idle. Receiver deactivated. Channel i disabled (HIGH impedance output). Channel capacity is 2-bits/time slot.
RFBC	00	Zero bytes received.
TSR	00	Time slot 0 selected.

Initialization

The purpose of the initialization is to set the IDEC into a state where it is able to correctly transfer HDLC frames and to manage collisions according to the requirements of the application.

The initialization process is divided into two phases. First, the common settings are determined via the registers CCR and VISM. These registers determine the number of HDLC channels used, the serial interface configuration, and common characteristics of the serial input/output connections (Table 6).

During the second phase, each of the HDLC channels is initialized via its own register set as shown in Table 7.

Table 6. Initialization of IDEC (Common Bits)

Function	Register	Bits	Effect
Configuration	CCR	MDS1–0	Basic configuration and timing mode
Serial interface characteristics	CCR CRS	ODS	Output driver type is open-drain or push-pull Clock rate = 1 or 2x data rate
	BNS		Number of bits per PCM frame
Interrupt configuration	VISM CCR	MIC3–0 VIS	Mask any HDLC channel(s) VISR may be read on AD bus bits 0–3 or 4–7

Table 7. Initialization of HDLC Channels (Channel-per-Channel)

Function	Register	Bits	Effect
Serial interface	MODE	CMS1-0	Collision mode
		CCS1-0	Channel capacity
	TSR	TSR7-0	Time slot
HDLC controller	MODE	ITF	Interframe time fill pattern
		TLP	Test loop
		CAC	Activate channel (enable receiver + transmitter, enable data outputs)
		RAC	Activate HDLC receiver

Interrupt Structure

Special events are reported to the processor by an interrupt logic in the IDEC. This logic allows the connection of more than one IDEC to one interrupt input of a microcontroller.

The interrupt structure of the IDEC is depicted in Figure 18. Each HDLC channel of the circuit has its own Interrupt Status Register (ISTA) where up to five possible interrupt causes may be read directly. When an interrupt occurs in one of the HDLC channels, the corresponding bit is set in the ISTA register and the interrupt line (INT) is activated. Simultaneously, a bit in the Vectored Interrupt Status Register (VISR) is set that indicates which of the four HDLC channels initiated the interrupt. Thus, to determine the cause of an interrupt, the microcontroller performs successively a read of the VISR register (address 36/3F) and a read of the ISTA register which was indicated by the contents of VISR.

A read of the ISTA clears the register and deactivates the INT line.

The position that the four bits of the VISR occupy on the AD7–0 bus when the register is read, is programmable via the Vectored Interrupt Selection bit (VIS, CCR register). Thus, when VIS = 0, the VISR bits are read on AD bit positions 0–3, and when VIS = 1, VISR bits are read on AD bit positions 4–7. Unoccupied bit positions on the bus remain in a high impedance state.

The bits in VISR can be selectively masked by setting the corresponding bits in the Vectored Interrupt Status Mask (VISM) register to prevent controllers from generating an interrupt. In that case, interrupts remain internally stored (pending) but are not displayed in the VISR or ISTA registers. Further, ISTA interrupts pertaining to a particular channel may be selectively masked via the Interrupt Status Mask register of that channel. Pending interrupts cause the INT line to be activated, and they will be reported via ISTA (and VISR) only when the mask bits in ISM (and VISM) have been reset.

Processing

After being initialized via the configuration/mode registers listed in Tables 6 and 7, the IDEC is operational.

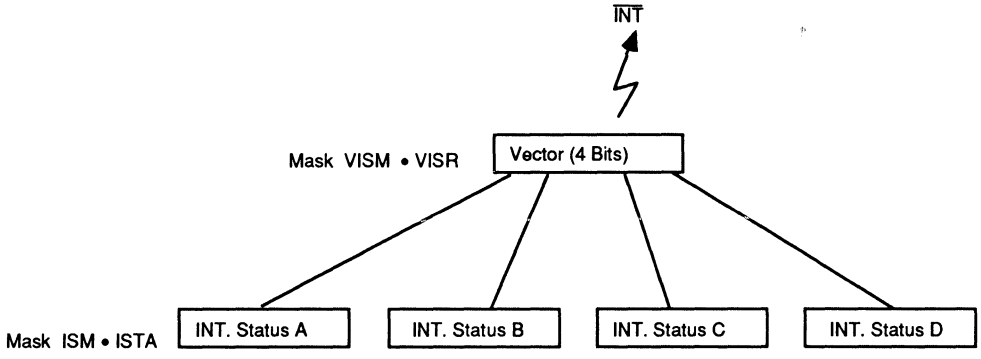
The control of the data transfer is performed by commands from the microcontroller written in the Command Register (CMDR). Events pertaining to the data transfer are reported by the Interrupt Status Register (ISTA) pointed to by the Vectored Interrupt Status Register (VISR). Other events that do not lead to interrupts may be monitored with the Status Register (STAR), and information about the receive frames is found in the RFIFO and in the Receive Frame Byte Counter (RFBC) Register.

The powerful FIFO logic, which consists of a 2 • 32 byte receive and a 2 • 32 byte transmit FIFO per channel, as well as an intelligent FIFO controller, builds a flexible interface to the upper protocol layers implemented in the microcontroller.

Receive Frame Processing

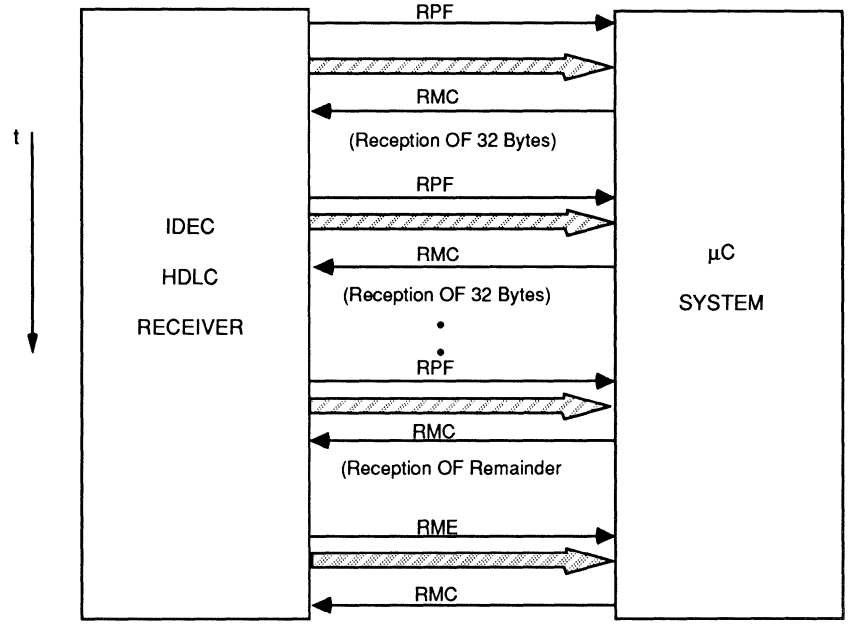
Reception of HDLC frames with three or more bytes between the opening and closing flags is always reported to the microcontroller. All bytes between the opening flag and the CRC field are stored in the RFIFO.

When the frame (excluding the CRC field) is no longer than 31 bytes, the whole frame is transferred in one block. The reception of the frame is reported by the Receive Message End (RME) interrupt. The length of the frame can be read out from an 8-bit register (RFBC). A status byte is appended to the data in the RFIFO after an RME interrupt. It includes information about the frame, such as frame aborted yes/no or CRC valid yes/no. The frame and the status byte remain stored until the microcontroller issues an acknowledgment (Receive Message Complete: RMC).



11134-021A

Figure 18. Interrupt Structure of the IDEC



11134-022A

Figure 19. Reception of an HDLC Frame

A frame longer than 31 bytes is transferred to the microcontroller in blocks of 32 bytes plus one remainder block of length 1 to 32 bytes. The reception of a 32 byte block is reported by a Receive Pool Full (RPF) interrupt and the data in RFIFO remains valid until this interrupt is acknowledged (RMC). This process is repeated until the reception of the remainder block reported by RME (Figure 19). Bits 0–4 of the RFBC register represent the number of bytes stored in the RFIFO (including the status byte). Bits 7–5 indicate the total number of 32 byte blocks that were stored until the reception of the remainder block. Bits 7–5 do not overflow when the counter status 7 has been reached and indicate in this case a message length greater than 223 bytes.

The contents of the RFBC register are valid only after the occurrence of the RME interrupt, and remain valid until the microprocessor issues an acknowledgment (RMC). All receive interrupts accumulated in the meantime are stored (along with the status bytes and respective frame lengths) inside the controller and transferred one by one to the microcontroller after each RMC acknowledgment. If a frame could not be stored due to a full FIFO, the microcontroller is informed of this via the Receive Frame Overflow interrupt (RFO).

Transmit Frame Processing

After checking the XFIFO status by polling the Transmit FIFO Write Enable (XFW) bit or after a Transmit Pool Ready (XPR) interrupt, up to 32 bytes may be entered by the microcontroller in XFIFO. Transmission of an HDLC frame is started when the Transmit HDLC Frame (XHF) command is issued. The HDLC controller will request another data block by an XPR interrupt if there are no more than 32 bytes in XFIFO and the frame close command bit (Transmit Message End XME) has not been set. When XME is set, all remaining bytes in XFIFO are transmitted, the CRC field and the closing flag of the HDLC frame are appended, and the controller generates a new XPR interrupt (Figure 20).

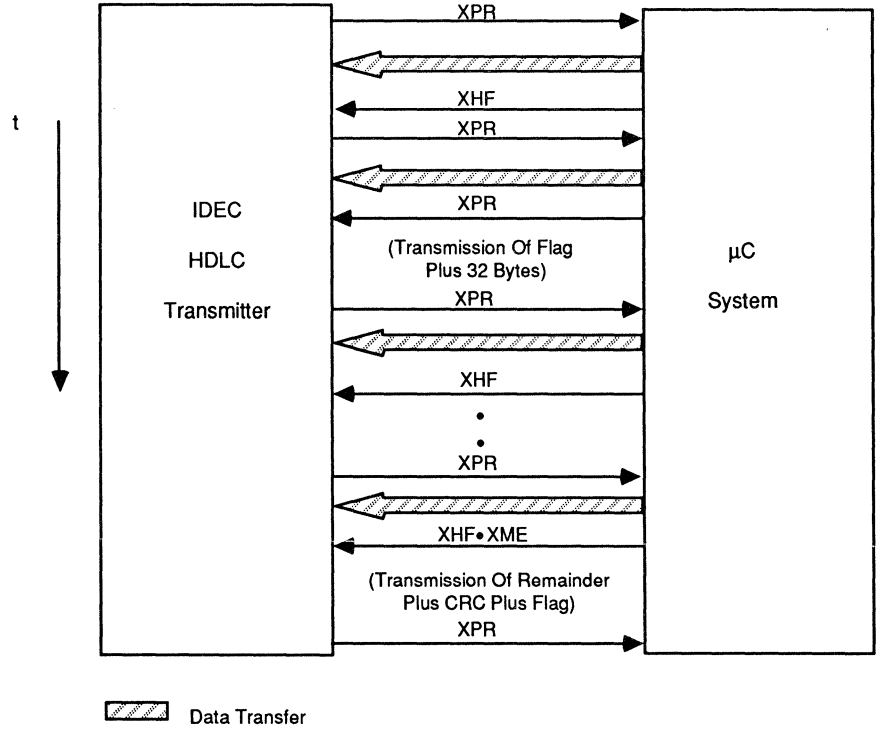


Figure 20. Transmission of an HDLC Frame

The microcontroller does not necessarily have to transfer a frame in blocks of 32 bytes. As a matter of fact, the sub-blocks issued by the microcontroller and separated by an XHF command can be between 1 and 32 bytes long.

If the XFIFO runs out of data and the XME command bit has not been set, the frame will be terminated with an abort sequence (seven "1"s) followed by interframe time fill, and the microcontroller will be advised by a Transmit Data Underrun (XDU) interrupt. An HDLC frame may also be aborted by setting the Transmit Reset (XRES) command bit.

Table 8 gives a summary of possible interrupts from the HDLC controller and the appropriate reaction to these interrupts.

Table 9 lists the most important commands which are issued by a microcontroller by setting one or several bits in the Command Register (CMDR).

Table 8. Possible Interrupt Causes and Reactions

Mnemonic	Meaning	Reaction
RPF	Receive Pool Full	Read 32 bytes from RFIFO and acknowledge with RMC
RME	Receive Message End	Read "RFBC4–0" bytes from RFIFO and acknowledge with RMC
RFO	Receive Frame Overflow	Error report for statistical purposes (loss of a complete frame). Probable cause: deficiency in software.
XPR	Transmit Pool Ready	Write data bytes in the XFIFO if the frame currently being transmitted is not finished or a new frame is to be transmitted, and issue an XHF (and possible XME) command.
XDU	Transmit Data Underrun	Acknowledged by a read of the ISTA. Possible causes: excessive software reaction times, or transmit data collision.

Table 9. List of Commands

Command Mnemonic	HEX	Bit 7–0	Meaning
RMC	80	1000 0000	Receive message complete. Acknowledges a block RPF) or a frame (RME) stored in the RFIFO.
RRES	40	0100 0000	Reset HDLC receiver. The RFIFO is cleared and the receiver enters the hunt phase.
RMD	20	0010 0000	Receive Message Delete. The part of the frame in the RFIFO is deleted and the rest of the frame will be ignored by the receiver.
XHF	08	0000 1000	Transmit HDLC Frame. Enables the transmission of the block entered last in the XFIFO. The frame is not yet complete.
XHFC	OA	0000 1010	Transmit HDLC Frame and close it with CRC and flag.
F_XHF	OC	0000 1100	Same as preceding, but used in Master mode to
F_XHFC	OE	0000 1110	enforce a transmission even in the case of a collision.
XRES	01	0000 0001	Reset Transmitter. Clears the XFIFO; any frame currently being transmitted is aborted.

DETAILED REGISTER DESCRIPTION

Register Address Layout

The register set consists of:

- one configuration register common to all four channels (CCR)
- a maskable vectored interrupt status register (VISR, VISM)

and, for each of the four channels, a set of individual registers (Figure 21).

In order to support the use of a 16-bit microcontroller, each register can be accessed with an even and an odd address value.

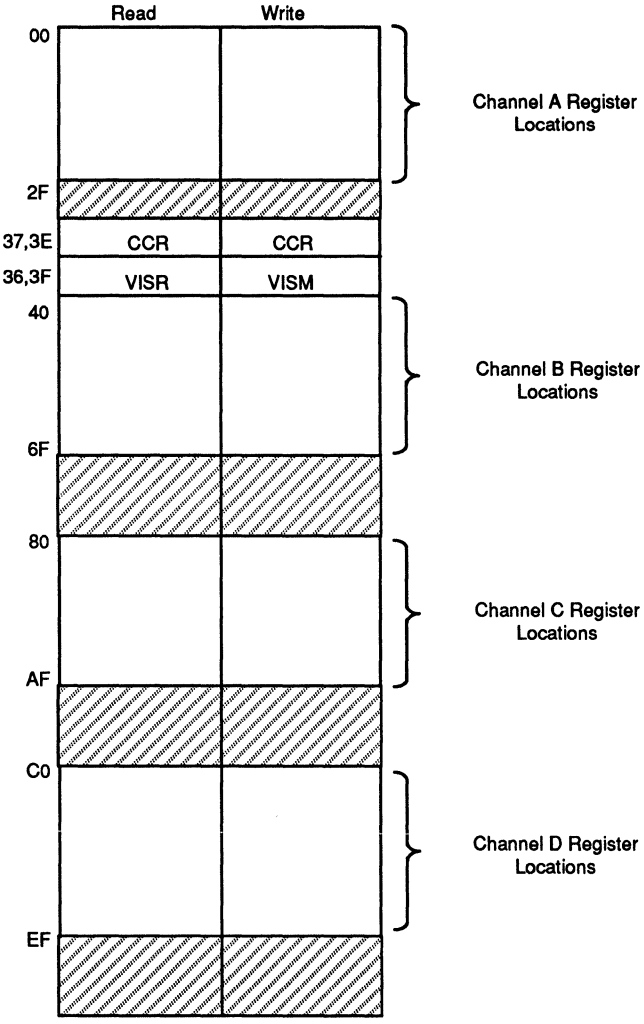


Figure 21. IDEC Register Map

The address map of the individual registers of each channel is shown in Table 10. In order to obtain the actual address of a register, a "base" has to be added to the address given in the table, as follows:

base = 00 for channel A
40 for channel B
80 for channel C
C0 for channel D

Table 10. Address Map of HDLC Channel Registers

Address Even		Odd	Read	Write
00	to	1F	RFIFO	XFIFO
20	or	29	ISTA	ISM
28	or	21	STAR	CMDR
22	or	2B	MODE	MODE
2C	or	25	RFBC	TSR

Register Description

Common Registers

Common Configuration Register (CCR) Read/Write. Value after reset: 00H

7							0
MDS1	MDS0	VIS	0	0	BNS	CRS	ODS

MDS1,0 Mode Select

MDS1	MDS0	Description
0	0	Single connection TS mode
0	1	Quad connection common control mode
1	0	Single connection IOM mode
1	1	Quad connection TS mode

VIS Vectored Interrupt Selection
1 IOM channel 4 to 7 (IOM mode), bus bits 4 to 7 for VISR
0 IOM channel 0 to 3 (IOM mode), bus bits 0 to 3 for VISR

BNS Bit Number Select
1 PCM frame is at most 256 bits long
0 PCM frame is 257 to 512 bits long

CRS Clock Rate Selection
1 DCL clock rate is equal to twice the data rate
0 DCL clock rate is equal to the data rate

ODS Output Driver Selection
0 Tri-state
1 Open drain

The ODS bit selects the driver type simultaneously on all data outputs (and control output SD2X in Master mode). However, in the Single connection IOM mode, SD0X is open-drain, independent of the value of ODS.

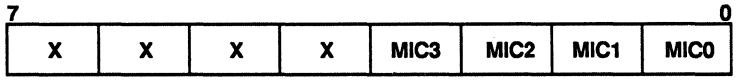
Vectored Interrupt Status Register (VISR) Read. Value after reset: 00H

7							0
X	X	X	X	IC3	IC2	IC1	IC0

IC0–3 Interrupt from Channel A–D

When VISR is read, these four bits are placed on the data bus with an offset determined by bit VIS (register CCR). Other bit positions on the bus remain high impedance.

Mask for Vectored Interrupt Status Register (VISM) Write. Value after reset: 00H



MIC0–3 Mask for Interrupt from Channel 0–3.

The mask bits are active HIGH.

A masked interrupt is not visible when VISR is read. Instead, it remains internally stored (pending). Any pending interrupt is generated and the corresponding IC0–3 bit is set when the mask bit is reset to zero.

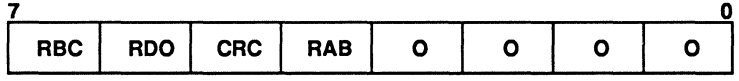
Individual Channel Registers

FIFOs

RFIFO (read), XFIFO (write)

The FIFOs have an identical address range. All the 32 addresses give access to the current FIFO location.

When the closing flag of a receive frame is detected, a status byte is appended to the data in the RFIFO. This byte has the following format:



RBC Receive Byte Count

The length of the received frame is N • 8 bits if RBC = 1 (NE {1, 2, 3,. . .}).

RDO Receive Data Overflow

If RDO = 1, part of the frame has been lost because the receive FIFO was full.

CRC CRC Check

The received CRC bytes were correct if CRC = 1.

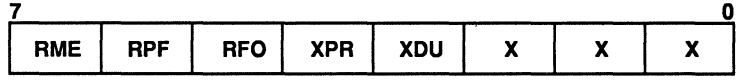
RAB: Receive Abort

RAB= 1 implies that the received frame was aborted.

A status byte equal to A0 indicates a correctly received frame.

Status/Command Registers

Interrupt Status Register (ISTA) Read. Value after reset: 00H



RME Receive Message End.

One complete frame of length less than or equal to 32 bytes, or the last part of a frame of length greater than 32 bytes is stored in the receive FIFO.

RPF Receive Pool Full

32 bytes of a frame have arrived in the receive FIFO. The frame is not yet completely received.

RFO Receive Frame Overflow

A complete frame was lost because no storage space was available in the RFIFO.

XPR Transmit Pool Ready

One data block may be entered into the XFIFO.

XDU Transmit Data Underrun

Transmitted frame was terminated with an abort sequence because either

1. no data was available for transmission in XFIFO and no XME command was issued, or;
2. a collision has occurred after at least one block of data has been completely transmitted, and thus an automatic retransmission cannot be attempted.

Note: It is not possible to transmit frames when an XDU interrupt remains unacknowledged.

Mask for Interrupt Status Register (ISM) Write. Value after reset: 00H

Each interrupt source in the ISTA register can be selectively masked by setting to "1" the corresponding bit in ISM. Masked interrupts are not indicated when ISTA is read. Instead, they remain internally stored and pending. An interrupt is generated after the mask is reset to zero.

Status Register (STAR) Read. Value after reset: 50H

7							0
XDOV	XFW	BSY	RNA	X	X	X	X

XDOV Transmit Data Overflow

More than 32 bytes have been written into the XFIFO.

XFW Transmit FIFO Write Enable

Data can be entered into the XFIFO.

BSY Busy state on the receive line

A "0" in this bit position indicates an "idle" state on the input data line (15 or more consecutive ones).

RNA Receive line Not Active

This indicates whether flags/frames are being received on the line (0) or not (1).

Command Register (CMDR) Write. Value after reset: 00H

7							0
RMC	RRES	RMD	X	XHF	FHF	XME	XRES

RMC Receive Message Complete

Reaction to RPF or RME interrupt. The receive frame (or one pool of data) has been read and the corresponding RFIFO storage space is freed.

RRES Receiver Reset

HDLC receiver is reset; the receive FIFO is cleared of any data.

RMD Receive Message Delete

Reaction to RPF or RME interrupt. The entire frame is to be ignored by the receiver. The part of frame already stored is discarded.

XHF Transmit HDLC Frame

Transmission of an HDLC frame is (or of a block thereof) initiated.

FHF Force HDLC Frame

Used in the Master collision mode (CMS1,0 = 11). When this bit is set and a Transmit HDLC Frame (XHF) command is issued, the controller aborts the frame from CDR (if any) by sending seven "1"s on SD0X and then starts transmission.

XME Transmit Message End

Indicates that the current transmit frame is to be closed with CRC and flag.

XRES Transmitter reset

HDLC transmitter is reset, XFIFO is cleared of any data, and the HDLC frame currently being transmitted (if any) is aborted.

Mode Register (MODE) Read/Write. Value after reset: 00H

7							0
TLP	CMS1	CMS0	ITF	RAC	CAC	CCS1	CCS0

TLP Test Loop

Input and output of HDLC channel are connected together (TLP = 1). The test loop is either transparent (if MDS1,0 = 01, 10) or not (if MDS1,0 = 00, 11).

CMS1,0 Collision Mode Select

CMS1	CMS0	Descriptions
0	0	Unconditional transmission
0	1	Slave Mode
1	0	Multi-master mode
1	1	Master mode

ITF Interframe Time Fill

Idle (ITF = 0) or flags (ITF = 1) are used as interframe time fill.

RAC Receiver Active

Receiver is activated (1) or deactivated (0).

CAC Channel Active

A channel is completely disabled (receiver and transmitter are inactive, transmit line is high impedance, no $\overline{TSC}$ is output) as long as CAC is "0". Only TS modes (MDS1,0 = 00 or 11).

CCS1,0 Channel Capacity Select

These bits select the number of bits in the time slot where data are received and transmitted. They have a significance only when MDS1,0 = 00 or 11 (Single connection TS mode and Quad connection TS mode).

The bit rates given below assume a channel repetition rate of 8 kHz.

CCS1	CCS0	Time slot Width	Channel Data Rate
0	0	2 bits	16 kbs
0	1	1 bit	8 kbs
1	0	8 bits	64 kbs
1	1	7 bits	56 kbs

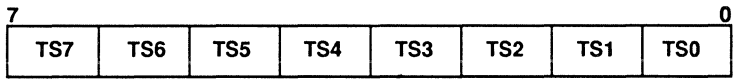
Receive Frame Byte Counter (RFBC) Read. Value after reset: 00H

7							0
RDC7	RDC6	RDC5	RDC4	RDC3	RDC2	RDC1	RDC0

RDC7–0 Receive Data Count

Total number of bytes of received frame, including the status byte. The contents of the register are valid after an RME interrupt. RDC4–0 indicate the length of the data block currently available in the receive FIFO. RDC7–5 count the number of full 32-byte blocks of a frame which have already been received. If the frame length exceeds 223 bytes, RDC7-5 hold the value "111"; only RDC4–0 continue to count modulo 32.

Time Slot Register (TSR) Write. Value after reset: 00_H



TS7–0 Time Slot Select

Determine the particular time slot where the HDLC controller is to receive and transmit. This register has a significance only when MDS1,0 = 00, 10 or 11 (single connection modes and quad connection TS mode). The register gives the position of a time slot (either 1, 2, 7 or 8 bits wide, cf. CCS1,0) in 2-bit increments (2-bit resolution). The position of the time slot is relative to a Frame Sync signal that marks the beginning of a PCM frame.

ABSOLUTE MAXIMUM RATINGS

Ambient temperature under bias . . . . . 0 to +70°C
Storage temperature . . . . . 65° to +125°C
Voltage on any pin with
respect to ground . . . . . −0.4 to V_{DD} +0.4 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Ambient Temperature (T_A) . . . . 0 to +70°C
Supply Voltage (V_{DD}) . . . . . 5 V ±5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges

T_A = 0 to +70°C; V_{DD} = 5 V ±5%, V_{SS} = 0 V.

Parameter Symbol	Parameter Descriptions	Test Conditions	Limit Values Min.	Max.	Unit
V_{IL}	Input low voltage		−0.4	0.8	V
V_{IH}	Input high voltage		2.0	V_{CC} + 0.4	V
V_{OL}	Output low voltage	I_{OL} = 2 mA		0.45	V
V_{OH}	Output high voltage	I_{OH} = −400 µA	2.4		V
V_{OH}	Output high voltage	I_{OH} = −100 µA	V_{DD} −.5		V
I_{CC}	Power operational supply current	V_{DD} = 5 V, Inputs at 0 V/V_{DD}, No output loads			mA
	power down				mA
I_{LI}	Input leakage current	0 V < V_{IN} < V_{DD} to 0 V		+10	µA
I_{LO}	Output leakage current	0 V < V_{OUT} < V_{DD} to 0 V		+10	µA

CAPACITANCES

T_A = 25°C, V_{DD} = 5 V ±5%, V_{SS} = 0 V.

Parameter Symbol	Parameter Descriptions	Test Conditions	Limit Values Min.	Max.	Unit
C_{IN}	Input capacitance			7	pF
C_{IO}	I/O			7	pF

SWITCHING CHARACTERISTICS

T_A = 0 to +70°C, V_{DD} = 5 V ±5%

Inputs are driven to 2.4 V for a logical "1" and to 0.4V for a logical "0". Timing measurements are made at 2.0 V for a logical "1" and at 0.8 V for a logical "0". The AC testing input/output waveforms are shown below.

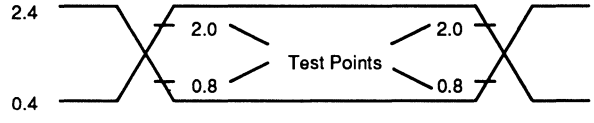


Figure 22. Input/Output Waveform for AC Tests

11134-025A

2

Microcontroller Interface Timing

Timing Characteristics

Parameter Symbol	Parameter Descriptions	Test Conditions	Min.	Max.	Unit
t_{AA}	ALE pulse width		50	—	ns
t_{AL}	Address setup time to ALE		20	—	ns
t_{LA}	Address hold time from ALE		10	—	ns
t_{RR}	$\overline{RD}$ pulse width		120	—	ns
t_{RD}	Data output delay from $\overline{RD}$		—	120	ns
t_{DF}	Data float delay from $\overline{RD}$		—	25	ns
t_{RI}	$\overline{RD}$ control interval		75	—	ns
t_{WW}	$\overline{WR}$ pulse width		60	—	ns
t_{DW}	Data setup time to $\overline{WR}$+$\overline{CS}$		30	—	ns
t_{WD}	Data hold time from $\overline{WR}$+$\overline{CS}$		10	—	ns
t_{WI}	$\overline{WR}$ control interval		70	—	ns

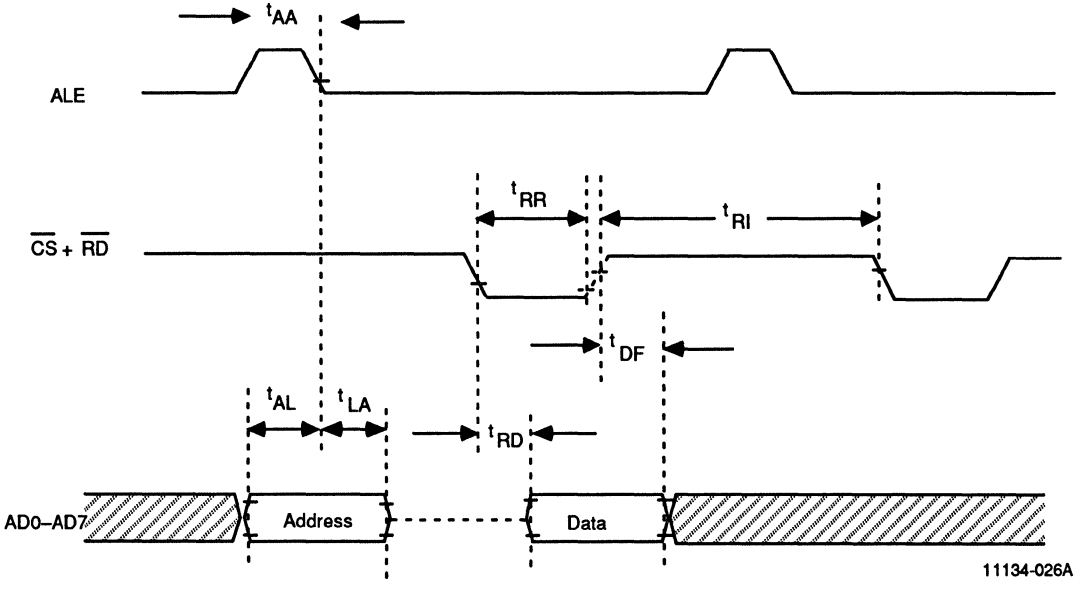


Figure 23. µP Read Cycle

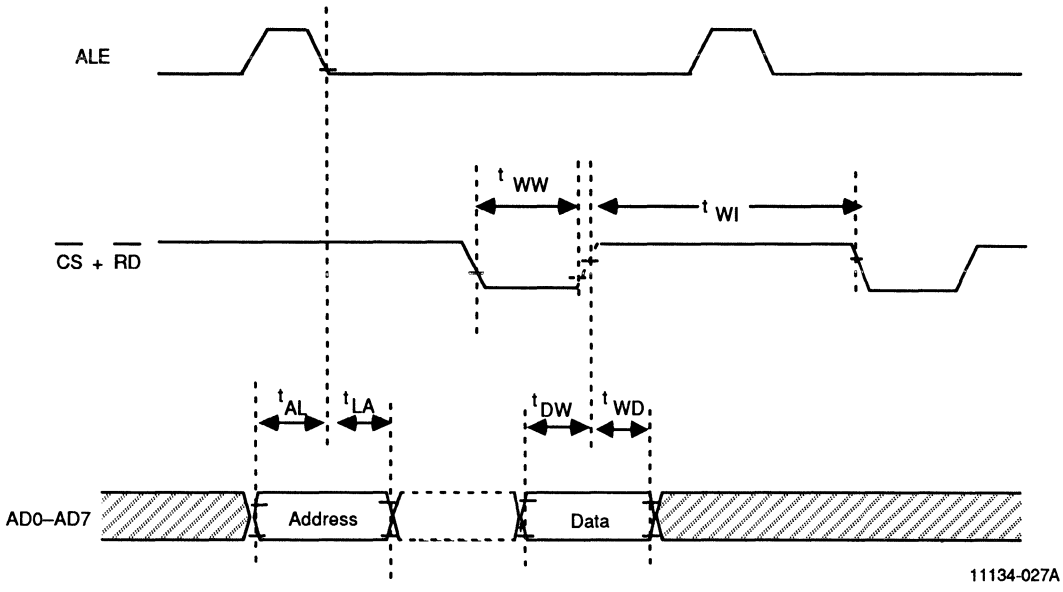
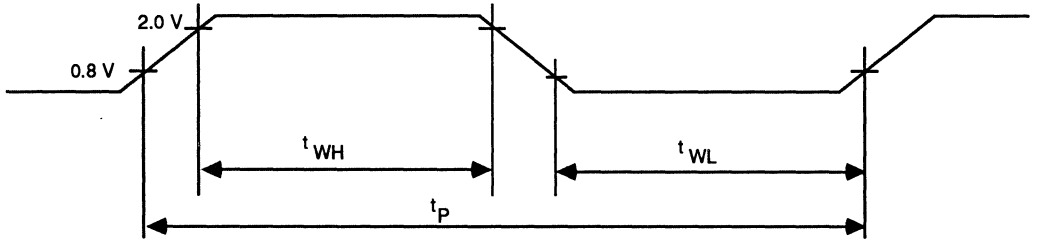


Figure 24. µP Write Cycle

SERIAL INTERFACE TIMING
DCL CHARACTERISTICS

Parameter Symbol	Parameter Descriptions	Test Conditions	Min.	Typ.	Max.
t_P	DCL period	single clock rate	230		
		double clock rate	160		
t_{WH}	DCL HIGH	single clock rate	90		
		double clock rate	50		
t_{WL}	DCL LOW		70		



11134-028A

Figure 25. Definition of DCL Period and Width

FSC Characteristics

Parameter Symbol	Parameter Descriptions	Test Conditions	Min.	Typ.	Max.
t_{FS}	FSC setup time		60		
t_{FH}	FSC hold time		30		
t_{ODD}	Output data delay delay from DCL			60	
t_{IDS}	Input data setup		25		
t_{IDH}	Input data hold		20		
t_{ODF}	Output data delay from FSC	See note		150	

Notes: This delay is applicable in two cases only:
1. When FSC appears for the first time; e.g., at system power-up
2. When the number of bits in the PCM frame is not equal to either 256 or 512

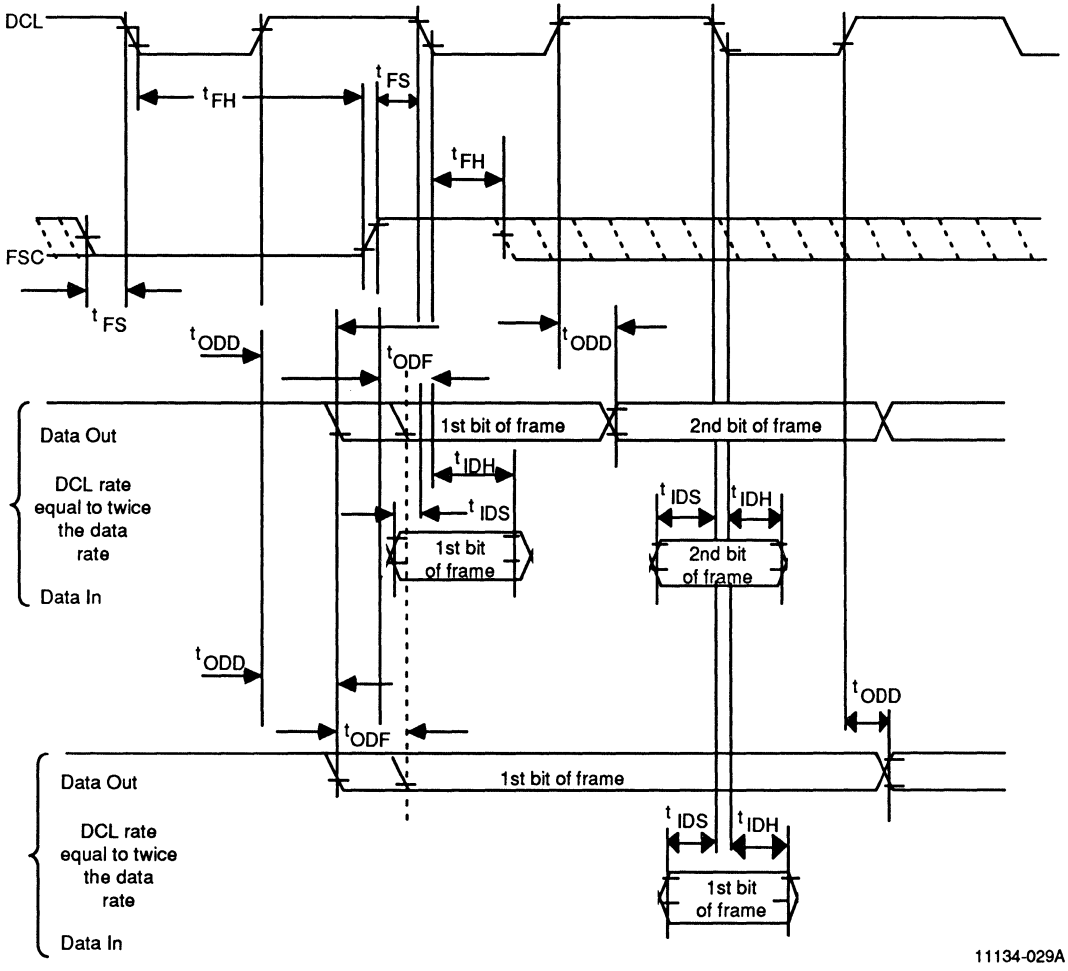


Figure 26. FSC Timing Characteristics

FSC in Quad Connection Common Control Mode

Parameter Symbol	Parameter Descriptions	Test Conditions	Min.	Typ.	Max.
t_{FS1}	FSC set up time		60		
t_{FH1}	FSC hold time		30		
t_{OZD}	Output data from high impedance to active			80	
t_{ODZ}	Output data from active to high impedance			40	
t_{ODD}	Output data delay from DCL			60	
t_{IDS}	Input data setup		25		
t_{IDH}	Input data hold		20		

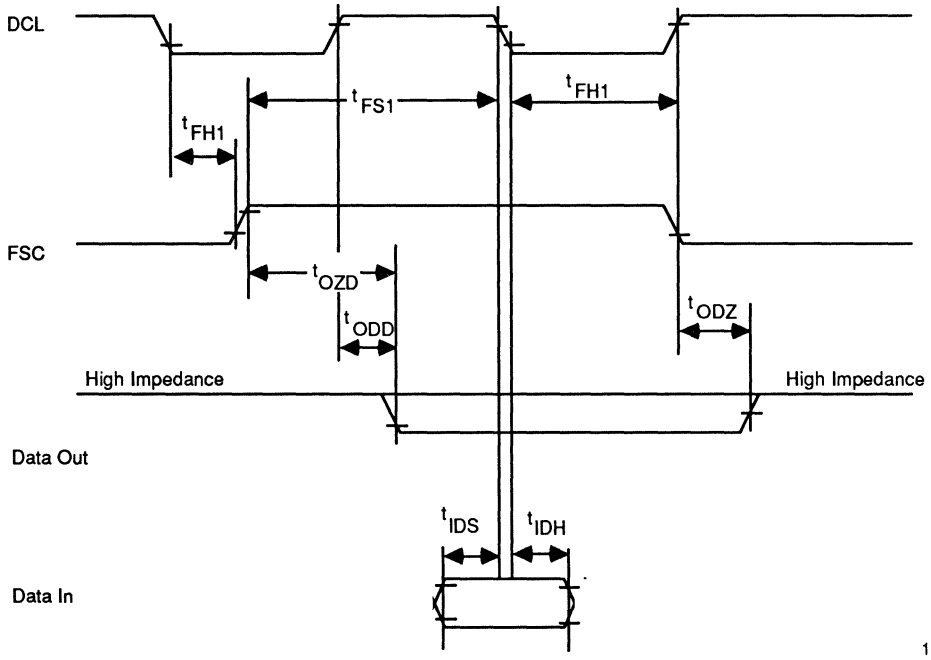


Figure 27. FSC (Strobe) Characteristics

Parameter Symbol	Parameter Descriptions	Test Conditions	Min.	Max.	Unit
t_{ODD}	Output data delay from DCL			60	
t_{IDS}	Input data setup		25		
t_{IDH}	Input data hold		20		
t_{TCD}	$\overline{TSC}$ delay from DCL			60	

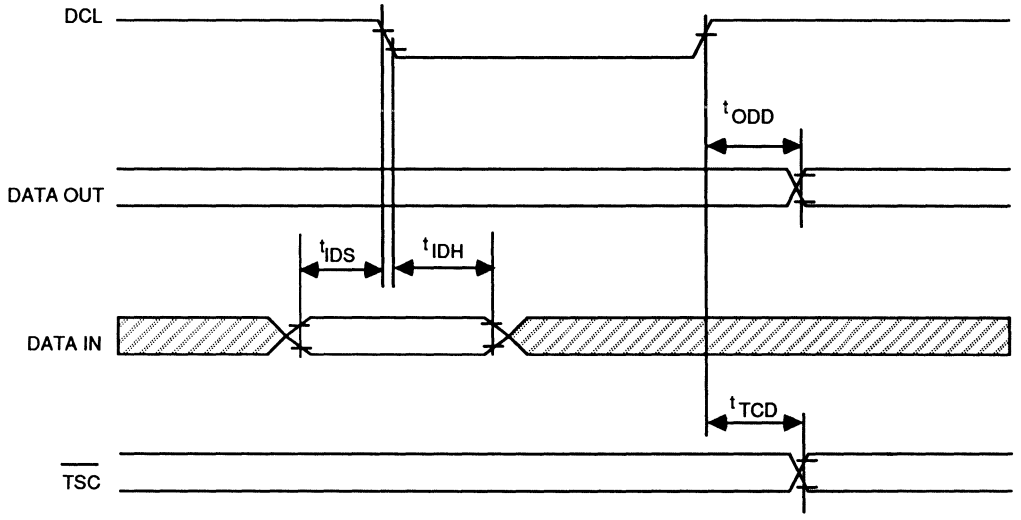


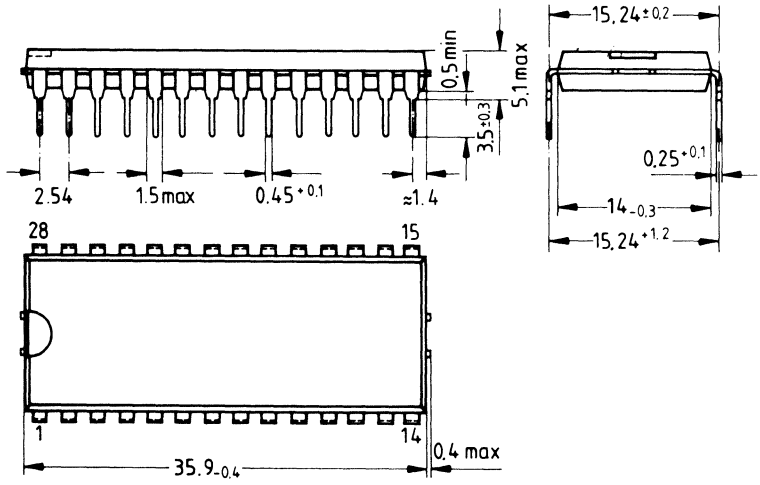
Figure 28. Data I/O Characteristics

Data out:	SD0X in Single connection modes SD0X, SD1X, SD2X, SD3X in Quad connection modes SD1X, SD2X in Master mode
Data in:	SD0R in Single connection mode SD0R, SD1X, SD2R, SD3R in Quad connection modes CDR in Slave, Multi-master and Master modes

RES Characteristics

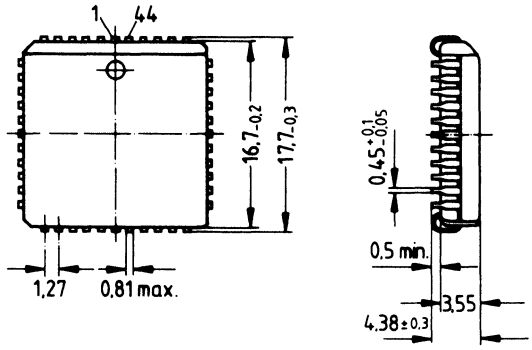
Parameter Symbol	Parameter Descriptions	Test Conditions	Min.	Typ.	Max.
t_{RWL}	RES HIGH		4xT_P		

PHYSICAL DIMENSIONS
PD 028



Note: Physical dimensions are in mm.

PHYSICAL DIMENSIONS (continued)
PL 044



Note: Physical dimensions are in mm.

Preliminary

Am2080/B

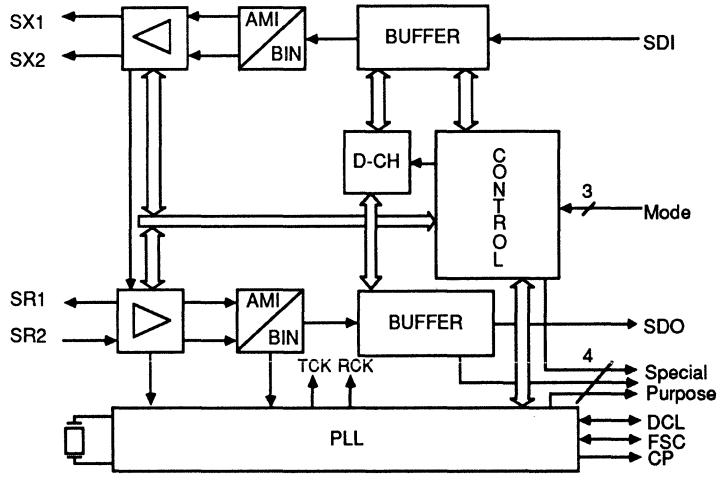
S-Bus Interface Circuit (SBC)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Full duplex 2B + D S/T-interface transceiver according to CCITT I.430
- Conversion of the frame structure between the S/T and IOM™ interfaces
- D-channel access control
- Activation and deactivation procedures according to CCITT I.430
- Built-in wake-up unit for activation from power-down state
- Adaptively switched receive thresholds
- Control via IOM interface
- NT, TE and LT operating modes
- Receive timing recovery according to selected operating mode
- Frame alignment with absorption of phase wander in trunk line applications
- Switching of test loops
- Advanced CMOS technology
- Low power consumption:
  -standby less than 4 mW
  -active max 60 mW

BLOCK DIAGRAM



11135-001A

Publication # 11135 Rev. B Amendment /0
Issue Date: June 1989

GENERAL DESCRIPTION

The Am2080 S-Bus Interface Circuit (SBC) implements the four-wire S/T-interface used to link voice/data terminals to an ISDN. Through selection of operating mode, the device may be employed in all types of applications involving an S-interface. In particular, two or more SBCs can be used to build a point-to-point, passive bus, extended passive bus, or star configuration.

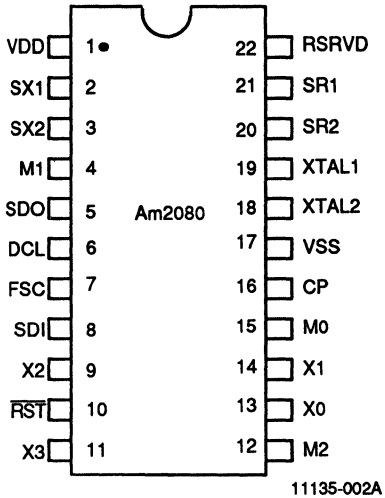
Specific ISDN applications of the SBC include: ISDN terminals, ISDN network termination (central office and PABX applications), and PABX trunk lines to central office.

The device provides all electrical and logical functions according to CCITT recommendation I.430. These include mode-dependent receive timing recovery, D-channel access and priority control, and automatic handling of activation/deactivation procedures. The SBC does not require direct microprocessor control.

The SBC is an IOM compatible, 22-pin CMOS device. It operates from a single +5 V supply and features a power-down state with very low power consumption.

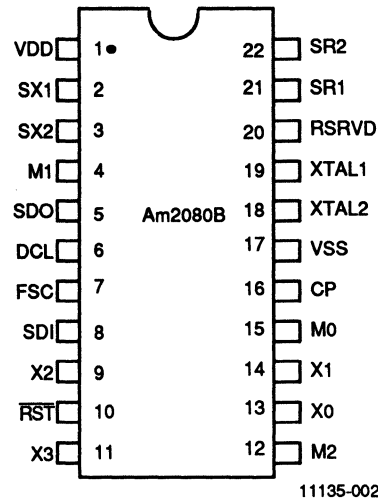
CONNECTION DIAGRAMS
Top View

22-Pin DIP



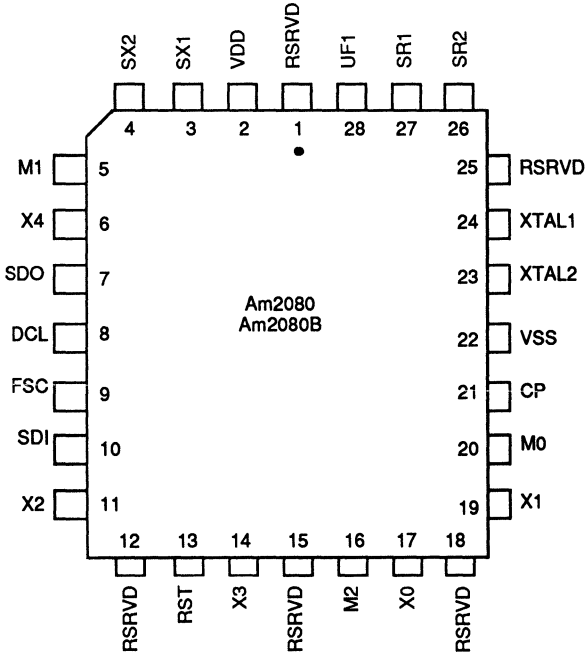
11135-002A

22-Pin DIP



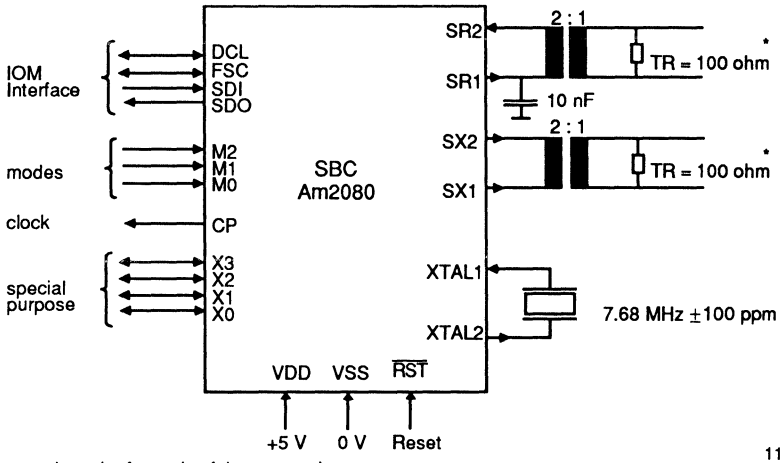
11135-002A

28-Pin PLCC



Note: Pin 1 is marked for orientation.

LOGIC SYMBOL



*Terminating resistors only at the far ends of the connection

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing

AM2080
AM2080B

P **C** **B**

e. OPTIONAL PROCESSING
Blank = Standard Processing
B = Burn-in

d. TEMPERATURE RANGE
C = Commercial (0 to + 70°C)

c. PACKAGE TYPE
P = 22-Pin Plastic Dip (PD 022)
J = 28-Pin Plastic Leaded Chip Carrier (PLR 028)

b. SPEED OPTION
Not Applicable

a. DEVICE NUMBER/DESCRIPTION
Am2080
S-Bus Interface Circuit (SBC)

Valid Combinations	
AM2080 AM2080B	PC, JC, PCB, JCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

CP
(Input/Output)

Clock pulses, depending on selected operating mode; CP provides synchronous clocks.

DCL
Serial Data Clock, IOM Interface (Input/Output)

FSC
Frame Sync, IOM Interface (Input/Output)

M2–M0, X3
Setting of Operating Mode (Input)

$\overline{\text{RST}}$
Reset, Active Low (Input)

SDI
Serial Data In, IOM Interface (Input)

SDO
Serial Data Out, IOM Interface (Output)

SR1
S-Bus Receiver (Output)

2.5 V Reference Output.

SR2
S-Bus Receiver (Input)

Signal Input.

SX1
S-Bus Transmitter (Positive Output)

SX2
S-Bus Transmitter (Negative Output)

UFI
User Filter (Output)

Connection for external pre-filter for S-Bus receiver, if used. Available on PLCC only.

VDD
(Input)

Power Supply, + 5 V ± 5%.

VSS
(Input)

Power Supply, Ground.

X2–X0
(Input/Output)

Functions depend on the selected operating mode; see Operating Modes section.

X4
Mode Selection (Input)

Must be tied Low (0) if external pre-filter for S-Bus receiver is used (pin UFI).

XTAL1
(Input)

Connection for External Crystal or Input for External Clock Generator

XTAL2
(Output)

Connection for external crystal, not connected when external clock generator is used.

FUNCTIONAL DESCRIPTION

The Am2080 S-bus interface circuit performs the OSI layer 1 functions for the S/T interface of the ISDN basic access.

General Functions and Device Architecture

The common functions for all operating modes are:

- line transceiver functions for the S interface according to the electrical specifications of CCITT I.430
- dynamically adaptive threshold control for the receiver
- conversion of the frame structure between IOM and S interfaces
- code conversion from/to binary to/from Alternate Mark Inversion

Mode-specific functions are:

- receive timing recovery
- timing generation using IOM timing synchronous to system, or vice versa
- D-channel access control and priority handling
- D-channel echo bit generation
- activation/deactivation procedures, triggered by primitives received over the IOM interface or by INFOs received from the line
- frame alignment according to CCITT Q.503
- execution of test loops

Analog Functions

The full-bauded pseudo-ternary pulse shaping is achieved with the integrated transmitter which is realized as a voltage-limited current source. A current of 7.5 mA is delivered over SX1-SX2, which yields a voltage of 1.5 V over 200 ohms. The receiver is designed as a threshold detector with adaptively switched threshold levels. Pin SR1 delivers 2.5 V as an output, which is the virtual ground of the input signal on pin SR2.

An external transformer of ratio 2:1 is needed in both receive and transmit direction to provide for isolation and transform voltage levels according to CCITT recommendations.

Digital Functions

A DPLL circuitry working with a frequency of 7.68 MHz 100 ppm serves to generate the 192 kHz line clock from the reference clock delivered by the network and to extract the192 kHz line clock from the receive data stream.

The 7.68 MHz clock may be generated with the use of external crystal between pins XTAL1 and XTAL2. It may also be provided by an external oscillator, in which case XTAL2 is left unconnected. The "Control" (Block Diagram) block includes the logic to detect OSI layer 1 commands and to communicate with external layer 1 or layer 2 devices via the IOM interface.

An incorporated finite state machine controls ISDN OSI layer 1 activation/deactivation.

The D-channel access procedure according to CCITT I.430, including priority management, is fully implemented in the SBC. When used as an S-bus master in a multipoint configuration, the device generates the echo bits necessary for D-channel collision detection. In the NT-mode, the echo channel may be made externally available through an auxiliary pin and thus "Intelligent NTs" (star configuration) may be implemented. In terminal applications (TE) the Q channel as specified by I.430 is supported, stepping A6 and up. The SBC sends a binary 1 in FA bit position to allow another terminal to use the extra transmission capacity.

The buffer memory serves to adapt the different bit rates of the S and IOM interfaces. In addition, in trunk line applications it absorbs the possible deviation between two system clocks, according to CCITT Q.503.

Operating Modes

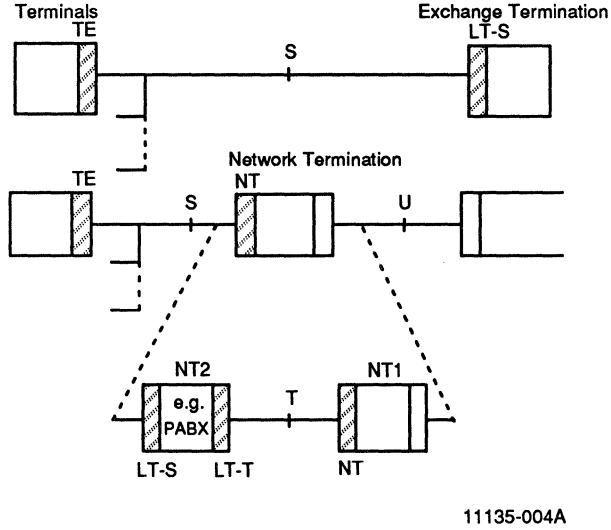
The operating modes are determined by pin strapping on pins M0 to M2. The four basic operating modes are: TE, NT, LT-S, and LT-T. In three of these operating modes, the IOM may be programmed to function in the normal mode, in the inverted mode (clock frequency 512 kHz), or in the inverted MUX mode (clock frequency 4096 kHz). To see which IOM timing mode is applicable in the four basic operating modes, refer to Table 1. The functions of the operating mode-specific pins are given in Table 1 as well.

SYSTEM INTEGRATION

The SBC implements the four-wire "S" and "T" interfaces used in the ISDN basic access. It may be used at both ends of these interfaces. The applications include:

- ISDN terminals (TE)
- ISDN network termination (NT)
- ISDN subscriber line termination (LT-S)
- ISDN trunk line termination (LT-T) (PABX connection to Central Office)

These applications are shown in Figure 1 where the usual nomenclature as defined by the CCITT for the basic access functional blocks and reference points has been used.



Note: Shaded areas indicate where SBC can be used.

Figure 1. Applications of the Am2080 SBC

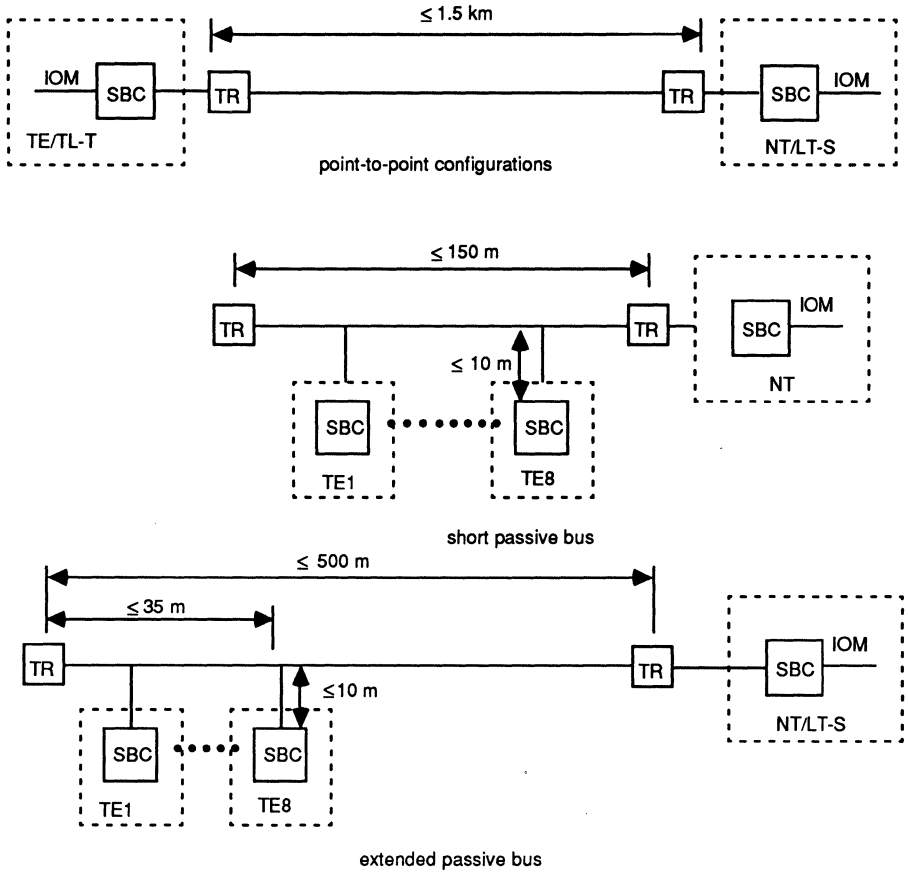


Figure 2. Some S-Interface Wiring Configurations

Some of the wiring configurations possible with the SBC for the S-interface are shown in Figure 2 with approximate typical distances. TR stands for terminating resistor. The maximum line attenuation tolerated by the SBC is 15 dB at 96 kHz.

Figure 3 gives an example of an application of the SBC in an IOM (ISDN Oriented Modular) architecture. By separate implementation of OSI layer 1 and layer 2 functions, and through unified control procedures, the architecture provides flexibility with respect to various transmission techniques. The IOM devices are all low-power, high-integration, single +5 V supply CMOS devices. Through mode switching, each device may be used in several applications. With one limited set of devices, all ISDN basic access configurations are covered. Note that none of the compatible layer 1 devices (SBC, IBC, IEC™) require direct microprocessor control. The IOM interface provides all necessary communication functions.

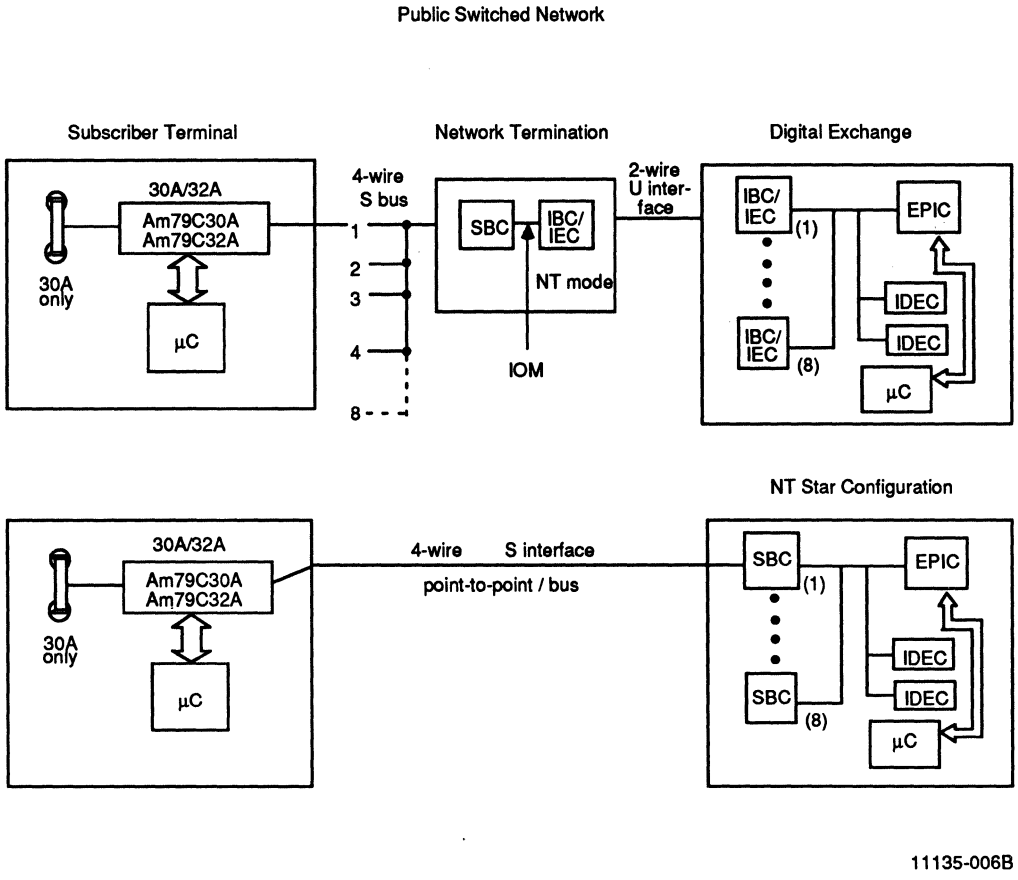


Figure 3. ISDN Oriented Modular (IOM) Architecture

SBC	Am2080	S-Bus Interface Circuit	IDC	Am79C32A	ISDN Data Controller
IBC	Am2095	ISDN Burst Transceiver Circuit	EPIC	Am2055	Extended PCM Interface Controller
IEC	Am2090	ISDN Echo Cancellation Circuit	IDEC	Am2075	ISDN D-channel Exchange Controller
DSC	Am79C30A	Digital Subscriber Controller			

Table 1. Operating Modes and Functions of Mode-Specific Pins of the Am2080 SBC

Appli-cation	Operation of IOM Interface	M2	M1	M0	DCL	FSC	CP	X3	X2	X1	X0
TE	inverted mode	0	0	0	O:512 kHz*	O:8 kHz*	O:1536 kHz*	I:ENCK	O:2560 kHz	O:3840 kHz	O:RDY
TE	inverted mode	0	0	1	O:512 kHz*	O:8 kHz*	O:1536 kHz*	I:ENCK	O:1280 kHz	O:3840 kHz	O:RDY
TE	normal mode	0	1	0	O:512 kHz*	O:8 kHz*	O:1536 kHz*	I:ENCK	O:ECHO	O:3840 kHz	I:CON
LT-T	MUX mode inverted	0	1	1	I:4096 kHz	I:8 kHz	O:512 kHz	I:fixed at 1	I:TS2	I:TS1	I:TS0
LT-T	normal mode	0	1	1	I:512 kHz	I:8 kHz	O:512 kHz*	I:fixed at 0	I:fixed at 0	I:fixed at 0	I:CON
NT	normal mode	1	1	1	I:512 kHz	I:8 kHz	I:SCZ	I:BUS	I:SSZ	I:DEX	I/O:DE
LT-S	MUX mode inverted	1	0	0	I:4096 kHz	I:8 kHz	I:fixed at 0	I:BUS	I:TS2	I:TS1	I:TS0
LT-S	normal mode	1	1	0	I:512 kHz	I:8 kHz	I:fixed at 0	I:BUS	I:fixed at 0	O:7680 kHz	I:fixed at 0
LT-S	normal mode	1	1	0	I:512 kHz	I:8 kHz	I:fixed at 0	I:BUS	o:192 kHz*	O:7680 kHz	I:fixed at 1

*synchronized to S
I: input
O: output

SCZ Send continuous binary zeros (96 kHz) (if CP = 0)
ENCK Enable clock at all times
BUS Bus configuration specified
TS2–0 Timeslot number on IOM
SSZ Send single binary zeros (2 kHz)
DEX D-channel echo external/internal
RDY D-channel status on S-interface
CON Connected to S-bus

2

S Interface

According to CCITT recommendation I.430, a modified AMI code with 100% pulse width is used on the S interface. A logical 1 (one) corresponds to a neutral level (no current), whereas logical 0s are coded as alternating positive and negative pulses. An example of a modified AMI code is shown in Figure 4.

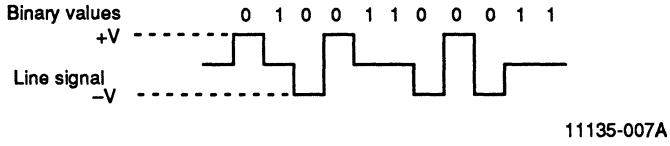
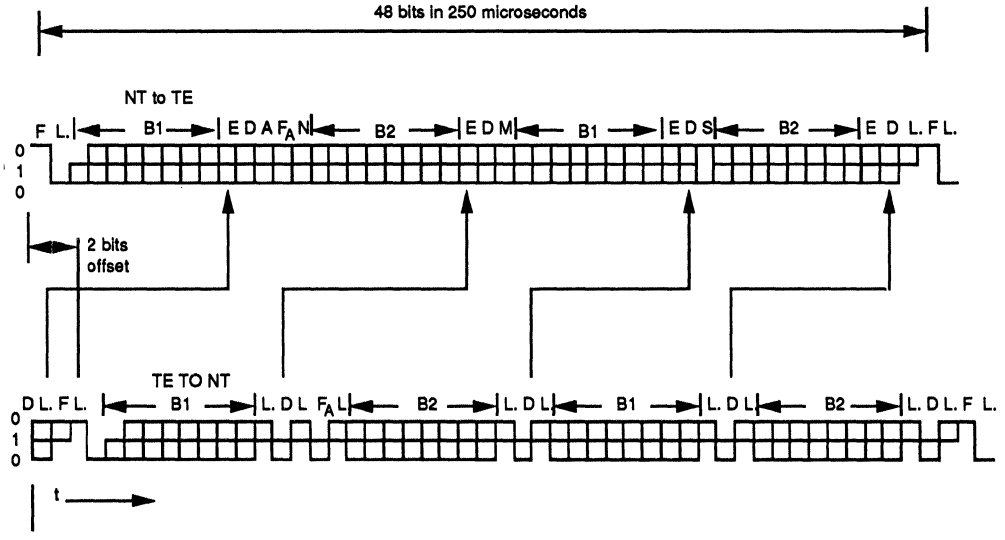


Figure 4. Modified AMI Code

One S-frame consists of 48 bits, at a nominal bit rate of 192 kbps. Thus each frame carries two octets of B1, two octets of B2, and four D-bits, according to the B1 + B2 + D structure defined for the ISDN basic access (total useful data rate: 144 kbps). Frame begin is marked using a code violation. The frame structures (from network to subscriber, and subscriber to network) are shown in Figure 5.



F = Framing bit
L = DC balancing bit
D = D-channel bit
B1 = Bit within B channel 1
B2 = Bit within B channel 2
E = D-echo-chanel bit
A = Bit used for activation
FA = Auxiliary framing bit or Q-bit
S = Reserved for future standardization
M = Multiframing bit
N = Binary Value : N = $\overline{FA}$

Note: Dots demarcate those parts of the frame that are independently DC-balanced.

Figure 5. Frame Structure at Reference Points S and T (CCITT I.430)

Digital Interface

The SBC has a digital interface with both a normal mode and a MUX mode in order to communicate with units realizing OSI layer 1 functions, such as the Am2090 ISDN Echo Cancellation (IEC), or layer 2 functions, such as the Am2075 IDEC.

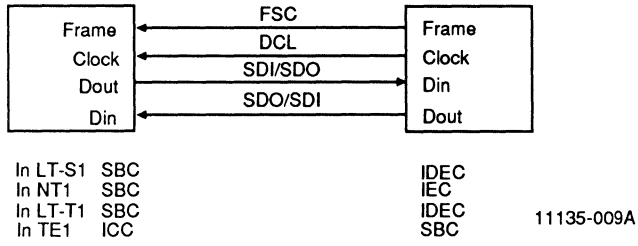


Figure 6. IOM Interface Signals

For each application, the ISDN data rate of 144 kbps is transmitted transparently via the interface. In addition, it is necessary to interchange control information for activation and deactivation of OSI layer 1 and for switching of test loops. This information is transferred using time division multiplexing with a 125 µs total frame length. The basic frame consists of four octets as shown in Figure 7.

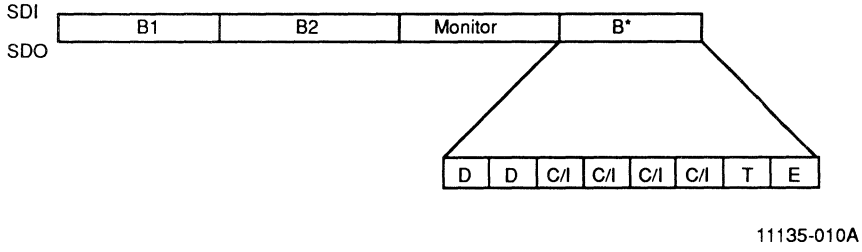


Figure 7. IOM Interface Frame Structure

1st octet B1: B channel (64 kbps), most significant bit first
2nd octet B2: B channel (64 kbps), most significant bit first
3rd octet: Monitor channel (64 kbps), most significant bit first
4th octet B*: 2 bit D channel (16 kbps)
4 bit C/I channel
T channel: not used with SBC
E bit: not used with SBC

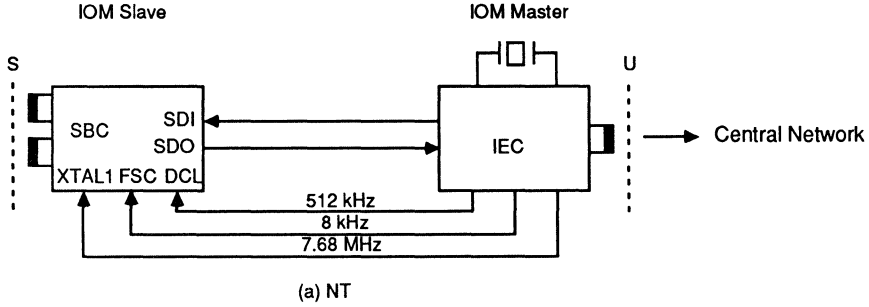
The OSI layer 1 functions are controlled by the state oriented four-bit Command/Indication codes (C/I). The codes originating from OSI layer 2 devices are called "command" primitives, and those sent by the SBC are termed responses, or "indication" primitives. For a list of the C/I codes and their uses, refer to List of Control Codes.

Individual Functions

The SBC transmits data between the IOM interface and the line interface. Data clock pulses are received from the IOM interface and applied to clock the line interface S (NT, LT-S) or vice versa (TE, LT-T). In the active state the data of the B-channels is switched through transparently. The data of the D-channel is handled in the same manner as in the NT, LT-S and LT-T application. In the TE mode, switching of the D-channel is subject to S-bus D-channel access procedure and collision detection.

Clock and frame synchronization varies from application to application. The timing relations in the different modes are summarized in Figure 8.

The frame positions have been selected to minimize the round-trip delays of the B channels, which are 125 µs for TE, NT and LT-S in normal IOM mode, max. 250 µs for LT-T in normal IOM mode and LT-S in inverted MUX mode, and max. 375 µs for LT-T in inverted MUX mode.



(a) NT

IEC = Am2090 ISDN Echo Cancellation Circuit

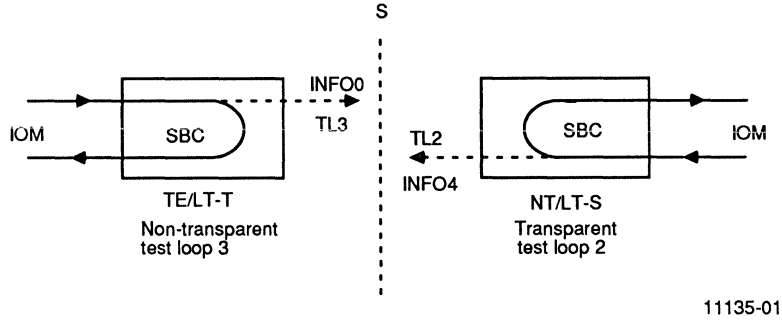
Note: Reference clock (512 kHz, duty cycle 1.2) may be used to drive; e.g., NT2 clock generator.

11135-011B

Figure 8. Clocking of SBC in Different Operating Modes

Test Functions

Two kinds of test loops may be closed in the SBC, which depend on the selected mode of operation. In both test loops, all three channels (B1, B2 and D) are looped back. In a "transparent" loop, the data is also sent forward (in addition to being looped back), whereas in a "non-transparent" loop, the forward data path is blocked. These test loops are shown in Figure 9.



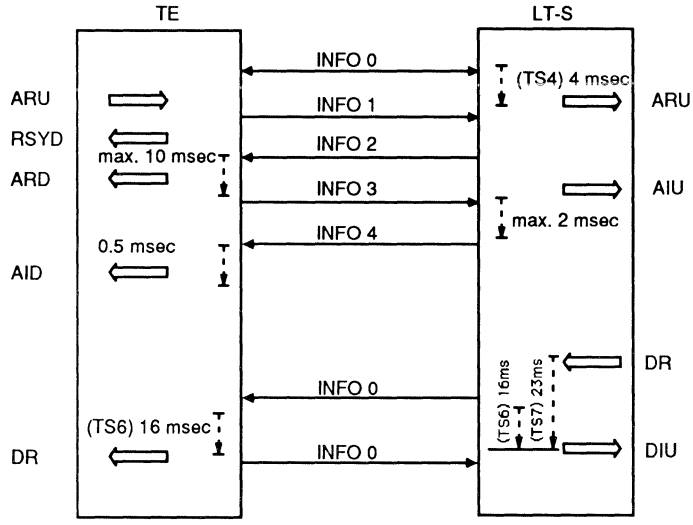
11135-012A

Figure 9. Test Loops of Am2080 SBC

Two kinds of test signals may be sent by the SBC: continuous zeros according to modified AMI coding (96 kHz repetition rate) and single AMI zeros (2 kHz repetition rate).

OPERATIONAL DESCRIPTION

The internal finite state machine of the Am2080 SBC controls the activation/deactivation procedures, switching of test loops, and transmission of special pulse patterns. An example of activation and deactivation with the respective commands and indications is shown in Figure 10.



11135-013A

Figure 10. Example of Activation/Deactivation

List of Control Codes

The exchange of control information in the C/I channel is state oriented. This means that a code in the C/I channel is repeated in every IOM frame until a change is necessary. To detect a change in a C/I code, the new code must be found in two consecutive IOM frames to be considered valid (double last look criterion). The Command/Indication codes are listed in Tables 2 through 5.

Table 2. Commands and Indications in LT-S Mode

Command (downstream)	Abbr.	Code	Remark
Deactivation request	DR	0000	(*)
Send continuous zeros	SCZ	0001	Transmission of AMI pulses at 96 kHz frequency (*)
Send single zeros	SSZ	0010	Transmission of AMI pulses at 2 kHz frequency (*)
Activate request	ARD	1000	
Activate request loop	ARL	1010	Activation request for loop 2
Deactivate indicate	DID	1111	Deactivation acknowledgment, quiescent state
Indication (upstream)			
Lost signal level	LSL	0001	No receive signal
Lost framing	RSYU	0100	Receiver is not synchronous
Activate request	ARU	1000	Info 1 received
Activate indication	AIU	1100	Synchronous receiver
Deactivate indication	DIU	1111	Timer (32 ms) expired or info 0 received (during 16 ms) after deactivation request

(*) unconditional commands

Table 3. Commands and Indications in NT Mode

Command (downstream)	Abbr.	Code	Remark
Deactivation request	DR	0000	(*)
Resynchronization of	RSYD	0100	Transmission of AMI pulses at 96 kHz frequency
Activate request	ARD	1000	Transmission of info 2
Activate request loop	ARL	1010	Transmission of info 2, switching of test loop 2
Deactivate indication	DID	1111	Deactivation acknowledgment, quiescent state
Activate indication	AID	1100	Transmission of info 4
Activate indication loop	AIL	1110	Transmission of info 4, switching of test loop
Send single zeros	SSZ	0010	Transmission of pseudoternary pulses at 2 kHz frequency (*)
Indication (upstream)			
Timing	TIM	0000	SBC requires clock pulses
Lost signal level	LSL	0001	No receive level
Lost framing	RSYU	0100	Receiver is not synchronous
Error indication	EI	0110	RST and SCZ both active
Activate request	ARU	1000	Info 1 received
Activate indication	AIU	1100	Synchronous receiver
Deactivate indication	DIU	1111	Timer (32 ms) expired or info 0 received during 16 ms after deactivation request

(*) unconditional commands

Table 4. Commands in TE/LT-T Mode

Command (upstream)	Abbr.	Code	Remark
Timing	TIM	0000	Clocking of all output clocks is required.
Reset	RS	0001	(*)
Send continuous zeros	SCZ	0100	Transmission of pseudoternary pulses at 96 kHz frequency (*)
Send single zeros	SSZ	0010	Transmission of pseudoternary pulses at 2 kHz frequency (*)
Activate request, set priority 8	AR8	1000	Activation command, set D-channel priority to 8
Activate request, set priority 10	AR10	1001	Activation command, set D-channel priority to 10
Activate request loop	ARL	1010	Activation of test loop 3 (*)
Deactivate indication	DIU	1111	Module interface can be disabled

(*) unconditional commands

Table 5. Indications in TE/LT-Mode

Indication (downstream)	Abbr.	Code	Remark
Power up	PU	0111	IOM clocking is provided
Deactivate request	DR	0000	Deactivation request by S
Slip detected	SD	0010	Wander is larger than 18 µs peak-to-peak
Disconnected	DIS	0011	Pin CON connected to GND
Error indication	EI	0110	(RST=0 ^ ENCK=0) TE or RS
Level detected	RSY	0100	Signal received, receiver not synchronous
Activate request	ARD	1000	Info 2 received
Test indication	TI	1010	Test loop activated or continuous zeros transmitted
Awake test indication	ATI	1011	Level detected during test loop
Activate indication with	AI 8	1100	Info 4 received, D-channel priority is 8 or 9 priority class 8
Activate indication with priority class 10	AI 10	1101	Info 4 received, D-channel priority is 10 or 11
Deactivate indication	DID	1111	Clocks will be disabled (in TE) quiescent state

(*) unconditional commands

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... –65 to +125°C
DC Voltage Applied to Any
Pin Relative to V_{SS} ..... –0.4 to Vcc±0.4 V
Power Dissipation ..... 1 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +5 V ±5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges

T_A = 0 to +70°C; V_{CC} = +5 V ±5%, V_{SS} = 0 V

Parameter Symbol	Parameter Descriptions	Test Conditions	Limit Values Min.	Max.	Unit
All pins except S_{X1}, S_{X2}, S_{R1}, S_{R2}					
V_{IL}	Input Low Voltage		–0.4	+0.8	V
V_{IH}	Input High Voltage		+2.0	V_{CC} + 0.4	V
V_{OL}	Output Low Voltage	I_{OL} = 2 mA		0.45	V
V_{OL1}	Output Low Voltage (SDO)	I_{OL} = 7 mA		0.45	V
V_{OL1}	Output High Voltage	I_{OH} = –400 µA		+2.4	V
V_{OH}	Output High Voltage	I_{OH} = –100 µA	V_{CC}–.5		V
I_{CC}	Power Supply Current:	Inputs at GND/V_{CC}			
	Operational	V_{CC} = 5 V		12	mA
	Power Down	No output loads		+0.8	mA
I_{LI}	Input Leakage Current	0 V < V_{IN} < V_{CC} to 0 V		±10	µA
I_{LO}	Output Leakage Current	0 V < V_{OUT} < V_{CC} to 0 V			
All pins except S_{X1}, S_{X2}					
V_X	Absolute value of	R_L = 25 ohm*		0.3	V
	output pulse amplitude	R_L = 200 ohm**	1.35	1.65	V
	(VSX2–VSX1)	R_L = 1600 ohm**	1.35	2.4	V
I_X	Transmitter output current	R_L = 200 ohm		8.25	mA
R_X	Transmitter output impedance	Inactive or during binary one	10		kohm
Z_X	Impedance	During binary zero R_L = 200 ohm	80		ohm
All pins except S_{R1}, S_{R2}					
V_{SR1}	Receiver bias voltage	I_o < 100 µA	2.4	2.6	V
V_{TR}	Receiver threshold voltage V_{SR2}–V_{SR1}	Dependent on peak level	+225	+375	mV
I_o	Output current	R_{REF} = 2.2 kohm ±1%	450	550	µA

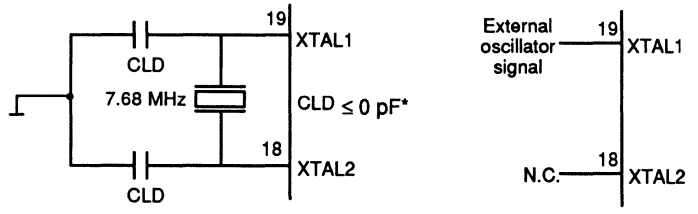
* Due to the transformer, the pulse amplitude zero to peak on S-interface line will be halved.

** Load resistance on S-interface line will be divided by four.

CAPACITANCE*

Parameter Symbol	Parameter Descriptions	Test Conditions	Limit Values Min.	Max.	Unit
All pins except S_{R1}, S_{R2}, X_{TAL1}, X_{TAL2}					
C_{IN}	Input capacitance			7	pF
$C_{I/O}$	I/O Pin capacitance			7	pF
All pins except S_{X1}, S_{X2}					
C_{OUT}	Output Pin capacitance			10	pF
All pins except S_{R1}, S_{R2}					
C_{IN}	Input capacitance			7	pF
All pins except XTAL1, XTAL2					
C_{LD}	Load capacitance			50*	pF

*Parameters are not "Tested."



Crystal oscillator mode

Driving from external source

*For the version up to and including A4, this value should not exceed 20 pF.

11135-014A

Figure 11. Recommended Oscillator Circuits

Table 6. Output Stages

Application	Operation of IOM Interface	M2	M1	M0	DCL	FSC	CP	X2	X1	X0	SDO
TE	Inverted Mode	0	0	0	Push/pull	Push/pull	Push/pull	Push/pull	Push/pull	Push/pull	Push/pull
TE	Inverted Mode	0	0	1	Push/pull	Push/pull	Push/pull	Push/pull	Push/pull	Push/pull	Push/pull
TE	Normal Mode	0	1	0	Push/pull	Push/pull	Push/pull	Push/pull	Push/pull		Push/pull
LT-T	MUX Mode Inverted	0	1	1			Push/pull				Open drain
LT-T	Normal Mode	0	1	1			Push/pull				Push/pull
NT	Normal Mode	1	1	1						*Open drain	*Open drain
LT-S	MUX Mode Inverted	1	0	0							Open drain
LT-S	Normal Mode	1	1	0					Push/pull		Push/pull
LT-S	Normal Mode	1	1	0				Push/pull	Push/pull		

*With integrated pull-up

Table 7. SBC Clock Signals

Application	Operation of IOM Interface	M2	M1	M0	DCL	FSC	CP	X2	X1	X0
TE	inverted mode	0	0	0	O:512 kHz* 1:2	O:8 kHz* 63:1	O:1536 kHz* 3:2	O:2560 kHz 1:2	O:3840 kHz 1:1	
TE	inverted mode	0	0	1	O:512 kHz* 1:2	O:8 kHz* 63:1	O:1536 kHz* 3:2	O:1280 kHz 1:2	O:3840 kHz 1:1	
TE	normal mode	0	1	0	O:512 kHz* 2:1	O:8 kHz* 1:1	O:1536 kHz* 3:2		O:3840 kHz 1:1	
LT-T	MUX mode inverted	0	1	1	I:4096 kHz	I:8 kHz	O:512 kHz 2:1			
LT-T	normal mode	0	1	1	I:512 kHz	I:8 kHz	O:512 kHz* 2:1			
NT	normal mode	1	1	1	I:512 kHz	I:8 kHz				
LT-S	MUX mode inverted	1	0	0	I:4096 kHz	I:8 kHz				
LT-S	normal mode	1	1	0	I:512 kHz	I:8 kHz			O:7680 kHz 1:1	I:fixed at 0
LT-S	normal mode	1	1	0	I:512 kHz	I:8 kHz		O:192 kHz* 1:1	O:7680 kHz 1:1	I:fixed at 1

*Synchronous to receive "S" line

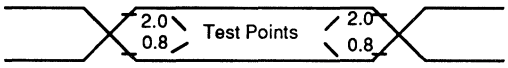
Input and Output Pin Configurations

In TE, LT-T and LT-S IOM normal modes, an integrated pull-up resistor is connected to SDI. For output pin configurations, see capacitance.

SWITCHING CHARACTERISTICS

T_A = 0 to +70°C, V_{CC} = 5 V + 5%

The AC testing input/output waveform is shown below.



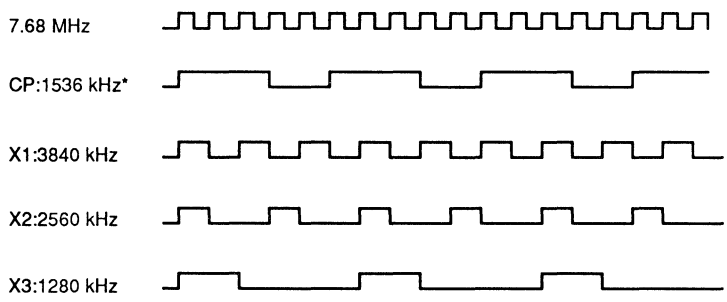
11135-015A

Clock Timing

The clocks in the different operating modes are summarized in Table 6, with the respective duty ratios. Clock CP is phase-locked to the receive S signal, and is derived using the internal DPLL and the 7.68 MHz +100 ppm crystal (TE and LT-T).

A phase tracking of CP with respect to "S" is performed once in 250 µs. As a consequence of this tracking, the high state of CP may be either reduced or extended by one 7.68 MHz period (CP duty ratio 2:2 or 4:2 instead of 3:2) once every 250 µs. Since DCL and FSC are derived from CP (TE mode), the high state (FSC) or the high or low state (DCL) may likewise be reduced or extended by the same amount once every 250 µs*.

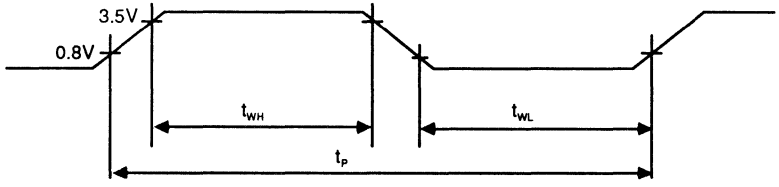
*The phase adjustment may take place either in the sixth, seventh, or eighth CP cycle counting from the beginning of an IOM frame in TE.



11135-016A

*Synchronous to receive S, see 5.6.1. Duty ratio 3:2 normally.

Figure 12. Phase Relationships of Auxiliary Clocks



11135-017A

Figure 13. Definition of Clock Period and Width

Tables 8 through 11 give the timing characteristics of the clocks.

Table 8. DCL Timing Characteristics

Symbol	Description	Min	Typ	Max	Unit	Conditions
t_P Output	(TE) 512 kHz	1822	1953	2084	ns	osc ± 100 ppm
t_{WH} Output	(TE) 512 kHz 2:1	1121	1302	1483	ns	osc ± 100 ppm
t_{WL} Output	(TE) 512 kHz 2:1	470	65	832	ns	osc ± 100 ppm
t_{WH} Output	(TE) 512 kHz 1:2	470	651	832	ns	osc ± 100 ppm
t_{WL} Output	(TE) 512 kHz 1:2	1121	1302	1483	ns	osc ± 100 ppm
t_{WH} Input	(NT, LT-S, LT-T)	90			ns	
t_{WL} Input	(NT, LT-S ,LT-T)	90			ns	

Table 9. CP Timing Characteristics

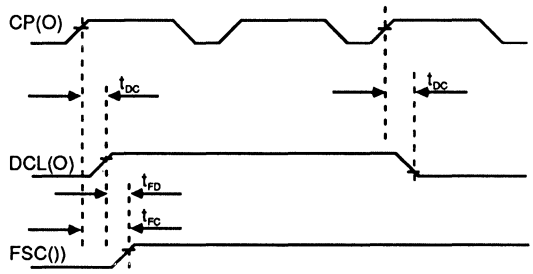
Symbol	Description	Min	Typ	Max	Unit	Conditions
t_P Output	(TE) 1536 kHz	520	651	782	ns	osc ± 100ppm
t_{WH} Output	(TE) 1536 kHz 2:1	240	391	541	ns	osc ± 100 ppm
t_{WL} Output	(TE) 1536 kHz 2:1	240	260	231	ns	osc ± 100 ppm
t_R, t_F Output	(TE, LT-T)			20	ns	C_L = 100 pF
			10			C_L = 50 pF
t_P Output	(LT-T) 512 kHz	1322	1953	2034	ns	osc ± 100 ppm
t_{WH} Input	(LT-T) 512 kHz	1121	1302	1483	ns	osc ± 100 ppm
t_{WL} Input	(LT-T) 512 kHz	470	651	832	ns	osc ± 100 ppm

Table 10. X1 Timing Characteristics

Symbol	Description	Min	Typ	Max	Unit	Conditions
t_P Output	(TE) 3840 kHz	–100 ppm	260	+100 ppm	ns	osc ± 100 ppm
t_{WH} Output	(TE) 3840 kHz	120	130	140	ns	osc ± 100 ppm
t_{WL} Output	(TE) 3840 kHz	120	130	140	ns	osc ± 100 ppm

Table 11. X2 Timing Characteristics

Symbol	Description	Min	Typ	Max	Unit	Conditions
t_P Output	(TE) 2560 kHz	–100 ppm	391	+100 ppm	ns	osc ± 100 ppm
t_{WH} Output	(TE) 2560 kHz	110	130	150	ns	osc ± 100 ppm
t_{WL} Output	(TE) 2560 kHz	250	260	270	ns	osc ± 100 ppm
t_P Output	(TE) 1280 kHz	–100 ppm	781	+100 ppm	ns	osc ± 100 ppm
t_{WH} Output	(TE) 1280 kHz	250	260	270	ns	osc ± 100 ppm
t_{WL} Output	(TE) 1280 kHz	511	521	531	ns	osc ± 100 ppm

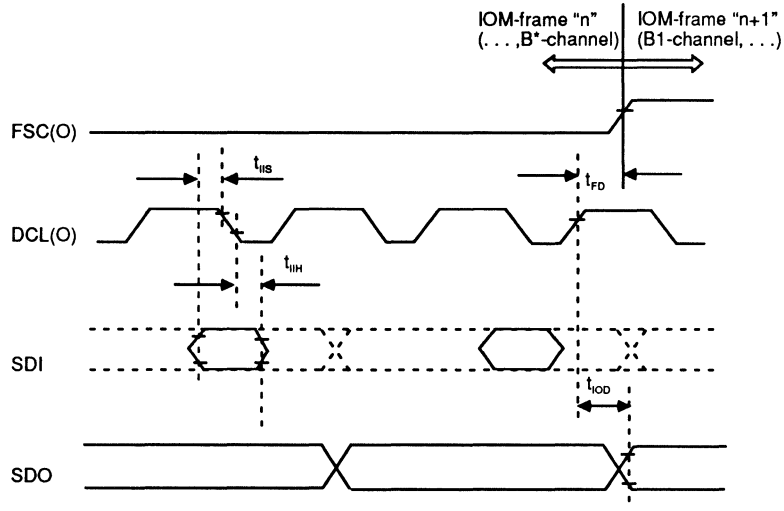


11135-018A

Figure 14. CP, DCL, and FSC Relationships in IOM Master Mode

Table 12. IOM Master Mode

Symbol	Description	Min	Max	Unit
t_{DC}	Clock delay CP-DCL	0	50	ns
t_{FC}	Clock delay CP-FSC	0	50	ns
t_{FD}	Delay DCL-FSC	–20	20	ns



11135-019A

Figure 15. Master Mode (TE)

Table 13. Master Mode (TE)

Symbol	Parameter	Min	Max	Unit	Conditions
t_{FD}	Frame sync delay	–20	20	ns	C_L = 100 pF
t_{IOD}	IOM output data delay		200	ns	C_L = 100 pF
t_{IIS}	IOM input data setup	20		ns	
t_{IIH}	IOM input data hold	50		ns	

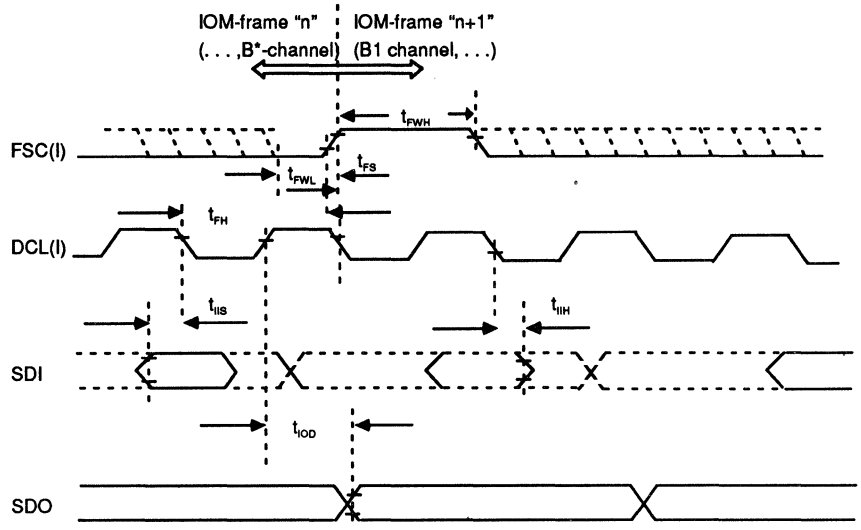
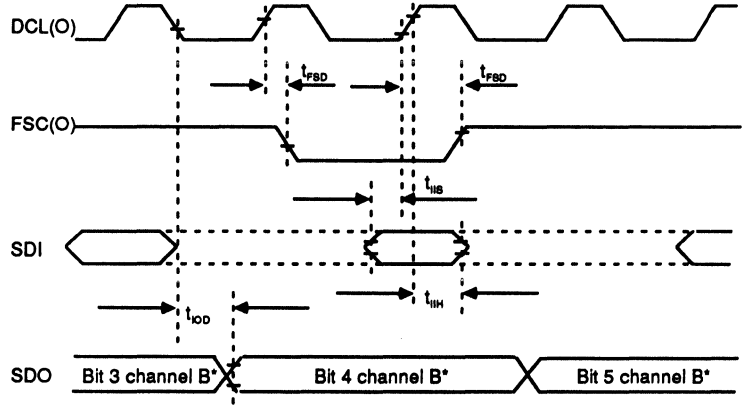


Figure 16. Slave Mode (NT, LT-S, LT-T)

Table 14. Slave Mode (NT, LT-S, LT-T)

Symbol	Parameter	Min	Max	Unit
t_{DH}	Frame sync hold	30		ns
t_{FS}	Frame sync setup	50		ns
t_{FWH}	Frame sync High	40		ns
t_{FWL}	Frame sync Low	2150		ns
t_{IOD}	IOM output data delay		200	ns*
t_{IIS}	IOM input data setup	20		ns
t_{IIH}	IOM input data hold	50		ns

* For push-pull output. For open drain output with integrated pull-up resistor, the maximum value is 900 ns.



11135-021A

Figure 17. Inverted Mode

Table 15. Inverted Mode

Symbol	Parameter	Min	Max	Unit	Conditions
t_{FSD}	Frame sync delay	−20	20	ns	C_L = 100 pF
t_{IOD}	IOM output data delay	200		ns	C_L = 100 pF
t_{IIS}	IOM input data setup	20		ns	
t_{IIH}	IOM input data hold	50		ns	

2

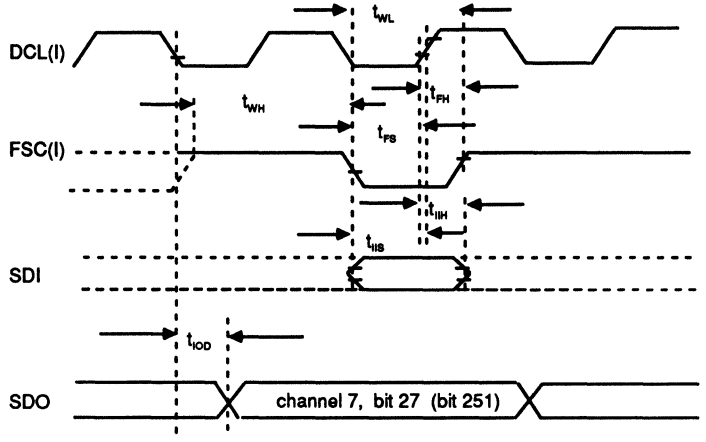


Figure 18. Inverted MUX Mode

Table 16. Inverted MUX Mode

Symbol	Parameter	Min	Max	Unit	Conditions
t_{FH}	Frame sync hold	50		ns	
t_{FS}	Frame sync setup	20		ns	
t_{FWH}	Frame sync High	124.8		µs	
t_{FWL}	Frame sync Low	70	200	ns	
t_{IOD}	IOM output data delay		200	ns	C_L = 150 pF; I_{OL} = 7 mA
t_{IIS}	IOM input data setup	20		ns	
t_{IIH}	IOM output data hold	50		ns	

Timing of Special Function Pins

Table 17. $\overline{RST}$ Timing

Symbol	Parameter	Min	Max	Unit
t_{WL}	Length of Low state	1		µs

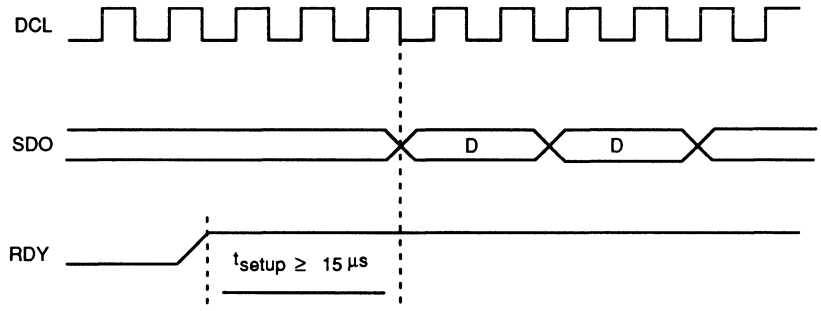


Figure 19. RDY Timing (X0 in TE Inverted Mode)

Table 18. RDY Timing

Symbol	Parameter	Min	Max	Unit
t_{WL}	Length of Low state	360		µs
t_{WH}	Length of High state	60		µs

DE

The timing for DE (pin X0, NT mode) is given by Figure 21 for the case of two S interfaces having a minimum frame delay and a maximum frame delay, respectively. The corresponding star configuration is shown in Figure 20. DE timing is shown in Figure 21.

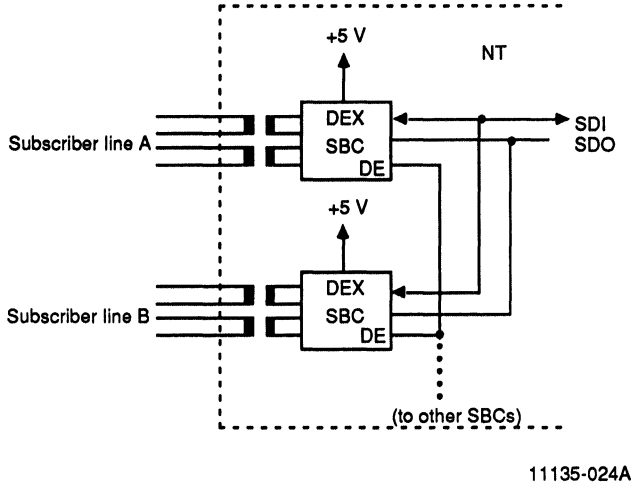
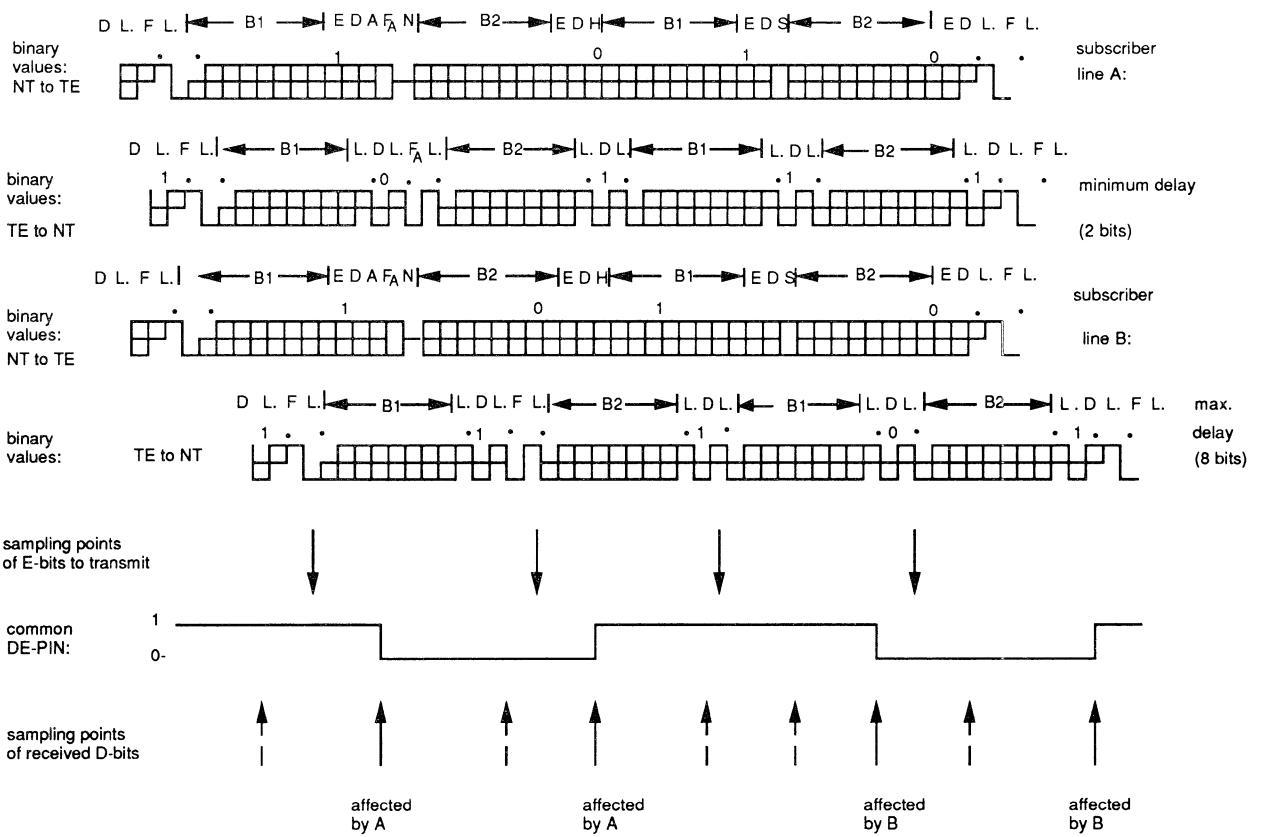


Figure 20. Star Configuration in NT

ECHO

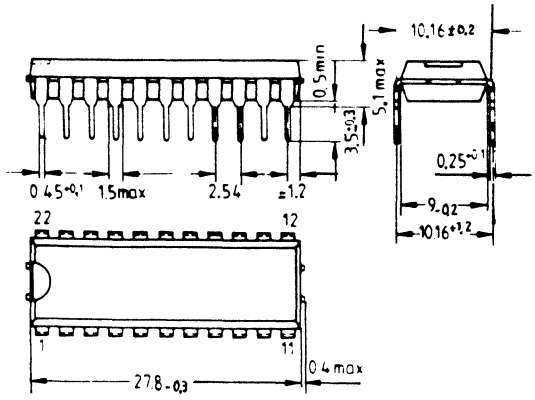
The timing of the ECHO output (pin X2, TE mode) is identical with that of output SDO; however, the signal is "1" everywhere except in bit positions 24 and 25 of IOM frame, where it is equal to the E-bits received from the "S" interface.



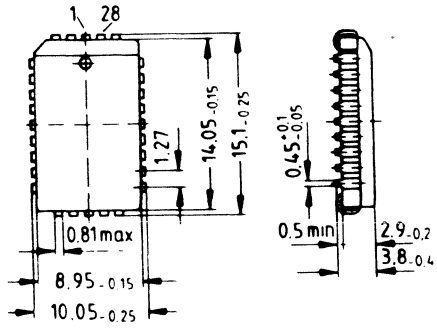
CONDITION: All transmit frames NT→ TE are in phase.

Figure 21. Timing of DE

PHYSICAL DIMENSIONS
PD022



PLR028



Note: Physical dimensions are given in mm.

Am2080

SBC-A7 S-Bus Interface Circuit

Advanced Micro Devices

According to CCITT I.430 the electrical characteristics of the S/T interface transmitter are to fulfill the following requirements:[1]

- The output impedance, when the transmitter is inactive or transmitting a binary '1,' should exceed 2500 ohms. Note that this also applies to TEs with local power sources when the local power is switched off, although the TE is connected to an activated S-bus.
- The output impedance, when the transmitter is transmitting a binary '0,' should be $\geq$ 20 ohms. Note that this also applies in the case of a 400 ohm load when the transmitter reaches a current- or voltage-limiting state.
- Pulse shape and amplitude shall be in accordance with the given pulse masks. Note that in TE applications the effective test load for the transmitter not only consists of twice the terminating resistance (50 ohms) but also of the series resistances of other external components such as the transformer and the cord (maximum length 7 m).

The transmitter circuitry of the Am2080 SBC A7 meets these requirements in full.

The transmitter essentially is a current-limited voltage source, delivering nominally 1.5 V to the 2:1 transformer. The high output impedance when transmitting a binary '1,' including when the power supply is switched off, is guaranteed by the circuit design.

When transmitting a binary '0,' the output impedance (Z_{OUT}) has to be in the range 20–25 ohms in order that the pulse mask be fulfilled. Since the internal output impedance of the transmitter is negligible for a binary '0,' the impedance is realized by the total sum of external resistances (Figure 1); for example:

Resistance of cord (TE only) (R_{CORD})	4-7 Ω
Copper resistance of transformer (R_{COPPER})	1-3 Ω
Resistances for overvoltage protection of the transmitter, transformed to the primary (line) side with 4:1 ($R_{OVERVOLT}$)	15 Ω

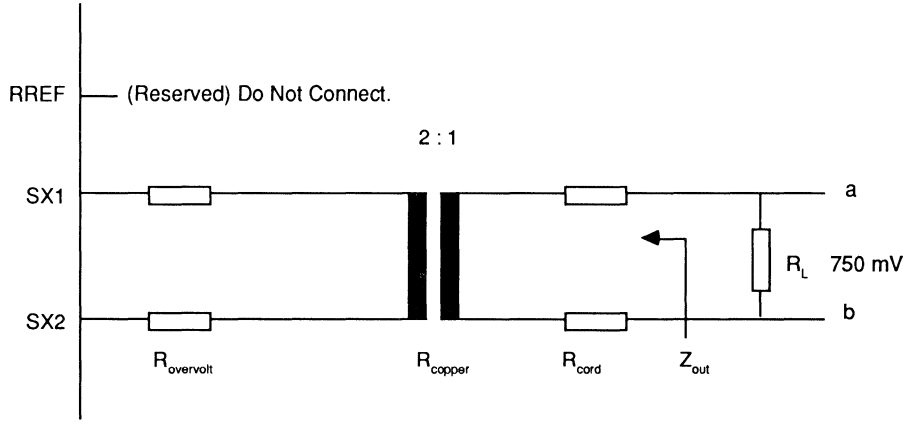


Figure 1. Transmitter Environment Model of the SBC A7

[1]ISDN User-Network Interfaces: Layer 1 Recommendations.

CCITT Recommendations of the Series I—Volume III Fascicle III.5 Integrated Services Digital Network (ISDN). VIIIth Plenary Assembly Malaga-Torremolinos, 8–10 October 1984.

Publication #	Rev.	Amendment
11135	B	/2

Issue Date: July 1989

DC CHARACTERISTICS

TA = 0 to 70° C; V_{DD} = 5 V ± 5%; V_{SS} = 0 V

Symbol	Parameter	Limit Values Min.	Max.	Units	Test Condition
VX	Absolute value of output pulse amplitude | (VSX1 - VSX2) |	2.03 2.10	2.31 2.39	V V	RL = 50 Ω * RL = 400 Ω *
IX	Transmitter output current	7.5	13.4	mA	RL = 5.6 Ω *

* Due to the transformer, the load resistance seen by the Am2080 is four times R_L.

SBC-A8

1. The 50 Ω pulse mask conforms to CCITT I.430 (no overshoot).
2. On activation of "SCZ" or "SSZ" (via Pins X2 or CP) the SBC pulls SDO to zero in order to get clocks from the IOM™ master device.
3. In the LT-T and LT-S MUX mode, timing signals according to IOM Rev. 2 Specification are accepted. The SBC recognizes automatically whether MUX mode inverted or IOM Rev. 2 mode timing signals are applied.

Advance Information

Am2081

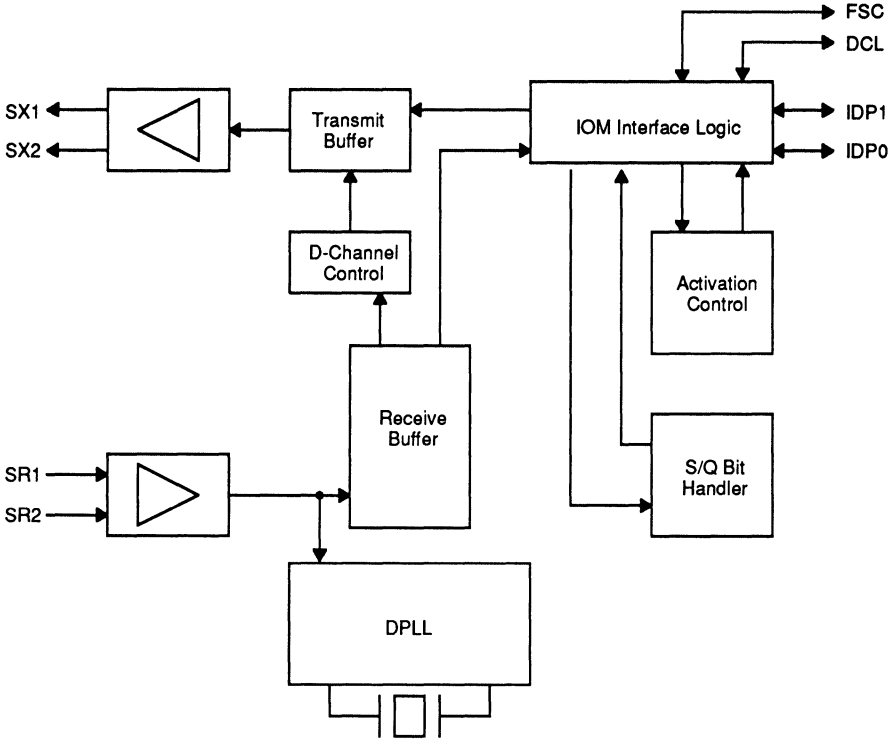
S/T Bus Interface Circuit Extended (SBCX)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Full duplex 2B + D S/T interface transceiver according to CCITT I.430
- Adaptive equalizer
- Receive timing recovery
- Built-in wake-up unit for activation from power-down state
- Conversion of the frame structure between the S/T interface and IOM™ Rev.2 Interface
- Activation and deactivation procedures according to CCITT I.430
- D-channel access control, also in trunk application
- Access to S and Q bits of S/T interface
- Automatic handling of S and Q bit messages
- Software controlled maintenance interface (I/O ports)
- Frame alignment with absorption of phase wander in NT2 network side applications
- Switching of test loops
- Several operating modes
- Advanced CMOS technology
- Low power consumption:
  - standby less than 6 mW
  - active max 80 mW

BLOCK DIAGRAM



Publication # 11150 Rev. B Amendment /0
Issue Date: March 1989

GENERAL DESCRIPTION

The S/T Interface Circuit Extended (SBCX) Am2081 implements the four-wire S/T interface used to link voice/data ISDN terminals, network termination (Central Office and PABX applications), and PABX trunk lines to Central Office. Through selection of operating modes, the device may be employed in all types of applications involving an S/T interface. Two or more Am2081 SBCX can be used to build a point-to-point, passive bus, extended passive bus or star configuration.

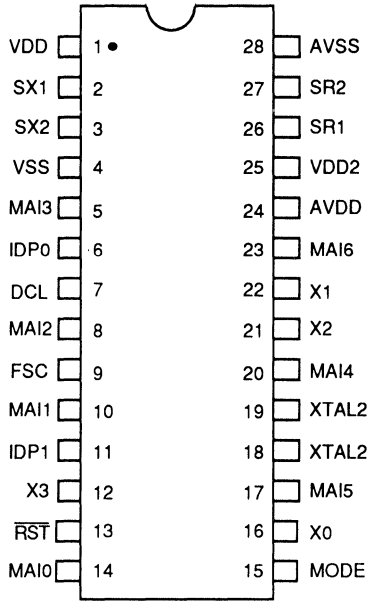
The Am2081 SBCX provides the electrical and functional link between the analog S/T interface according to CCITT recommendation I.430 and T1D1 Basic User Network Interface Specification, respectively, and the ISDN Oriented Modular (IOM) interface Rev. 2.

The Am2081 SBCX exceeds both the electrical and functional requirements of the S/T interface in order to provide high flexibility to the user with respect of S/T interface wiring configuration and implementation of layer-1 maintenance functions. By provision of some additional features at the IOM Rev. 2 interface the user is able to combine the SBCX with other IOM Rev. 2 devices in various configurations.

The Am2081 SBCX is a 28-pin CMOS device offered in both DIP and PLCC packages. It operates from a single 5 V supply and features a power-down state with very low power consumption.

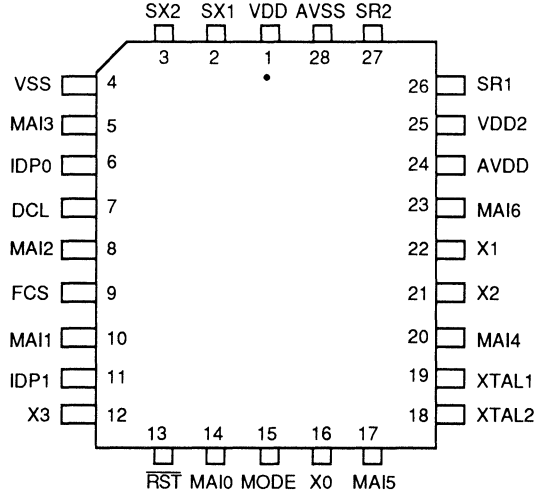
CONNECTION DIAGRAMS
Top View

28-Pin DIP



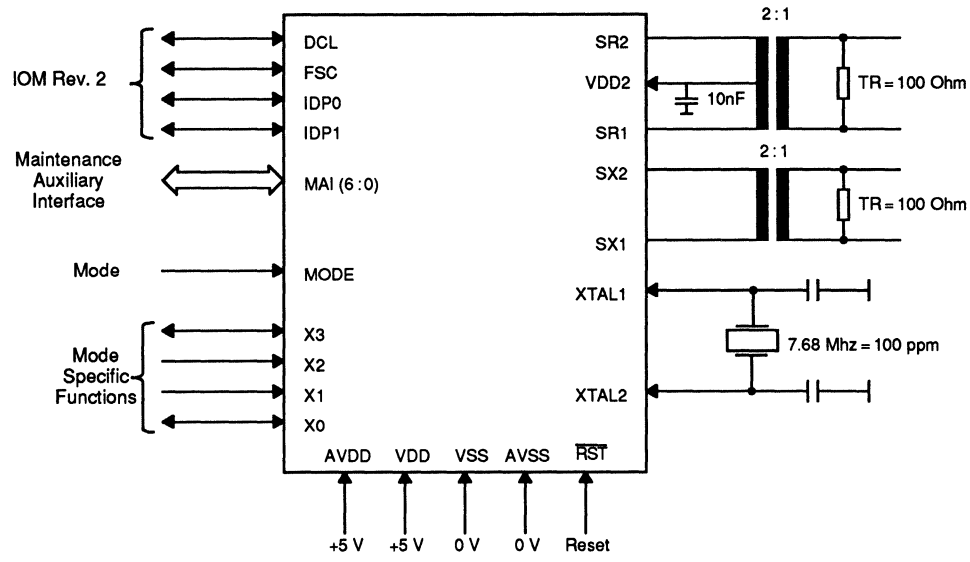
11134-002B

28-Pin PLCC



Note: Pin 1 is marked for orientation.

LOGIC SYMBOL



ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing

AM2081 J C B

e. OPTIONAL PROCESSING
Blank = Standard Processing
B = Burn-in

d. TEMPERATURE RANGE
C = Commercial (0 to +70°C)

c. PACKAGE TYPE
J = 28-Pin Plastic Leaded Chip Carrier (PL 028)
P = 28-Pin Plastic DIP (PD 028)

b. SPEED OPTION
Not Applicable

a. DEVICE NUMBER/DESCRIPTION
Am2081
S/T Bus Interface Circuit Extended (SBCX)

Valid Combinations	
AM2081	JC, JCB, PC, PCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Am2091

ISDN Echo Cancellation Circuit (IEC-Q)

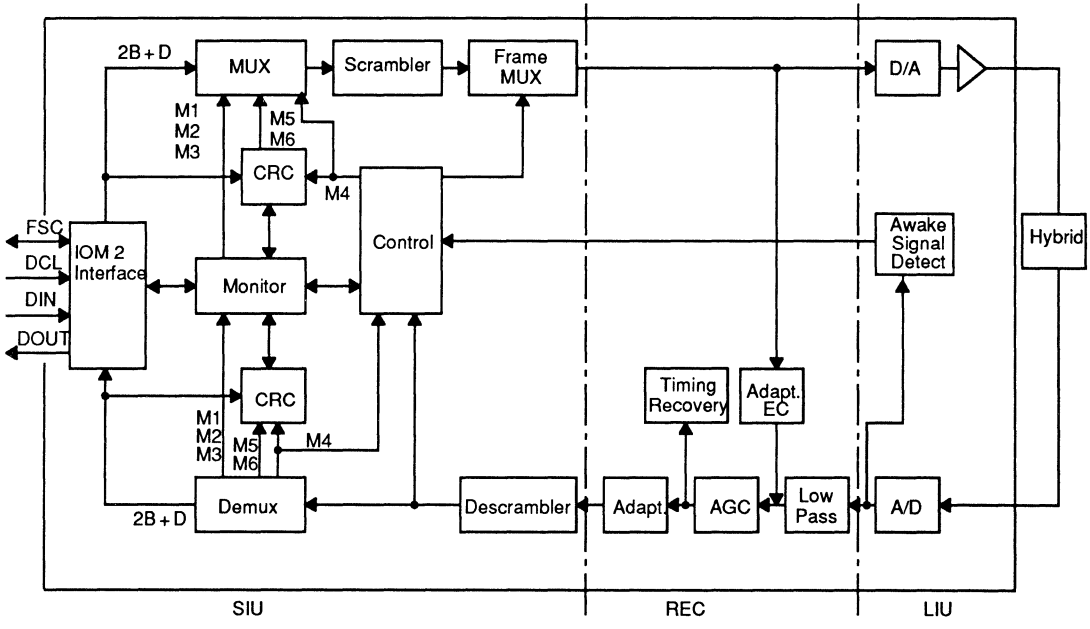
Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Full duplex data transmission and reception at the
  - –U-Reference point according to the layer 1
  - –Specification of the American Standard of Telecommunications:
  - –144 kbps user bit rate over a two wire subscriber loop
  - –2B1Q block code (2 binary, 1 quaternary)
  - –4 kbps maintenance channel for transmission of data loop back commands and detected transmission errors
  - –monitoring of transmission-errors
  - –operating at telephone loop plant LOOP #1 up to LOOP #15 as defined by American National Standard
- Transposition of quaternary to binary data and vice versa (coding, decoding, scrambling, descrambling, phase adaptation)
- Built-in wake up unit for activation from power down state
- Activation and deactivation procedure according to T1D1 layer 1 Specification and CCITT I.430
- Adaptation of internal interfaces to the current signal direction by programmable operation modes:
  - –LT: Line termination in public or private exchange
  - –TE: Terminal mode
  - –NT: Network termination connected to SBCX
  - –NT-PABX: Trunk module (TDM)

2

BLOCK DIAGRAM



Publication # 11892 Rev. B Amendment /0
Issue Date: March 1989

DISTINCTIVE CHARACTERISTICS (continued)

- Adaptive echo cancellation
- Adaptive equalization
- Clock recovery (frame and bit synchronization) in all applications
- Optimized for working in conjunction with SBCX, EPIC™ and IDEC™ Telecom ICs via IOM™2 interface
- Data speed conversion between the U-Reference Point and the IOM frame
- Handling of the Commands and Indications contained in the IOM2 C/I channel for deactivation, activation, supervision of power supply unit and equipment for testing
- Data availability via monitor-channel:

  CRC transmission error

  Measurement value of the loop current

  Echo Canceller Coefficients and Status values, which can be used to indicate the state of the loop
- Switching test loops
- Generation of synchronized 7.68 MHz clock for SBCX in NT mode
- Low power consumption:
  - standby max 30 mW
  - active max 300mW
- 40-pin DIP and 44-pin PLCC package

GENERAL DESCRIPTION

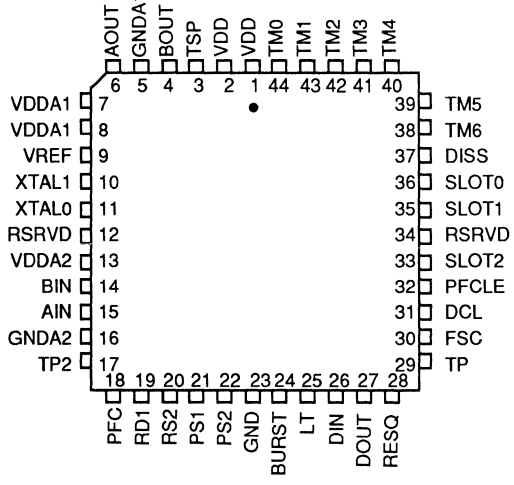
The Am2091 ISDN Echo Cancellation Circuit (IEC-Q) is an advanced CMOS single chip transceiver for ISDN Basic Access Digital Subscriber Loops with 2B1Q line code.

According to the Layer 1 Specification of the American National Standard of Telecommunication control and algorithmic requirements are implemented for a 144 kbps full duplex data transmission.

Together with the flexible IOM2 interface the IEC-Q is fully compatible with the Am2081 (SBCX). Am2055 (EPIC), and Am2075 (IDEC) devices.

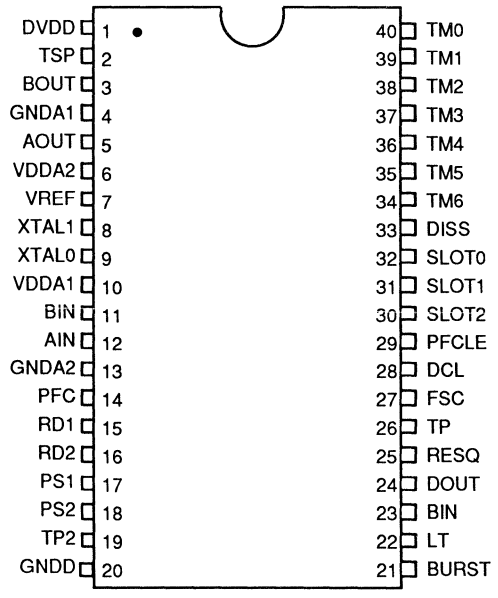
CONNECTION DIAGRAM
Top View

44-Pin PLCC



Note: Pin 1 is marked for orientation.

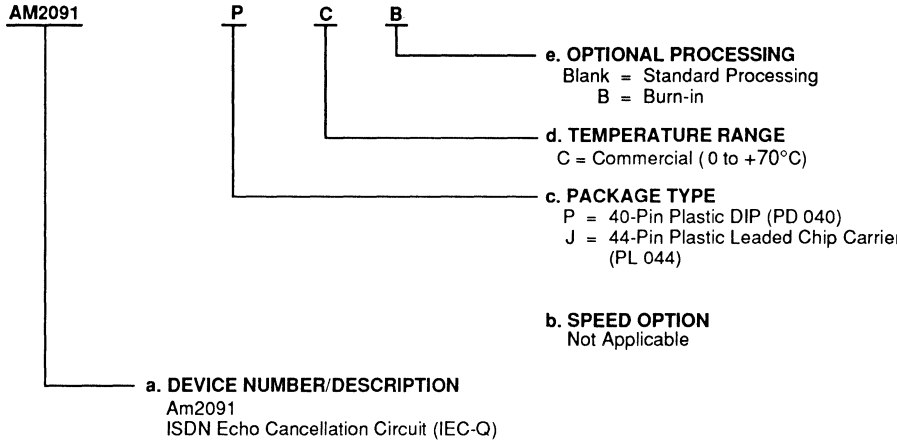
40-Pin DIP



ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM2091	PC, PCB, JC, JCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

2

Preliminary

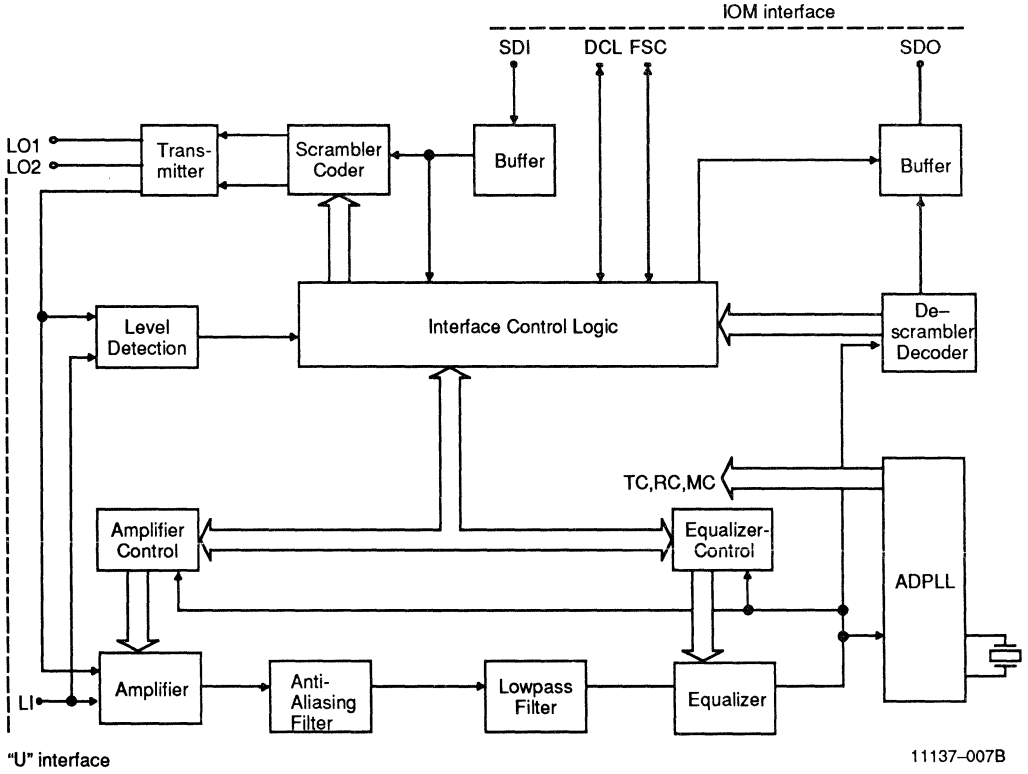
Am2095
Burst Transceiver Circuit (IBC)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Half-duplex burst mode 2-wire "U" interface transceiver
- 144 kbps user bit rate (2B + D)
- 384 kHz line clock rate
- IOM compatible
- Clock and frame recovery
- Adaptive line equalization and amplification at receiver
- Implementation of activation/deactivation procedures
- Mode configurable to function at both ends of the line
- Built-in wake-up function for activation from power-down state
- Switching of test loops
- Typical length of loop: up to 3.5 km with 0.6 mm diameter wire
- Advanced CMOS technology
- Low power consumption:
  - 6 mW power down
  - 80 mW power up (maximum)

BLOCK DIAGRAM



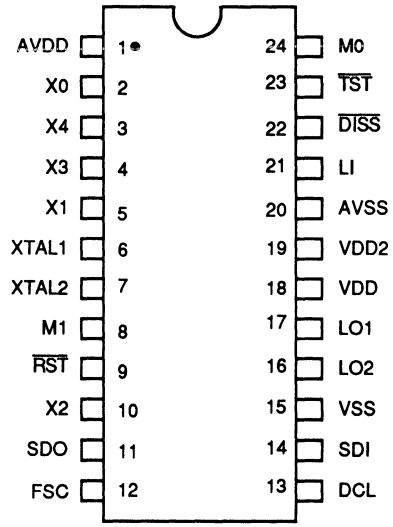
Publication # 11137 Rev. B Amendment /0
Issue Date: April 1989

GENERAL DESCRIPTION

The Am2095 ISDN burst transceiver circuit (IBC) is a full duplex transceiver for the 2-wire transmission line (CCITT U-reference point). Full duplex transmission is achieved using a time compression multiplex (ping-pong) technique. The device links the 2-wire transmission line to the ISDN Oriented Modular (IOM™) interface and to the AMD family of ISDN devices. The device manages layer 1 of the interface protocol and can communicate with other layer 1 or layer 2 devices over the IOM interface.

CONNECTION DIAGRAM
Top View

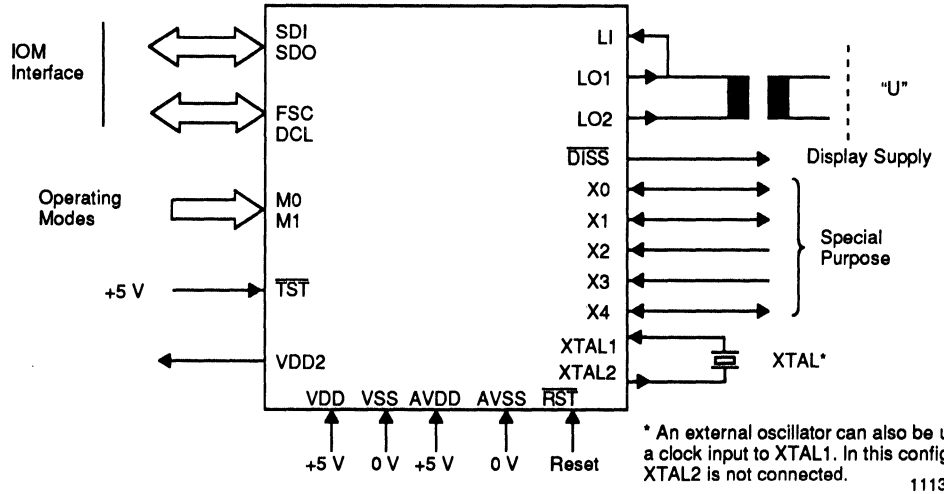
24-Pin DIP



Note: Pin 1 is marked for orientation.

11137–002B

LOGIC SYMBOL



* An external oscillator can also be used as a clock input to XTAL1. In this configuration XTAL2 is not connected.

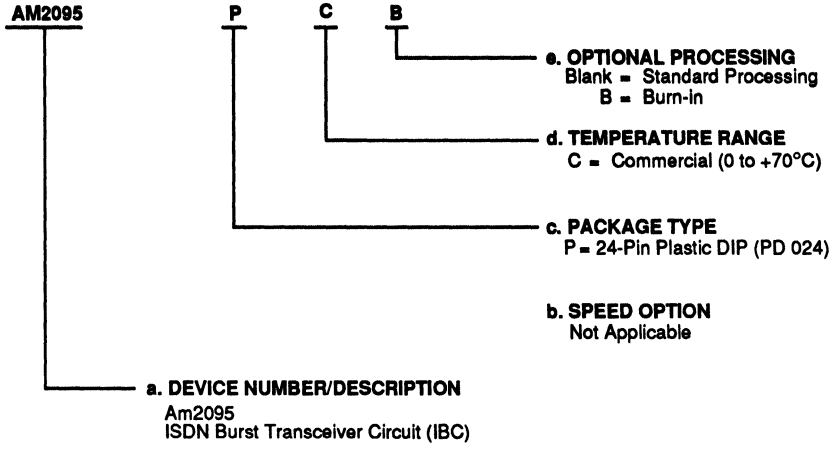
11137–003B

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM2095	PC, PCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military-grade products.

PIN DESCRIPTION

LO1
Line transmitter (Output)
Output 1; "U" interface.

LO2
Line transmitter (Output)
Output 2; "U" interface.

LI
Line receiver (Input)
"U" interface (output)

SDO
Serial Data Out (Output)
IOM-interface.

SDI
Serial Data In (Input)
IOM-interface.

DCL
Serial Data Clock (Input/Output)
IOM-interface.

FSC
Frame Sync. (Input/Output)
IOM-interface.

M0-M1
Operating mode setup pin (Input)

X0-X4
Multifunctional pins (Input/Output)
Mode specific functions. (See Table 3.)

XTAL1
External crystal or external oscillator (Input)

XTAL2
External crystal connection (Output)
N.C. when external oscillator is used.

TST
Device test pin (Input)
Not for general use; tie high always.

$\overline{DISS}$
Disable supply (Output)

$\overline{RST}$
Hardware reset pin (Input)
Active low.

VDD
Digital power supply (Input)
5 V ±5%.

VSS
Digital ground (Input)

AVDD
Analog power supply (Input)
5 V ±5%.

AVSS
Analog ground (Input)

VDD2
(Output)
2.5 output. Connected to VDD via 10 nF capacitor; connected to AVSS via 10 nF.

OPERATIONAL DESCRIPTION

In essence the Am2095 IBC is a layer 1 transceiver between the U-reference point and the IOM-interface. All higher layers are passed transparently through the transceiver. In NT-mode, no higher layer functions need to be performed. However in TE and LT modes this is not the case. A layer 2 device must be connected to the IBC to handle layer 2 functions. A microcontroller connected to the IDEC handles all higher layers. In this operational description a brief outline of how the device operates is presented with special emphasis on the latter, more complex, case.

Reset

The reset state is entered unconditionally after either a software RES command is received or after the $\overline{RST}$ pin is set to "0".

In the LT mode all outputs are high impedance during $\overline{RST}$ = 0. However the IOM clocks must always be enabled. After reset the device must be deactivated (using a DR command) before it can be activated.

In the TE/NT modes $\overline{RST}$ acts with $\overline{ENCK}$ to give the following states. $\overline{ENCK}$ is the name of the multifunctional pin x3 when the IBC is in the TE/NT mode.

RST	ENCK	
0	1	RESET; Outputs high impedance
0	0	RESET; Outputs low impedance Clocks are running
1	1	NORMAL OPERATION
1	0	NORMAL OPERATION; IOM clocks not disabled in "deactivated" state

Power consumption is under 6mW in the deactivated state (also called the "Power-Down" state). The IOM clocks may be enabled or disabled (see above). The device is simply waiting to be activated (see Activation/ Deactivation).

IOM Timing

The two possible IOM clocking states:

- idle state: clocks disabled (clock lines low) data lines high
- clocked state: clocks enabled (stand-by)

In the LT mode only the clocked state is possible, in the TE/NT mode both are possible. Selection of one or the other state is governed by the internal state machine.

Activation/Deactivation

During normal operation (that is, no test modes implemented, and so on) the IBC is continually activated and deactivated as the link path is required. In a deactivated state the IBC activation procedure may be initialized by receiving either a wake signal on the line (U-reference point) or receiving an activation request over the IOM-interface.

The progress of the activation procedure from this point is governed by an internal state machine within the IBC. Each successive state is entered on receiving a particular info ("U" interface) or command (IOM-interface) and results in a particular info ("U" interface) or indication (IOM-interface) being transmitted. Other procedures (such as deactivation, switching of test loops, and transmission of special pulse patterns) are also governed by the state machine. Furthermore the state machine is mode dependent.

Figure 1 illustrates such an activation and deactivation procedure between an IBC in LT-mode and an IBC in TE-mode over the "U" interface line. In this case, activation was initiated by an ARN request at the terminal and deactivation by a DR command at the LT. Activation could also have been initialized at the LT using an ARN request. A complete list of Commands and Indications (with their respective codes) is given in Table 1 and Table 2.

The exchange of infos along the "U" interface is of no real consequence to the user except in two cases:

- In the deactivated state, info 0 is exchanged between TE and LT. Info 0 effectively means there is no signal sent on the line in either direction.
- In the activated state, info 3 upstream and info 4 downstream are continually exchanged. Both info 3 and info 4 are effectively normal "U" interface data frames, exchanged in the normal burst mode.

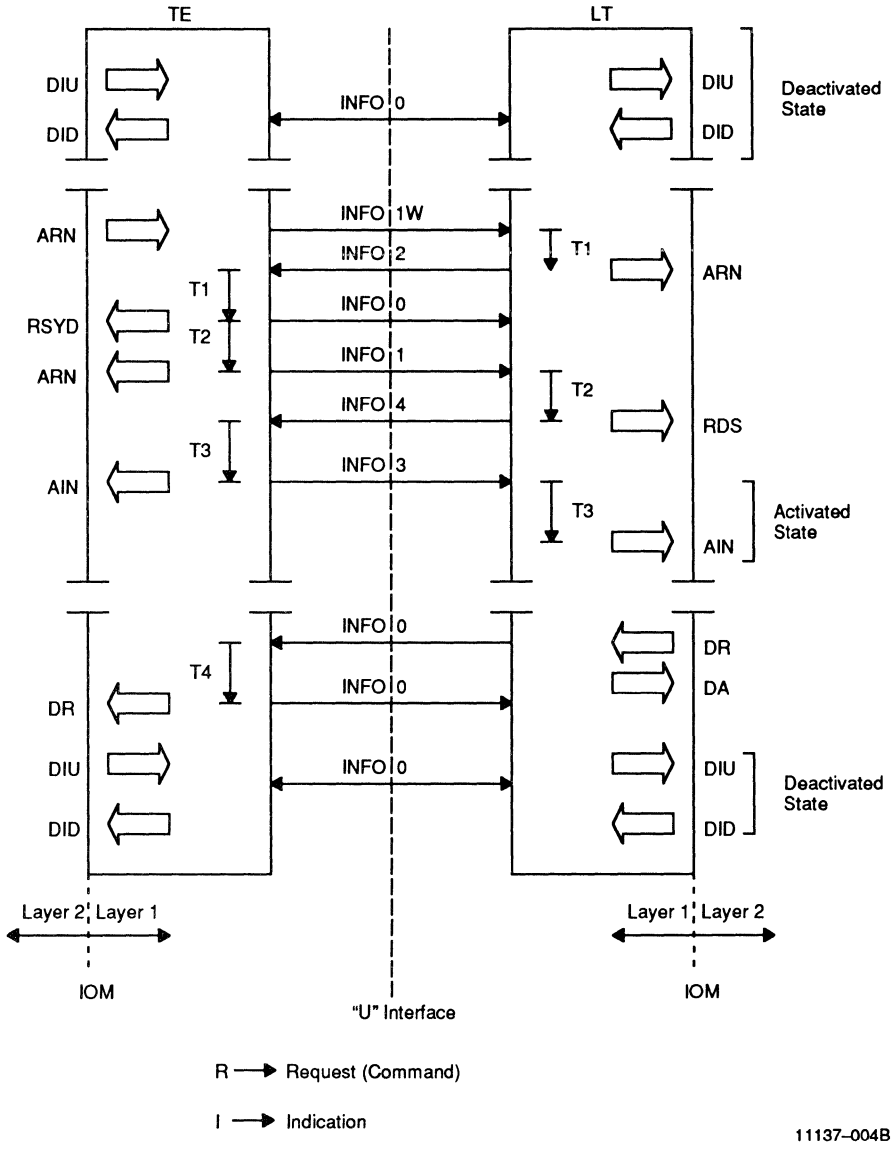


Figure 1. An Example of IBC Activation/Deactivation

Notes for Figure 1:

- T1: 1.5 ms; time for error-free level detect
- T2: < 80 ms; time for synchronization and equalizer adaption
- T3: 0.5 ms; two subsequent bursts with no CV in LF
- T4: 2 ms; time for error-free detection of info 0

FUNCTIONAL DESCRIPTION

System Integration

Figure 2 illustrates the three basic uses of the IBC in IOM architecture. These constitute the three basic operating modes.

- LT: Line Termination in the Local Exchange/PABX
- TE: Terminal Equipment in the Subscriber Terminal
- NT: Network Termination

As can be seen from Figure 2 a second device, the Am2090, ISDN Echo Cancellation Circuit (IEC), may also be used at the U-reference point. The device chosen depends on the application. The IBC proves more cost-effective for shorter range transmission applications (2–3.5 km), such as PABX.

It is available as a 24-pin CMOS device.

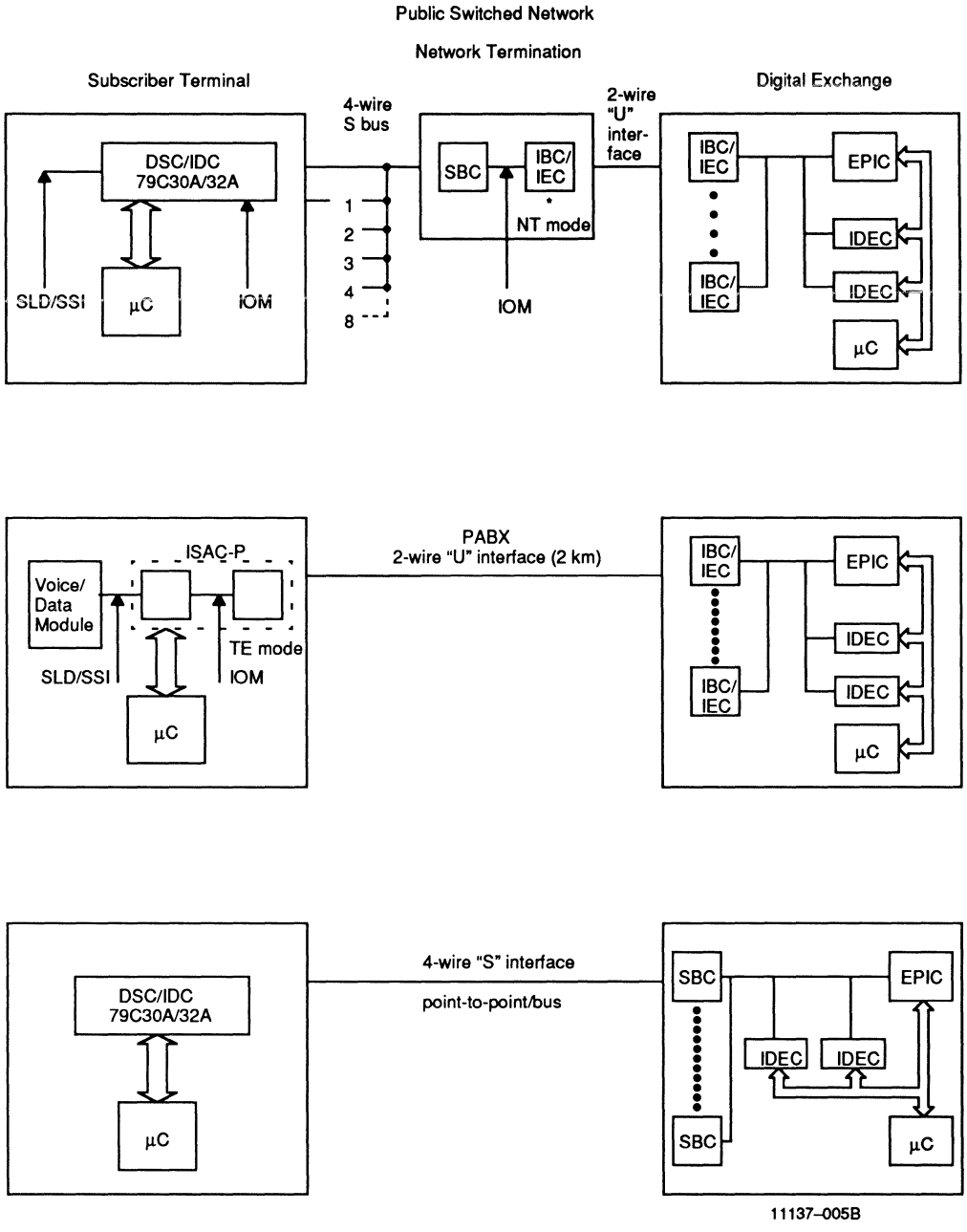
The device operates from a single 5 V power supply.

The maximum power consumption is 80 mW.

Two examples of LT mode are illustrated, one connected directly to the terminal and one connected to a Network Termination. In the latter case, the terminal is connected over the S-interface to the Network Termination. The multiplexing facility on the S-bus allows up to eight terminals to be connected to one Network Termination and one subscriber line. Without a Network Termination, only one terminal per subscriber line is possible. The diagram also indicates that either the IBC or the IEC may be used for 2-wire transmission. The choice depends upon the transmission line characteristics, but in general the IBC is the more cost effective for shorter range transmission applications (especially PABX).

In the LT mode, the IBC manages layer 1 functions and communicates over the IOM interface with the ISDN Digital Exchange Controller (IDEC), which handles layer 2 functions. A microprocessor (handling higher layer functions) controls and communicates with the IDEC. A similar configuration is required in the TE mode employing the same division of tasks.

In the NT mode, however, the configuration is much different. In this case the Network Termination acts as an NT1 (according to CCITT notation). Figure 3 illustrates two possible NT configurations.



* An IBC or IEC pair is chosen depending on application

Am79C30A/32A	Digital Subscriber Controller/ISDN Data Controller	DSC/IDC
Am2080	S Bus Interface Circuit	SBC
Am2090	ISDN Echo Cancellation Circuit	IEC
Am 2095	ISDN Burst Tranceiver Circuit	IBC
Am20950	ISDN Subscriber Access Controller (PABX, "U" Interface)	ISAC-P

Figure 2. ISDN Oriented-Modular (IOM) Architecture

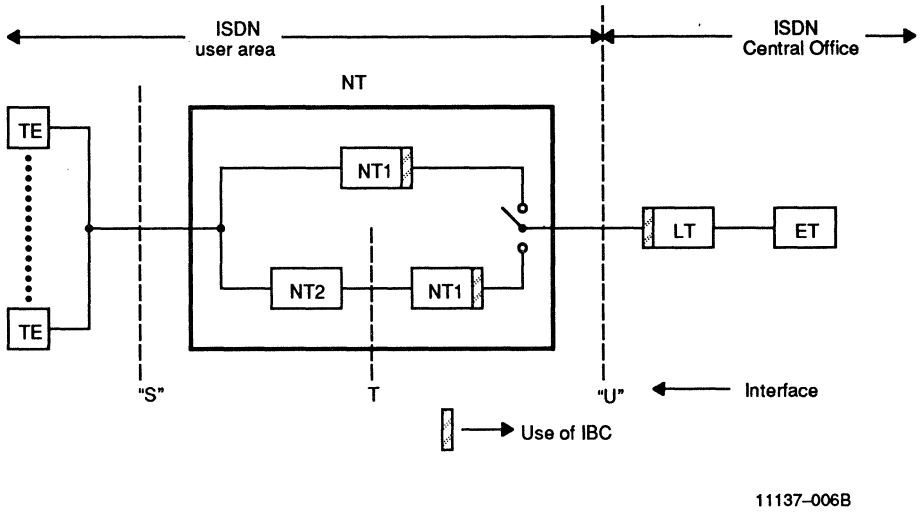


Figure 3. NT Configuration

In both cases NT1 refers to a simple layer 1 translation between the "U" interface and the S/T-interface. This is achieved by the simple pairing of the IBC with an Am2080 S-bus Interface Circuit (SBC) as illustrated in the first example in Figure 2. NT1 is a direct transformation between layer 1 of the "U" interface and layer 1 of the S/T-interface. In this configuration, neither the IDEC or microprocessor is required. The IOM-interface acts as an intermediate interface common to both device.

The NT2 in Figure 3 differs from NT1 in that it includes higher level OSI functions. It could, for example, be a PABX. In this case the PABX would be connected directly over the S-interface (not "U" interface) to the subscriber terminal(s). An example of this case is also illustrated in Figure 2 (bottom diagram).

IBC Selection Criteria

The Am2095, ISDN Burst Transceiver Circuit (IBC), performs the layer 1 functions of the time-division multiplex implementation of the "U" interface. This is a half-duplex technique (ping-pong) involving transmission by only one device at any one time. As mentioned earlier a second device, the Am2090 ISDN Echo Cancellation Circuit (IEC), may be used at the U-reference point instead of the IBC. It uses an echo cancellation technique to implement full-duplex communication along the 2-wire transmission line. Including maintenance and synchronization, the IEC transmits over the line at 160 kbps. To achieve the same user channel data communication rate, the IBC must transmit at 384 kbps (within a burst). Hence, the bandwidth of the line has to be greater for an IBC implementation. Depending on the cable used, the range of IBC transmission is up to 3.5 km while the IEC has a transmission range of up to 8 km. The lower cost of the IBC makes it the logical choice for short transmission line applications.

Note: According to CCITT, the U-reference point is defined as the interface between a local exchange and a Network Termination. The direct connection of terminals to a PABX is not considered by CCITT. For simplicity, the use of the term "U" interface in this document refers both to local exchange-Network Termination connections and PABX-terminal connections. The term will refer only to the 2-wire ping-pong transmission implementation.

IBC Device Architecture and General Functions

Figure 4 illustrates the internal device architecture. The IBC acts not only as a "U" interface transceiver but also links it to the IOM-interface. In effect the IBC implements and links contrasting interfaces. To do this transparently, the IBC must compensate for the following main differences between them:

- The "U" interface is a burst mode interface while the IOM interface is continuous
- The frame structure and data transmission techniques on both interfaces are different
- The B-channels are scrambled on the "U" interface and unscrambled on the IOM-interface
- The clock rates are different and are transmitted in a different manner. In the "U" interface the clock is implicit in the data stream; in the IOM-interface separate clocks, DCL and FSC, must be provided.

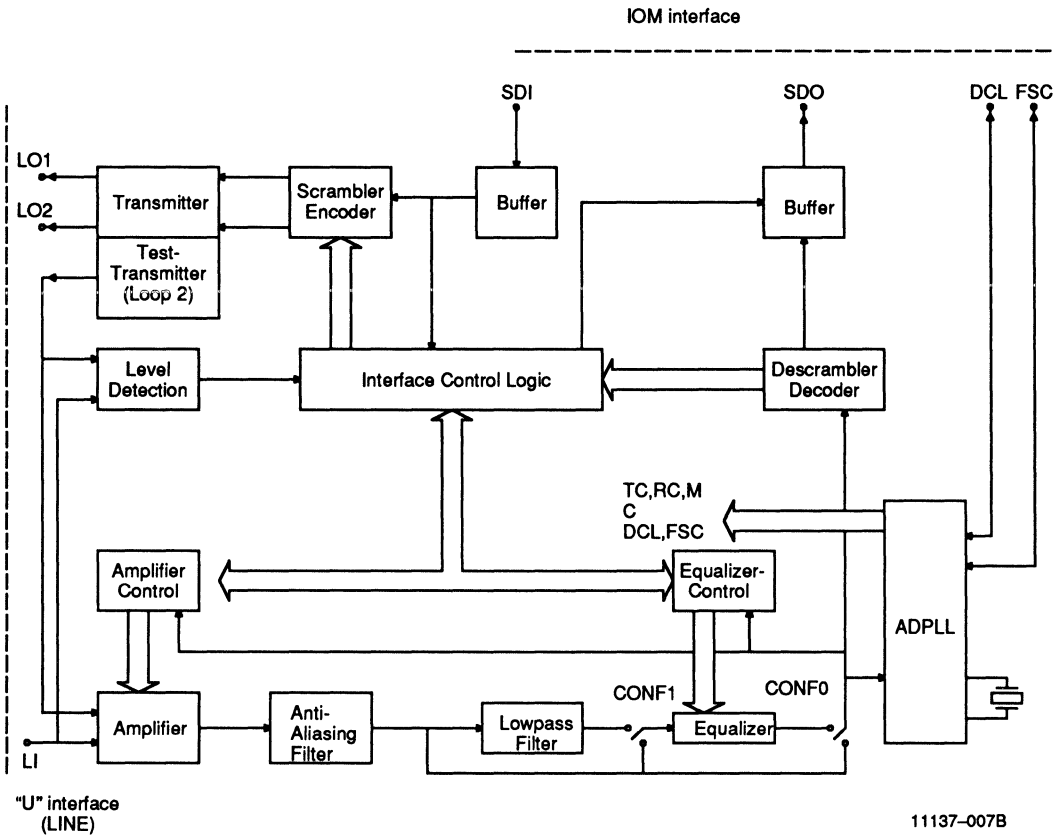
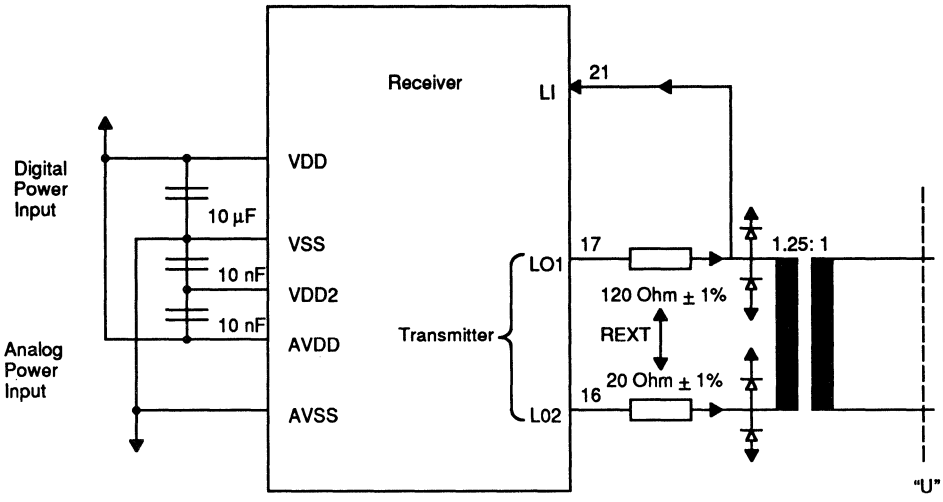


Figure 4. IBC Device Architecture

Besides an interface transformation the IBC must also carry out the following functions:

- Synchronous timing must be maintained on both sides. All internal clocks are synchronized to the upstream data clock (system clock).
- All generated downstream clocks are synchronized, in turn, to these internal clocks.
- Activation/Deactivation procedures. Activation may be initialized by either infos from the line or primitives from the IOM-interface.
- To increase the quality of signal received from the line, the receiver stage contains both an adaptive amplifier and equalizer.
- Testing and Diagnostic functions: Test loops may be closed, test signals may be generated.



Note: VDD2 is a 2.5V reference output.

11137–008B

Figure 5. Analog Connections

Analog Functions

Figure 5 depicts the analog and power connections to the IBC. Both analog and digital power may be connected to a single power source. The reference voltage VDD2 must be linked by two 10 nF capacitors to VSS and AVDD. External to the transmitter and receiver a transformer (ratio 1.25:1) and external resistance (REXT = 140 ohms ± 1) are connected as shown. Voltage overload protection is achieved by splitting REXT into 120 ohms and 20 ohms (for current limitation) and adding clamping diodes.

The transmitter stage is realized as a voltage source with an internal resistance R1 = 15 ohms 40. It delivers a pulse of amplitude 2 V ±10 (0-to-peak). Assuming a transformer winding resistance of the order of 1 ohm, the output resistance seen from the "U" interface will be 100 ohms.

Referring again to Figure 5, the receiver input stages can be seen. They consist of a variable gain amplifier to compensate for signal losses on the line (dynamic range 30 dB). This is followed by an anti-aliasing filter and a switched capacitor low pass filter. Finally a switched capacitor equalizer suppresses the out-of-band noise, which has passed the (anti-aliasing) filter stage, while keeping the pulse distortion low (dynamic range 15.36 dB).

Both the amplifier and the equalizer are adaptive. The amplifier has 128 possible settings and the equalizer 8 (in this sense they are digital). The adaptive logic can be stopped by externally setting the amplifier and equalizer over the IOM interface (Table 3). Once set in this way the settings remain constant. The Monitor Channel can also be used to bypass the equalizer and/or the low pass filter.

The level detection block monitors the receive line and informs the interface logic when an incoming signal is present. It also monitors the test transmitter to perform a similar function during test loop implementation.

Digital Functions

The DPLL circuitry works with an external oscillator or crystal of 15.36 MHz ± 100 ppm. This is used to synchronize all bit and frame clocks with the incoming system clock (that is, from upstream). In the LT mode the system clock is supplied over the IOM-interface. Generation of half-bauded AMI pulses for the line is accomplished by deriving a synchronous transmitter clock using the DPLL. At the NT/TE end of the line the data clock of 384 kHz is implicitly received in the data stream and is extracted by the IBC. From this all synchronous clocks are derived with the aid of the DPLL.

An incorporated finite state machine controls ISDN layer 1 activation/deactivation. This includes wake signal recognition in the "deactivated" state.

Due to the burst nature of "U" interface communication and the continuous nature of communication on the IOM-interface a buffer memory is required to compensate for timing differences.

A scrambler guarantees that the line data contains enough pulses for a reliable clock extraction. The corresponding descrambler on the receiver side regenerates the correct data stream.

The digital control logic also sets the adaptive coefficients on the AGC amplifier and the SC equalization filter.

Interfaces

The IBC operates three interfaces:

- "U" interface
- IOM-interface
- SLD-interface

"U" interface

Figure 6 demonstrates the general principles of the "U" interface burst mode communication technique. A frame transmitted by the exchange (LT) is received by the terminal equipment (TE) after a given propagation delay. The terminal equipment waits a minimum guard time (5.2 µs) while the line clears. It then transmits a frame to the exchange. The exchange will begin a transmission every 250 µs (known as the burst repetition period). However, the time between the reception of a frame from the TE and the beginning of transmission of the next frame by the LT must be greater than the minimum guard time. Communication between an LT and an NT follows the same procedure.

Within a burst, the data rate is 384 kbps and the 38 bit frame structure is as shown in Figure 6. The framing bit (LF) is always logical "1". The frame also contains the user channels (2B+D). Note that the B-channels are scrambled. It can readily be seen that in the 250 µs burst repetition period, 4 D-bits, 16 B1-bits and 16 B2-bits are transferred in each direction. This gives an effective full-duplex data rate of 16 kbps for the D-channel and 64 kbps for each B-channel.

The final bit of the frame is called the M-bit. Four successive M-bits, from four successive U-frames, constitute a superframe (Figure 6). Three signals are carried in this superframe. Every fourth M-bit is a code violation (CV) and is used for superframe synchronization. This can be regarded as the first bit of the superframe. From this reference, bit 3 of the superframe is the service channel bit (S). The S-channel bit is transmitted once in each direction in every fourth burst repetition period. Hence the duplex S-channel has a data rate of 1 kbps. It conveys test loop control information from the LT to the TE/NT and reports of transmission errors from the TE/NT to the LT. Bit 2 and bit 4 of the superframe are T-bits. These constitute the 2 kbps T-channel which extends the T-channel of the IOM frame (Figure 9) onto the "U" interface.

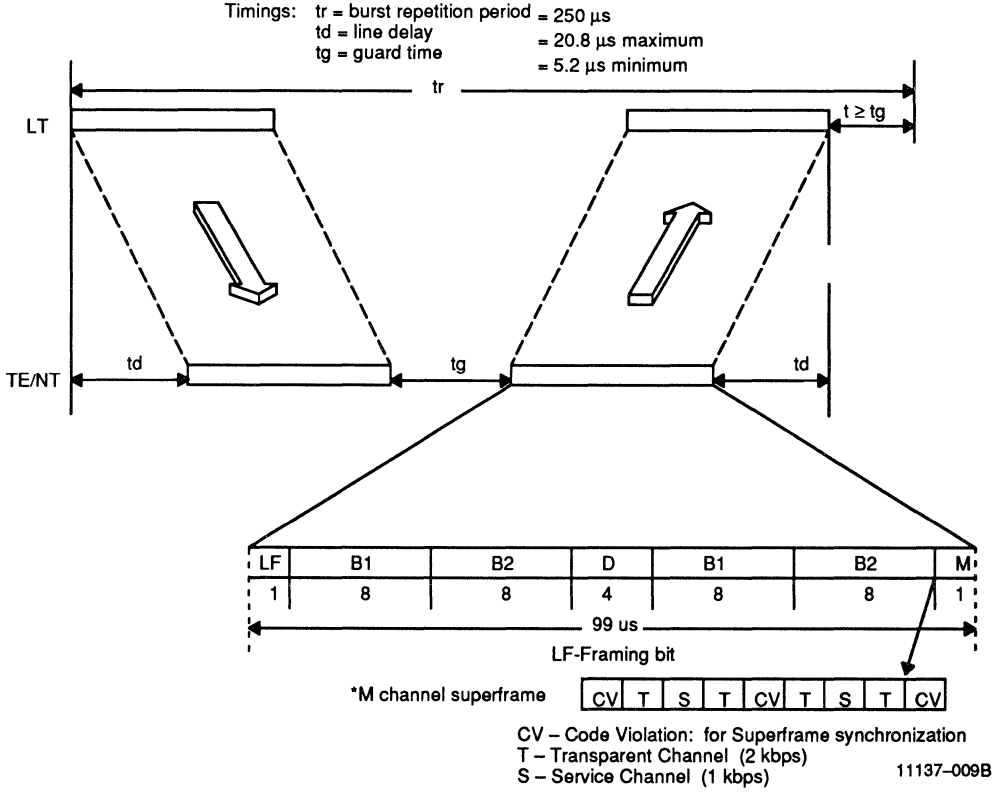


Figure 6. "U" Interface Transmission/Reception

The coding technique used on the "U" interface is half-bauded AMI code (that is, with a 50 pulse width). Figure 7 illustrates the code. As can be seen, a logical "0" corresponds to a neutral level, a logical "1" is coded as alternate positive and negative pulses. The figure also illustrates how a code violation may be achieved (CV); either two successive positive (as shown) or negative pulses.

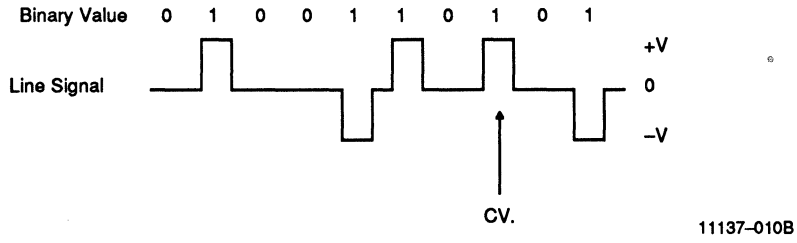


Figure 7. Half-Bauded AMI Code

IOM-Interface

The IOM-interface consists of two clocks: DCL (data clock) and FSC (Frame Synchronization Clock), and two data lines: data out and data in.

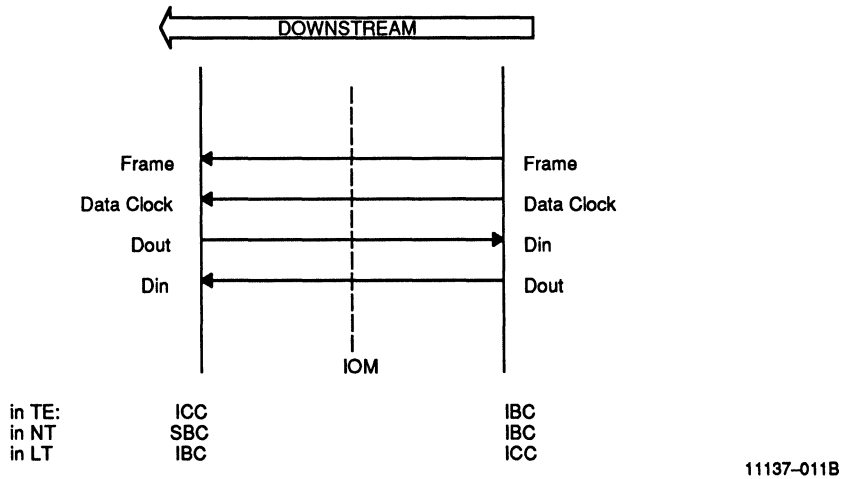
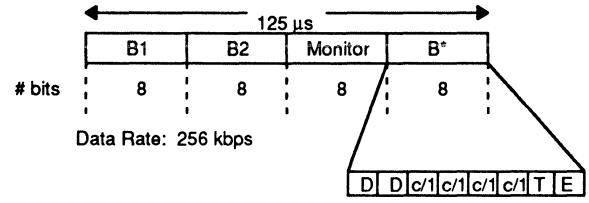


Figure 8. IOM-Interface of IBC

Over the IOM-interface the IBC communicates with layer 2 devices or layer 1 devices (that is, Am2080 SBC) depending on the application (Figure 8).

The IOM frame structure is illustrated below. It contains the 2B+D-channels, transmitted transparently over the interface at 144 kbps. The so-called B-channel is in fact made up of four channels:

- The D-channel (2 bits)
- The Command and Indication Channel (4 bits) used to convey control information across the IOM-interface. This information refers directly to OSI model layer 1 — layer 2 communication. Commands are sent from layer 2 to layer 1 and indications from layer 1 to layer 2.
- The T-channel (1 bit)
- The E-channel (1 bit)



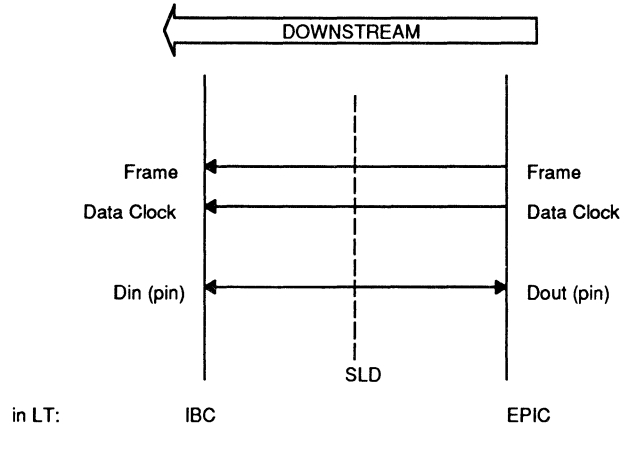
*Timings shown refer to normal mode.

11137-012B

Figure 9. IOM Frame Structure

The 8-bit monitor channel handles information exchange between the layer 2 and layer 1 devices not covered by OSI (that is, AGC coefficient setting). Its use may be restricted by the IDEC MODE register's configuration. For a complete description of available monitor codes and their use, refer to the Operational Description.

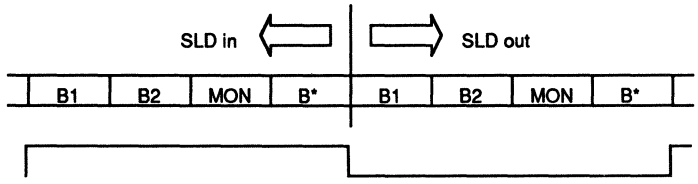
SLD-Interface



11137-013B

Figure 10. SLD-Interface of IBC

The standard SLD-interface with an effective full-duplex rate of 256 kbps, has three lines; a 512 kHz clock (DCL) an 8 kHz frame direction signal (FSC), and a serial data line (SDO). A possible configuration using the SLD-interface is outlined above (Figure 10). Referring to Figure 11 the FSC line indicates the direction of the frame on the data line. Each frame is 32 bits long and within each burst the data rate is twice that in an IOM frame (512 kbps as opposed to 256 kbps). As only one frame can be transmitted in each 125 µs period, the effective transmission rate is the same.



B1 – B channel
B2 – B channel
MON – Monitor
B* – Control and signaling (including D-channel)

*All 4 channels contain 8 bits.

11137–014B

Figure 11. SLD Timing

Operating Modes

The IBC may be configured for three basic operating modes:

- TE mode: ISDN terminals
- NT mode: ISDN Network Termination
- LT mode: ISDN line Termination

Configuration is achieved by pinstrapping (pins M0, MI and sometimes X2). Both the IOM clock signals (DCL, FSC) and the multifunctional pins (X0, X1, X2, X3) have mode dependent functions (refer to Table 4). Besides the three basic operating modes (normal modes), three further modes exist:

- TE: inverted mode (SEL). This differs from TE normal mode only in respect to the clock signals; DCL is inverted, FSC's mark space ratio is altered.
- LT: SLD mode. This uses the SLD interface (that is, SDO only, half duplex at 512 kbps) rather than the IOM-interface (that is, SDO and SDI at 256 kbps each).
- LT: MUX mode. Figure 12 illustrates the difference between LT normal and multiplex (MUX) modes. The FSC period is the same (the mark space ratio is different). The DCL clock runs at 4096 kHz, allowing a data rate of 2048 kbps. Eight 32 bit IOM frames can be transferred in one FSC period, each destined for a separate IBC. Each IBC (all in LT MUX mode) reads only one of the eight IOM frames (that is, from time slot determined by pin-strapping X0, X1 and X2).

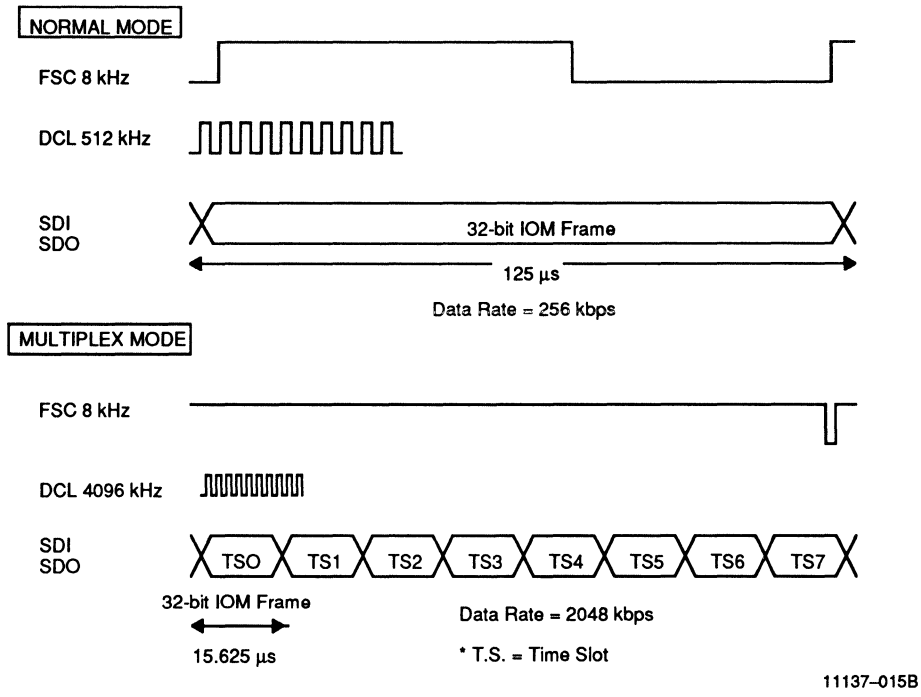


Figure 12. LT Normal and Multiplex Mode Timing

Figure 12 illustrates the different IBC modes in relation to timing recovery.

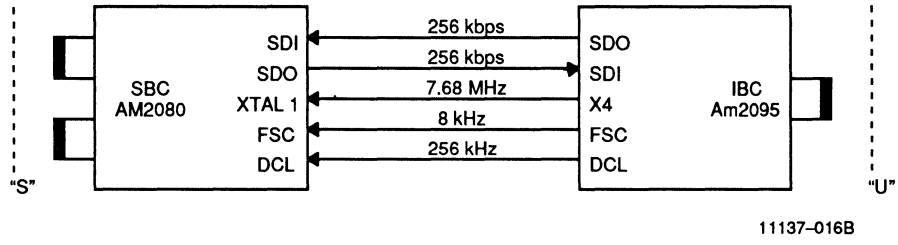


Figure 13. IBC Application

Individual Functions

As mentioned previously, pins X0 to X4 have mode-specific functions. These are detailed in Table 5. Three types of functions exist:

- Auxiliary mode selection (X2 only).
- Special input pins (that is, PFOFF in LT:, time slot selection in LT:MUX).
- Clock outputs (both synchronized and unsynchronized to the line).

The clock outputs act as external oscillators for other devices. For example in Figure 13, NT Mode, the SBC uses the 7.68 MHz synchronized output from the IBC as an external oscillator. Further possible applications include:

- 15.36 MHz (LT) — external oscillator for other IBCs
- 1.536 MHz (TE) — CODEC clock supply

Test Functions

Three types of test loops may be closed in the IBC, depending on the operating mode. In all test loops, all three channels, B1, B2, and D, are looped back. In a transparent loop the data is sent forward unmodified as well as being looped back. In a non-transparent loop the forward data path is blocked.

Figure 14 illustrates these test loop configurations.

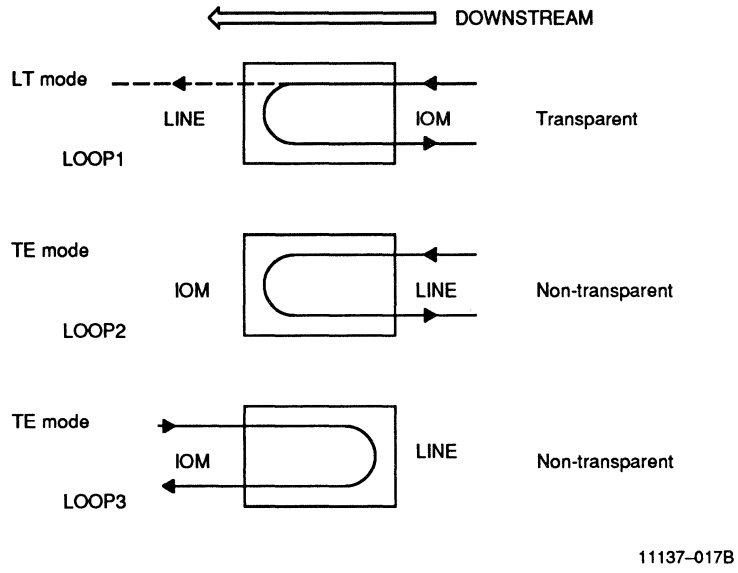


Figure 14. IBC Test Loops

No test loops are implemented in the IBC in NT mode. However a test loop is possible in the SBC. The test loop signal is passed onto the SBC, by the IBC, by means of an IOM Command.

Two test signals can be sent by the IBC: Test Mode 1, sending continuous pulses and Test Mode 2, sending single pulses. In both cases, half-bauded AMI coding is used, but only one of the IBCs at either end of the "U" interface line transmits. Hence, burst mode transmission is abandoned. In Test Mode 1, a one is transmitted at 384 kbps. Because of the alternate positive-negative nature of the coding, the signal on the line has a frequency of 192 kHz. In Test Mode 2, a one is transmitted at 4 kbps yielding a signal of 2 kHz on the line.

CONTROL CODES

Control/Indication Commands

As mentioned in the IOM-Interface section, the C/I codes transfer information between layer 1 (IBC) and layer 2 (IDEC) directly related to the OSI reference model (that is, activation/deactivation, test loop initialization, reset, and so on). The exchange of C/I codes is governed by the state machine. Commands are sent from layer 2 to layer 1, indications from layer 1 to layer 2. A complete list of all codes in each mode is given in Tables 1 and 2.

C/I codes are transmitted in the channel of the IOM frame and must follow the following procedure: When a new command or indication is to be sent it must be repeated in two successive IOM frames before it is recognized at the receiver.

Table 1. Commands and Indications in the LT Mode

Command (downstream)	Abbr.	Code	Remark
Deactivate Request	DR	0000	(*x*)
Activate Request No loop	ARN	1000	
Activate Request Local loop	ARL	1001	Loop 1 activation requested
Activate Request Loop 2	AR2	1010	
Deactivate Indication	DID	1111	
Reset	RES	1101	Software Reset (*x*)
Disable Supply	DIS	0011	Used to control the drain on local power supply (*x*)
Send Single Pulses	SSP	0101	Ones (AMI pulses) transmitted at 4kHz (TEST MODE 1) (*x*)
Send Continuous Pulses	SCP	0111	Ones (AMI pulses) transmitted continuously (TEST MODE 2) (*x*)

Indication (upstream)	Abbr.	Code	Remark
Deactivate Indication	DIU	1111	
Deactivate Acknowledge	DA	0001	
RDS Indication	RDS	0111	Running Digital Sum. Code violation register enabled. Receiver synchronized
Activate Request	ARU	1000	
Activate Indication	AIU	1100	
Resynchronization	RSYU	0100	Lost Framing: receiver not synchronized to the input signal
High Impedance	HI	0011	After PFOFF the phantom power supply becomes high impedance

(*x*) unconditional commands

Table 2. Commands and Indications in the NT/TE Mode

Command (upstream)	Abbr.	Code	Remark
Timing	TIM	0000	Layer 2 device requires clocks to be activated
Activate Request No loop	ARN	1000	Activate Request Local
Activate Request Local Loop	ARL	1001	*Test loop 3 activation request
Activate Indication	AIN	1100	**
Deactivate Indication	DIU	1111	
Reset	RES	1101	Software Reset (*x*)
Send Single Pulses	SSP	0101	Ones (AMI pulses) transmitted at 4 kHz (TEST MODE 1) (*x*)
Send Continuous Pulses	SCP	0111	Ones (AMI pulses) transmitted continuously (TEST MODE 2) (*x*)

Indication (downstream)	Abbr.	Code	Remark
Deactivate Request	DR	0000	
Power Up	PU	0111	
Resynchronization	RSYU	0100	Lost Framing: receiver not synchronized with input
Activate Request No loop	ARN	1000	
Activate Request Local loop	ARL	1001	*
Activate Request Loop 2	AR2	1010	
Activate Indication No loop	AIN	1100	
Activate Indication Local loop	AIL	1101	*Test Loop 3 activated
Activate Indication Loop 2	AI2	1110	
Testmode Acknowledge	TMA	0101	*
Deactivate Indication	DID	1111	

(*x*) unconditional commands
* only in TE
** only in NT

Monitor Commands

Besides the C/I section of the B-channel, inter-device communication on the IOM-interface is also possible via the monitor channel. However, the monitor channel protocol is more straightforward; the layer 2 device commands, the layer 1 device responds. The monitor codes deal mainly with auxiliary tasks not directly related to the maintenance of the link. They are totally under the control of the user and contain 17 NOP codes (00_H and FX_H). Table 3 lists all IBC monitor codes.

The monitor channel is accessed by the IDEC by writing to the MONR register. Upon receipt of a message the IBC will send a response in the monitor channel and will indicate this by setting the E-bit of the IOM frame to zero. This will be input into the MONR register of the IDEC. However if bit MODE: HMDI is zero (IDEC), the response will not be input into the MONR register.

The monitor channel is used to read the IBC internal registers I(7:0) and R(7:0)) and to set or simply read the adaptive amplifier and equalizer settings. Note, however, that once both amplifier and equalizer coefficients have been set, the adaptive logic is turned off and the given settings remain constant regardless of any changes to the input signal.

However, the coefficients are zeroed and must be reprogrammed after either:

- The device has been through the deactivated (Power-Down) state
- A local loop has been implemented (that is, loop 1 in LT mode and loop 3 in TE mode)

The adaptive logic is turned on again only after a hardware or software reset (coefficients again zeroed).

Merely reading the values however does not affect the adaptive nature of the settings. The monitor codes can also be used to bypass certain stages of the IBC internal architecture. The response $3F_H$ to the message 80_H (identification) identifies the responding device as an IBC rather than another IOM compatible device. Note that each device has a separate set of codes.

Table 3. IBC Monitor Commands

Message	Response	Exploration
00000000	----	NOP
11111111	----	NOP
10000000	00111111	Identification
11101110	I(7:0)	Supply current equivalent
11101111	R(7:0)	Code violation register
00000010	1XX,A(6:2)	Gain factor (part 1)
00000011	1XX,C(2:0),A(1:0)	Equalizer coefficient and gain factor (part II)
001,A(6,2)	101,A(6:2)	External programming of gain factor (part I)
010,C(2:0),A(1:0)	110,C(2:0),A(1:0)	Programming of gain factor (part II) and equalizer coefficient
011,LPF,CONF(3:0)	111,LPF,CONF(3:0)	Programming of LPF and CONF(3:0)

To bypass any of the following integrated functions set the corresponding bit to 0 (Reset Values all HIGH).

LPF : DPLL loop filter
CONF3 : Always set high
CONF2 : Always set high
CONF1 : SC-low pass filter
CONF0 : Equalization filter

Note: 1. Messages are sent by the layer 2 device to the IBC.

2. Responses are sent by the IBC to the layer 2 device.

IBC Registers

The IBC contains two registers which are user-accessible over the monitor channel.

Supply Current Equivalent

This register is useful only in the LT mode when the power supply control functions are active. In this mode X3 functions as the MPF pin (Main Power Feed). Through this pin the 8-bit supply current equivalent from the local power supply can be read into the I(7:0) register. This read is synchronous to the BI channel in the IOM frame (in time slot 0 if the LT is in MUX mode). The IDEC uses the information to control the power supply. If the current value is too high the software issues a DIS command. Pin DISS on the IBC goes HIGH and, if properly connected, this pin can be used to disable the power supply. After reset, I(7:0) = 00H.

R(7:0) Code Violation Register

This register counts the number of code violations in the data channels encountered by the IBC, and this information may be sued by the software to estimate the error rate.

2

ABSOLUTE MAXIMUM RATINGS

Ambient Temperature under bias ..... 0° to +70°C
Storage Temperature ..... –65 to +125°C
Voltage on any pin with respect to ground ..... –0.3 to V_{DD} +0.3

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Ambient Temperature (T_A) ..... 0° to +70°C
Operating V_{DD} Range ..... 4.75 V to 5.25 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS

T_A = 0 to +70°C; V_{DD} = +5 V ±5%, C_{AA} = 0 V, AV_{SS} = 0 V.

	Symbol	Parameter	Limit Values Min	Max	Unit	Test Condition
	V_{IL}	Input low voltage	V_{SS}–.4	0.8	V	
	V_{IH}	Input high voltage	2.0	V_{DD}+ 0.4	V	
All	V_{OL1}	Output low voltage[1]		0.45	V	IOL = 2 mA
pins	V_{OL2}	Output low voltage[2]		0.45	V	IOL = 7 mA
except	V_{OH}	Output high voltage	2.4		V	IOL = –400 µA
LO1,2	V_{OH}	Output high voltage	V_{DD}–.5		V	IOL = –200 µA
LI						
XTAL1	I_{CC}	Power Supply Current:				V_{DD} = 5 V, inputs at 0 V or V_{DD},
XTAL2		Operational		13	mA	output loads.
		Power Down		1.3	mA	
	I_{LI}	Input leakage current				0 V < V_{IN} < V_{DD} to 0 V
	VX	Absolute value of[3]	4.75	5.25	V	
		output pulse amplitude[4]	–5.25	–4.75	V	
		(VLO1–VLO2)[5]	0	0	V	
LO1,2	PW	Pulse width	1.22	1.38	µs	
	RX	Transmitter output impedance	9	21	ohm	
XTAL1	V_{IL}	Input low voltage		0.5	V	
	V_{IH}	Input high voltage	V_{DD}–.5		V	
XTAL2	V_{OL}	Output low voltage		0.5	V	Io ≤ 100 µA
	V_{OH}	Output high voltage	V_{DD}–.5		V	C_L ≤ 100 pF

Notes:
1. All outputs except SDO
2. Output SDO only
3. Positive pulse
4. Negative pulse
5. No pulse

Capacitances

T_A = 0 to +70°C; V_{DD} = 5 V ±5%, V_{SS} = 0 V, AV_{SS} = 0 V.

	Symbol	Parameter	Limit Values Min	Max	Unit	Test Condition
	CIN	Input capacitance		7	pF	
	CIO	Output capacitance		7	pF	
		Output capacitance		7	pF	
LO1,2	COUT	against AV_{SS}		10	pF	
LI	CIN	Input capacitance		7	pF	
XTAL1,2	CLD	Load capacitance		50	pF	

Recommended Oscillator Circuits

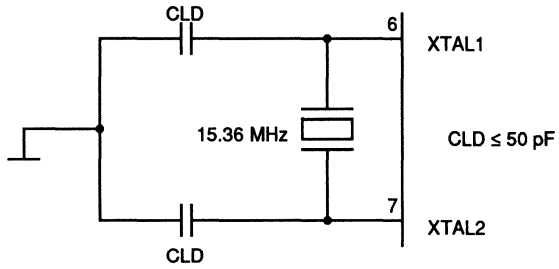


Figure 15. Crystal Oscillator Mode

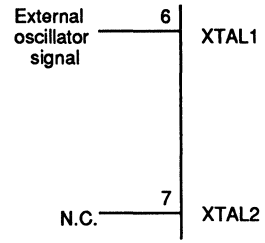


Figure 16. Driving From External Source

SWITCHING CHARACTERISTICS

T_A = 0 to +70°C; V_{DD} = 5 V ± 5%

Inputs are driven at 2.4 V for a logic "1" and at 0.4 V for a logic "0". Timing measurements are made at 2.0 V for a logic "1" and at 0.8 V for a logic "0".

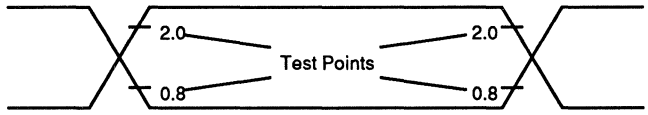


Figure 17. AC Testing Input/Output Waveforms

Clock Timing

Table 4 summarizes the clocks produced in the different operating modes and their respective duty cycles. The table also indicates which clocks are derived directly from the crystal and which are synchronized to the line using the on-board DPLL circuitry.

Table 4. IBC Operating Modes and Mode Specific Pin Configurations

	Application					
	TE	TE	NT	LT	LT	LT
Operation of IOM Interface	**Inverted MODE (SEL)**	**Normal Mode**	**Normal Mode**	**Normal Mode**	**SLD Mode**	**MUX Mode Inverted**
M1	0	0	0	1	1	1
M0	0	0	1	0	0	1
DCL	O:512 kHz* 1:1	O:512 kHz* 1:1	O:512 kHz* 1:1	O:512 kHz	O:512 kHz	O:512 kHz
FSC	O:8 kHz* 63:1	O:8 kHz* 1:1	O:8 kHz* 1:1	I:8 kHz	I:8 kHz	I:8 kHz
X4	O: 2.56 MHz 1:2	O: 2.56 MHz 1:2	O: 7.68 MHz* 1:1	I:PFOFF	I:PFOFF	I:PFOFF
X3	I:ENCK	I:ENCK	I:ENCK	I:MPF	I:MPF	I:MPF
X2	I:1	I:0	I:SCP	I:0	I:1	I:TS2
X1	O: 1.536 MHz* 1:1	O: 1.536 MHz* 1:1	I:$\overline{SSP}$	O: 1.536 MHz 1:1	O: 1.536 MHz 1:1	I:TS1
X0	O:3.84 MHz	O:3.84 MHz	I:LPF	———	———	I:TS0

*: synchronized to S I: Input O: Output

Note:
1. SDI - (IOM Interface)
   - the pin is connected to an internal pull-up resistor in TE normal and LT normal modes.
2. SDO - (IOM Interface)
   - open drain output in LT MUX mode inverted.
   - open drain output with internal pull-up resistor in NT-mode.
   - push-pull otherwise.
3. The following clock outputs are derived from the 15.36 MHz crystal/external oscillator: (that is, unsynchronized)
   - 15.36 MHz
   - 3.84 MHz
   - 2.56 MHz
4. The following clock outputs are synchronized to the line
   - 7.68 MHz
   - 1.536 MHz

Table 5. Mode Specific Pin Functions

Mode	Name	Description	Pin	I/O	Function
LT	PFOFF	Power feed off	X4	I	Puts the IBC into a powerfeed off state. This state is indicated by C/I code HI.
LT	MPF	Main Power Feed	X3	I	The 8-bit supply current equivalent is read serially through this pin into I(7:0) from the local power supply. The read is synchronous to the B1 channel in the IOM frame (time slot 0 in LT:MUX mode). Used for power supply control by the layer 2 device. Tie LOW when not in use.
TE/NT	ENCK	Enable clocks	X3	I	Enables clocks in "deactivated" state. Also during RST = 0, outputs are low impedance when ENCK = 0 and high impedance otherwise.
NT	SSP	Send Single Pulses	X1	I	Test Mode 1
NT	SCP	Send Contin. Pulses	X2	I	Test Mode 2
NT	LPF	Loop Filter	X0	I	The PLL loop filter can be switched on (1) or off (0). Also available in Monitor Channel.
LT	TSO-2	Time Slot 0–2	X0, 1, 2	I	In MUX mode, one of eight possible time slots is selected to be read by the device (TSO = LSB).

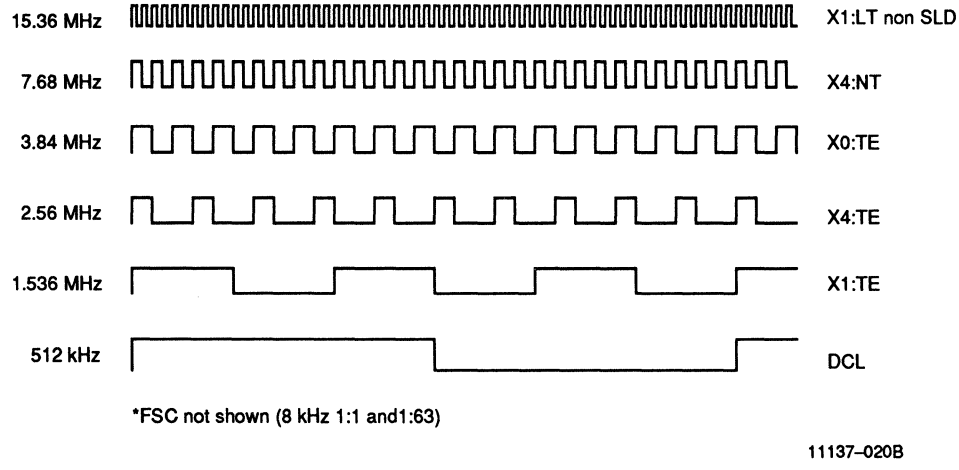


Figure 18. Output Clock Relationships

Figure 18 shows the relationship between the various clock outputs from the IBC. The crystal frequency is 15.36 MHz. All clock outputs have a duty cycle of 1:1 except 2.56 MHz (1:2). Note that the following are derived directly from the crystal oscillator; 15.36 MHz and 2.56 MHz. They are not synchronized to the line. Their accuracy will be, to a first order, governed by the crystal accuracy (+ppm maximum).

The following clocks are derived both from the crystal and, with the help of the DPLL, from the line; 7.68 MHz, 1.536 MHz, DCL and FSC. Synchronization may be regarded as a two-stage process. First, a synchronous 7.68 MHz signal is derived using the DPLL. Second, all other synchronous clocks are derived, by simple division, from 7.68 MHz synchronous. Because of the internal method of synchronization employed, the 7.68 MHz signal may "step forward or back" by 1 crystal period (see Figure 18). Therefore, the period of 7.68 MHz, and all synchronous clocks derived from it, may vary by one crystal period (+ 65 ns). This, to a first order, gives the accuracy of the various synchronous clocks. Tables 6 and 7 detail the accuracy of the clock outputs with respect to the symbols defined in Figure 20.

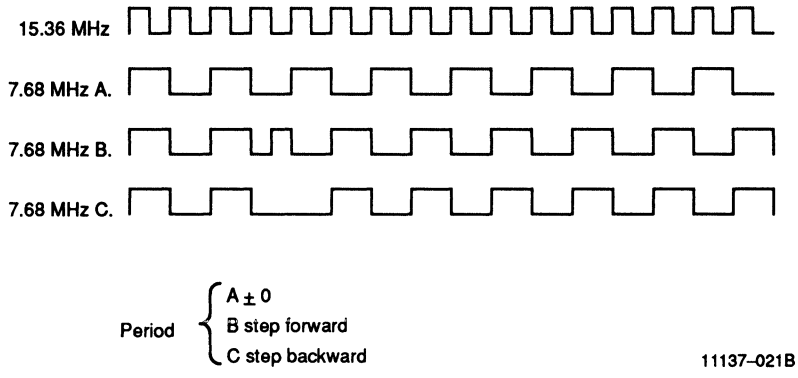


Figure 19. Possible 7.68 MHz Clocks

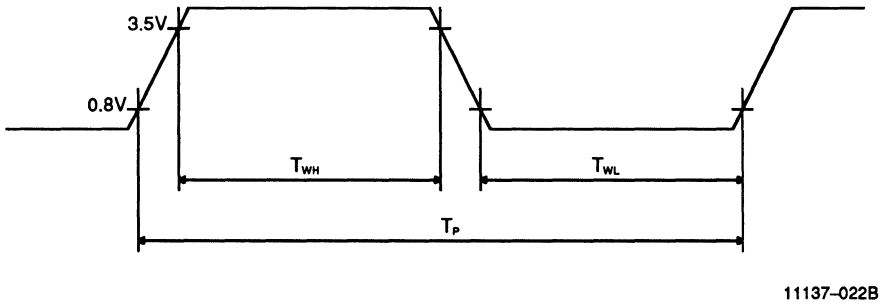


Figure 20. Clock Timing Symbols

Table 6. DCL Timing

	Symbol	Description	Min	Typ	Max	Unit
Output	T_P	TE/NT 512 kHz	1988	1953	2019	ns
	T_{WH}	TE/NT 512 kHz	944	977	1009	ns
	T_{WL}	TE/NT 512 kHz	944	977	1009	ns
Input	T_{WH}	LT mode	90			ns
	T_{WL}	LT mode	90			ns

Table 7. FSC Timing

	Symbol	Description	Min	Typ	Max	Unit
Output	T_P	TE/NT 8 kHz 1:1	124.93	125	125.07	ns
	T_{WH}	TE/NT 8 kHz 1:1	62.46	62.5	62.54	ns
	T_{WL}	TE/NT 8 kHz 1:1	62.46	62.5	62.54	ns
Input	T_{WH}	TE(SEL) 8 kHz 63:1	124.93	125	125.07	ns
	T_{WL}	TE(SEL) 8 kHz 63:1	122.08	123.05	124.02	ns
	T_{WL}	TE(SEL) 8 kHz 63:1	1888	1953	2019	ns

Table 8. X4 Clock Timing

Symbol	Description	Min	Typ	Max	Unit	Conditions
T_P	TE 2.56 MHz 1:2	−100 ppm	781	+100 ppm	ns	osc+100 ppm
T_{WH}	TE 2.56 MHz 1:2	−100 ppm	260	+100 ppm	ns	osc+100 ppm
T_{WL}	TE 2.56 MHz 1:2	−100 ppm	520	+100 ppm	ns	osc+100 ppm
T_P	NT 7.68 MHz 1:1	65	130	196	ns	
T_{WH}	NT 7.68 MHz 1:1	65	65	131	ns	
T_{WL}	NT 7.68 MHz 1:1	65	65	131	ns	

Table 9. X1 Clock Timing

Symbol	Description	Min	Typ	Max	Unit	Conditions
T_P	TE 1.536 MHz	585	651	717	ns	
T_{WH}	TE 1.536 MHz	260	326	391	ns	
T_{WL}	TE 1.536 MHz	260	326	391	ns	
T_P	LT* 15.36 MHz	−100 ppm	65.1	+100 ppm	ns	osc+100 ppm
T_{WH}	NT*15.36 MHz	−100 ppm	65.1	+100 ppm	ns	osc+100 ppm
T_{WL}	NT*15.36 MHz	−100 ppm	65.1	+100 ppm	ns	osc+100 ppm

* in normal and SLD modes only

Table 10. X0 Clock Timing

Symbol	Description	Min	Typ	Max	Unit	Conditions
T_P	TE 3.84 MHz	−100 ppm	260	+100 ppm	ns	osc+100 ppm
T_{WH}	TE 3.84 MHz	−100 ppm	130	+100 ppm	ns	osc+100 ppm
T_{WL}	TE 3.84 MHz	−100 ppm	130	+100 ppm	ns	osc+100 ppm

Table 11 defines the rise and fall times of DCL and FSC clocks in the various modes.

Table 11. DCL/FSC Rise and Fall Timing

Description	Mode	Min	Typ	Max	Unit	Condition
TRD; DCL rise time	NT/TE			50	ns	
	LT normal			60	ns	
	LT MUX			25	ns	
TFD; DCL fall time	NT/TE			50	ns	
	LT normal			60	ns	
	LT MUX			25	ns	
TFR; FSC rise time	NT/TE			50	ns	
	LT normal			60	ns	
	LT MUX			50	ns	
TFF; FSC fall time	NT/TE			50	ns	
	LT normal			60	ns	
	LT MUX			50	ns	

IOM-Interface

Given the clock accuracies defined in the previous section the following paragraphs define the timing relationships between the data and the DCL and FSC clocks.

Normal Mode

Master mode (TE/NT mode)

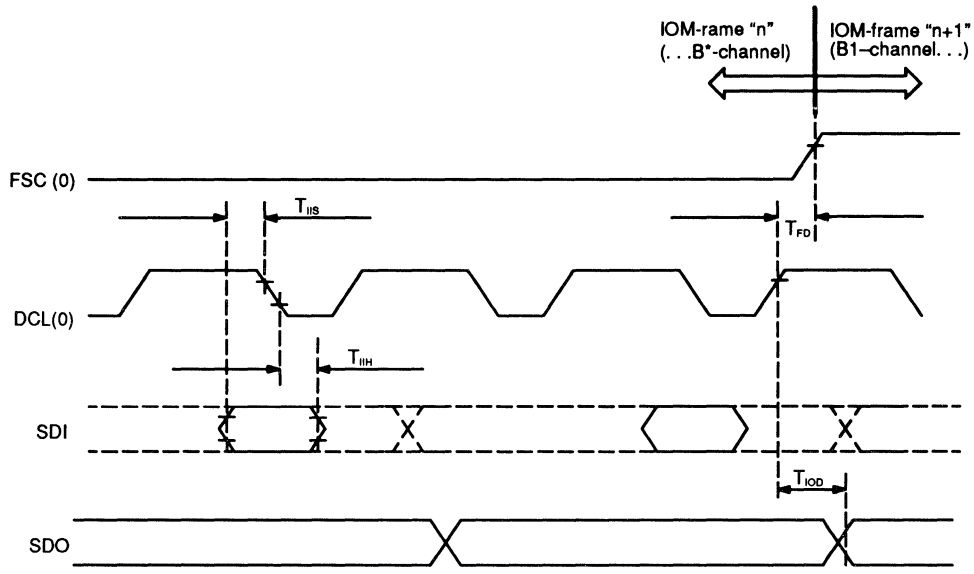


Figure 21. Normal TE/NT Mode Timing Diagram

Table 12. Normal TE/NT Mode Timing

Symbol	Parameter	Min	Max	Unit	Conditions
T_{FD}	Frame sync. delay	–20	20	ns	CL=100 pf
T_{IOD}	IOM output data delay		200	ns	CL = 100 pf
T_{IIS}	IOM input data setup	20		ns	
T_{IIH}	IOM input data hold	50		ns	

Slave Mode (LT)

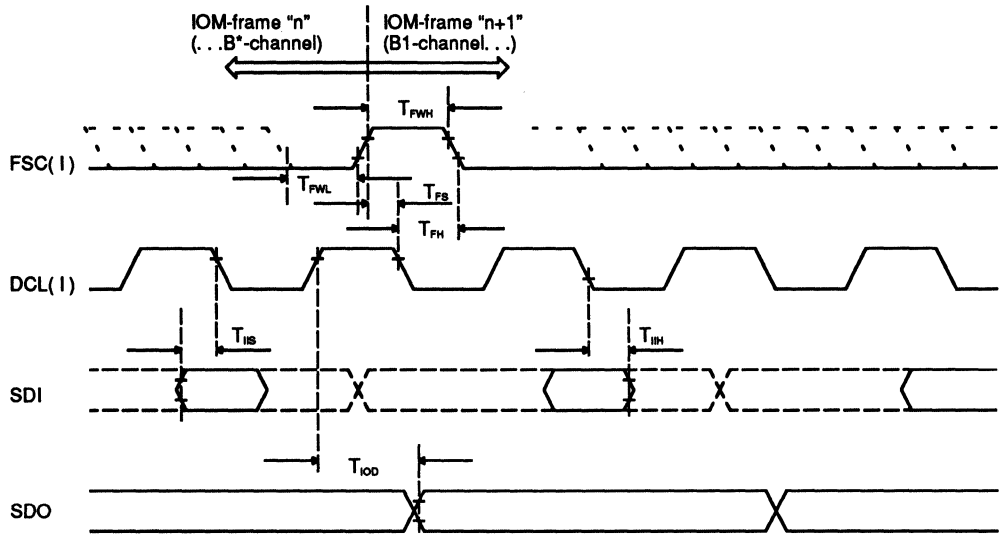


Figure 22. Normal LT Mode Timing Diagram

Table 13. Normal LT Mode Timing

Symbol	Parameter	Min	Max	Unit	Conditions
T_{FH}	Frame sync. hold	50		ns	
T_{FS}	Frame sync. setup	30		ns	
T_{FWH}	Frame sync. high	80		ns	
T_{FWL}	Frame sync. low	2150		ns	
T_{IOD}	IOM output data delay		200	ns	
T_{IIS}	IOM input data setup	20		ns	
T_{IIH}	IOM input data hold	50		ns	

TE Inverted Mode

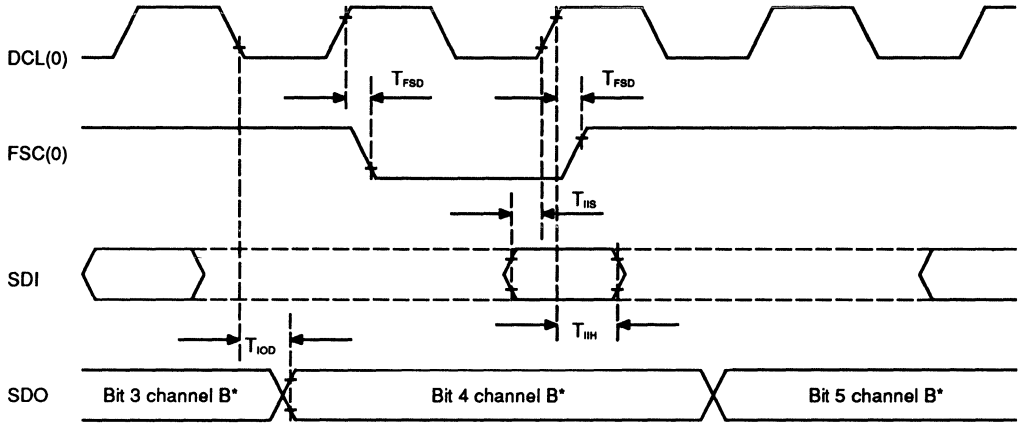


Figure 23. Inverted TE Mode Timing Diagram

Table 14. Inverted TE Mode Timing

Symbol	Parameter	Min	Max	Unit	Conditions
T_{FSD}	Frame sync. delay	-20	20	ns	C_L = 100 pF
T_{IOD}	IOM output data delay		200	ns	C_L = 100 pF
T_{IIS}	IOM input data setup	20		ns	
T_{IIH}	IOM input data hold	50		ns	

LT: MUX Mode

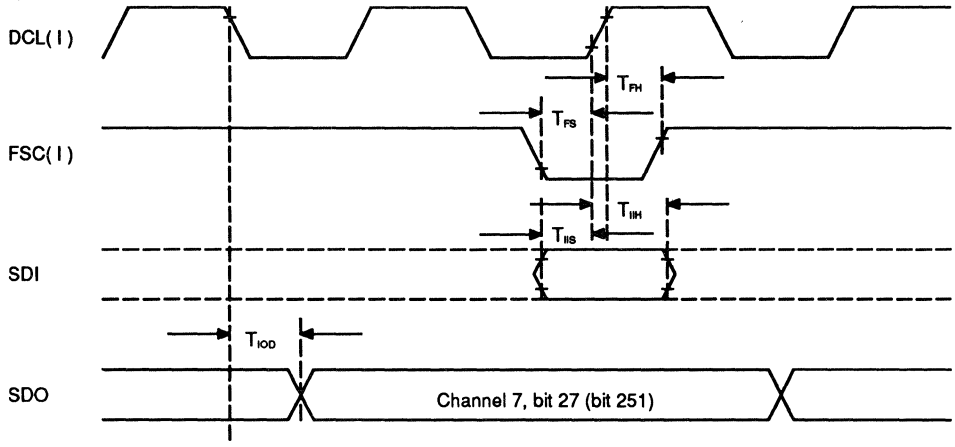


Figure 24. Inverted LT MUX Mode Timing Diagram

Table 15. Inverted LT MUX Mode Timing

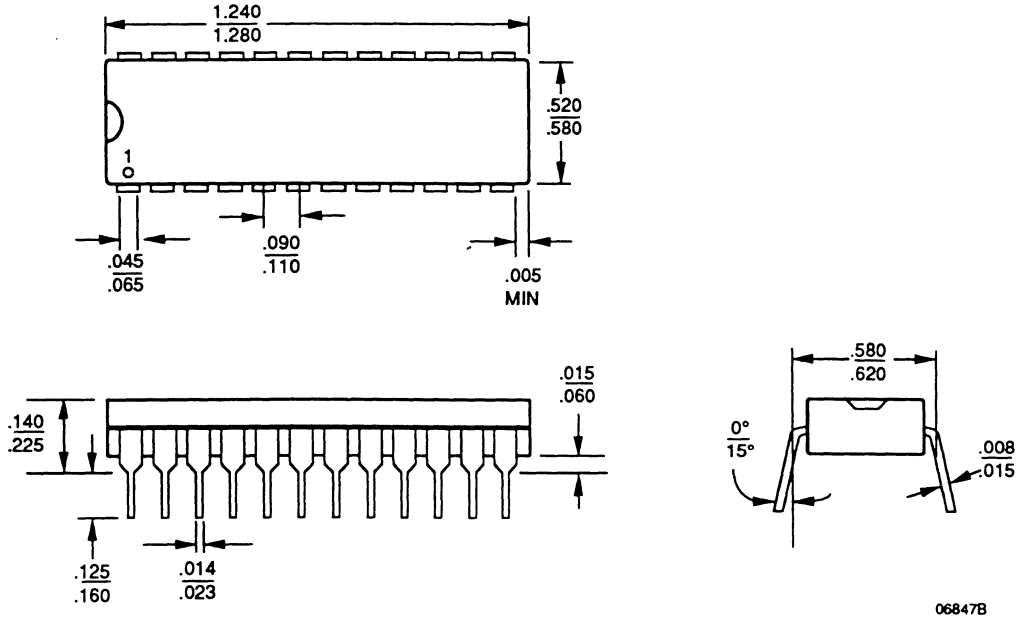
Symbol	Parameter	Min	Max	Unit	Conditions
T_{FSD}	Frame sync. hold	50		ns	
T_{FS}	Frame sync. setup	20		ns	
T_{FH}	Frame sync. high	124.8		µs	
T_{FL}	Frame sync. low	70	200	ns	
T_{IOD}	IOM output data delay		200	ns	
T_{IIS}	IOM input data setup	20		ns	
T_{IIH}	IOM output data hold	50		ns	

Receiver Stage Properties

Table 16. Receiver Stage Properties

Input stage/measured property			dB
Line Amplifier	-	Dynamic Range	0–30
	-	Resolution (128 setting)	0.236
Anti-Aliasing Filter and Low Pass Filters			
> 1.1 MHz	-	Minimum attenuation	30
> 1.1 MHz	-	Typical attenuation	35
Equalizer	-	Dynamic Range	0–15.36 dB
	-	Resolution (8 settings)	2.194

PD 024


1.240
1.280
.520
.580
1
.045
.065
.090
.110
.005
MIN
.015
.060
.140
.225
.125
.160
.014
.023
.580
.620
0°
15°
.008
.015
06847B

Am20901/Am20902

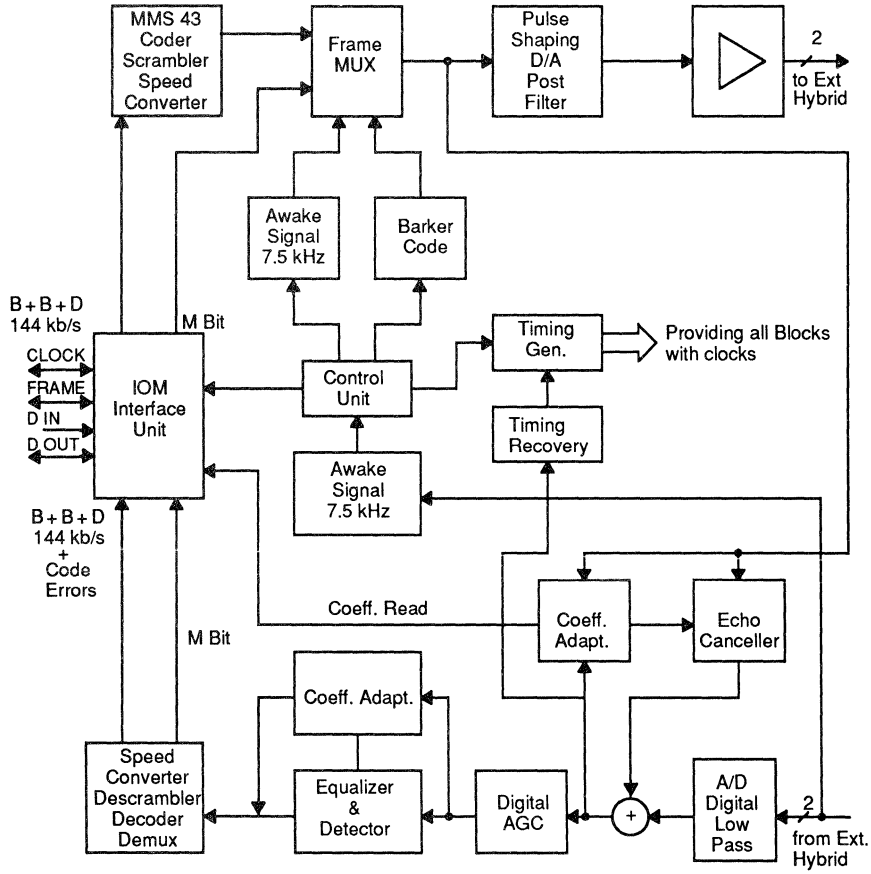
ISDN Echo Cancellation Circuit (IEC) Two-Chip Set

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Full duplex transmission and reception of the "U" (4B3T) interface signals according to the FTZ Guideline 1R 220 of the Deutsche Bundespost (DBP)
  - –144 kbps user bit rate over standard local telephone loops
  - –1 kbps maintenance channel for transmission of data loop back commands and detected transmission errors
  - –4B3T ternary block code (subscriber line symbol rate 120 kbaud)
  - –Monitoring of transmission errors
  - –Subscriber loop length without repeater:
    up to 4.2 km on 0.4 mm wire
    up to 8.0 km on 0.6 mm wire
- Adaptive echo cancellation
- Adaptive equalization
- Automatic polarity adaptation
- Clock recovery (frame and bit synchronization) in all applications

BLOCK DIAGRAM



Publication # 11152 Rev. B Amendment /0
Issue Date: March 1989

DISTINCTIVE CHARACTERISTICS (continued)

- Transposition of ternary to binary data and vice versa (coding, decoding, scrambling, descrambling, phase adaptation)
- Built-in wake-up unit for activation from power-down state
- Activation and deactivation procedure according to CCITT I.430 and to FTZ Guideline 1R 210 of the DBP
- Adaptation of internal interfaces to the current signal direction by programmable operation modes:
  LT: Line Terminator in public or private exchange
  NT: Network Terminator connected to SBC
  NT-PABX: Trunk Module (TMD)
  NT-TE: Terminal equipment
  LT-RP: U Repeater unit subscriber side
  NT-RP: U Repeater unit exchange side
- Optimized for working in conjunction with SBC via IOM interface
- Data speed conversion between the U_{K0} frames and the IOM frames. In the LT and NT-PABX modes absorption of received phase-wander of up to 18 µsec peak to peak (CCITT Rec. Q.512)
- Handling of commands and indications contained in the IOM C/I channel for (de-) activation, supervision of power supply unit and equipment for wire testing
- Data availability via the MONITOR channel:
  - –accumulated RDS transmission errors; in the LT mode for the whole U_{K0} link, in the NT mode only for those detected in the circuit itself
  - –measurement value of the loop current
  - –Echo canceller coefficients and status values, which can be used to give evidence of the state of "U" interface link
- Switching of an analog test loop at the U_{K0} interface for testing as many units of the IEC as possible (loop 1 in LT, loop 4 in the repeater and loop 3 in the NT-PABX mode in reverse direction to the public exchange)
- Switching of a digital test loop as near to IOM as possible (loop 2 in NT-PABX and NT-TE)
- Remote control of test loop switching via maintenance channel
  Test loop 2 in the SBC (NT mode)
  Test loop 2 in the IEC near to the IOM interface in the NT-PABX and TE modes
  Test loop 4 in the IEC LT-repeater mode near to the U-line
- Generation of a synchronized 7.68 MHz clock for the SBC in the mode

GENERAL DESCRIPTION

The Am20901/20902 ISDN Echo Cancellation Circuit (IEC) is an advanced CMOS circuit for transmission over the U_{K0} interface. The adaptive filter concept of the IEC is based on a highly digital approach which utilizes a sophisticated digital signal processing capability.

The Am20901/20902 enables digital full duplex voice/data transmission via the standard twisted pair telephone cable. (U interface) with a user bit rate of 144 kbps according to the ISDN standards. Together with the flexible IOM interface, it is fully compatible to operate with the Am2080 (SBC) devices and also enables a repeater (two IEC's back to back) for longer telephone loops.

The IEC is capable of operating in the following applications by means of pin strapping: the exchange, the network termination, the terminal equipment, and the trunk model connecting a PABX to the public network.

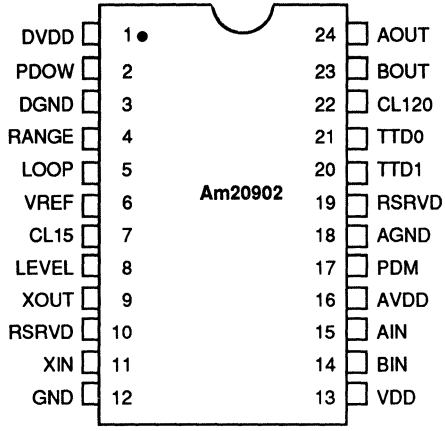
At the present the IEC is available in a two chip set only.

A "Digital" Circuit, called IEC-D (PEB 20901) contains the digital receiver functions and the IOM™ -U_{K0} interface functions.

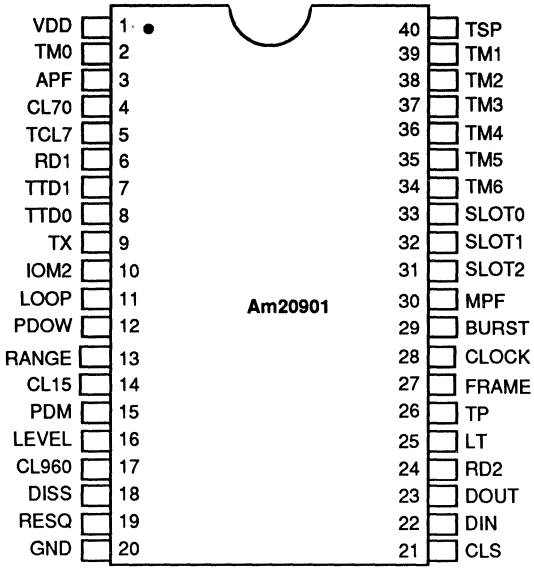
An "Analog" Circuit, called IEC-A (PEB 20902) contains the crystal oscillator and all of the analog functions of the line port, namely the A/D converter in the receive path and pulse shaping D/A converter and line driver in the transmit path.

CONNECTION DIAGRAMS
Top View

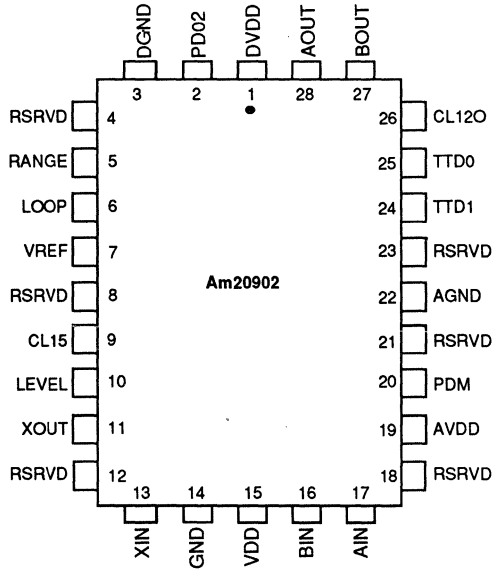
24-Pin DIP



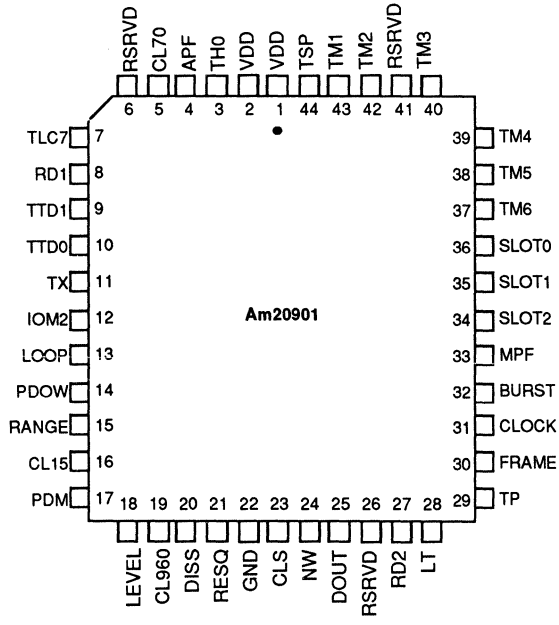
40-Pin DIP



28-Pin PLCC

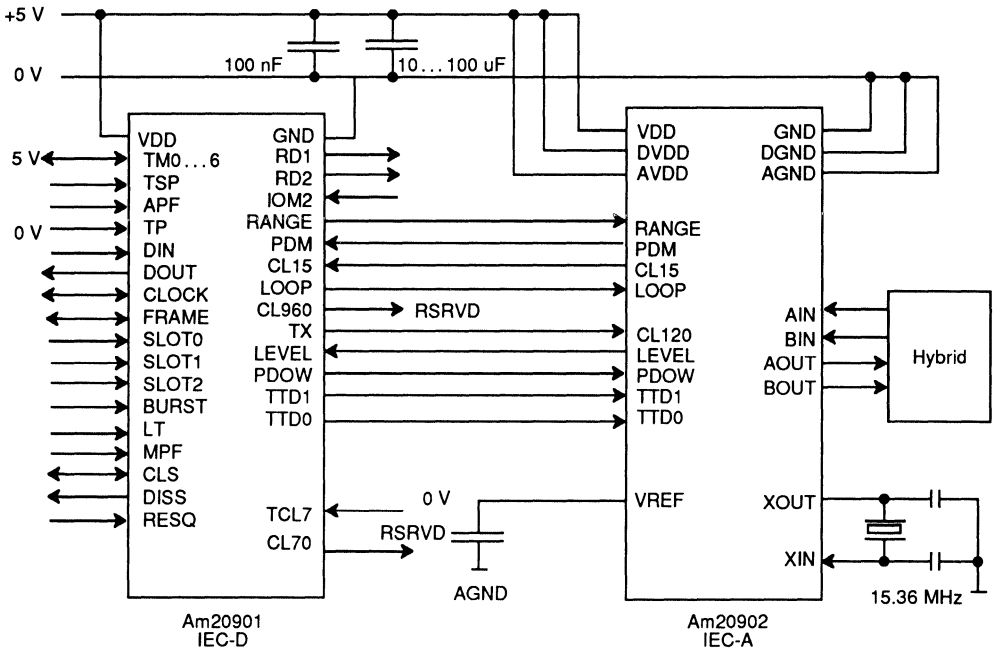


44-Pin PLCC



Note: Pin 1 is marked for orientation.

LOGIC SYMBOL

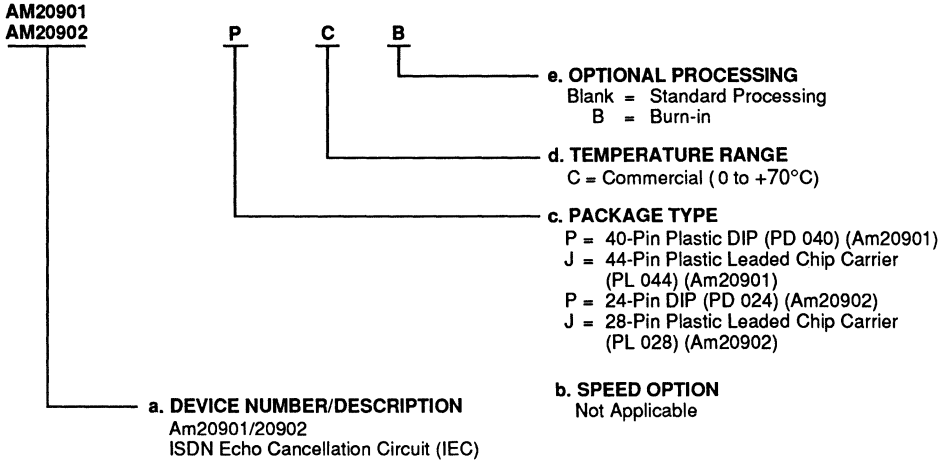


2

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM20901 AM20902	PC, JC, PCB, JCB PC, JC, PCB, JCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

*The Am2090 is a two-chip set. It is ordered as an Am20901 and Am20902.

Preliminary

Am7938

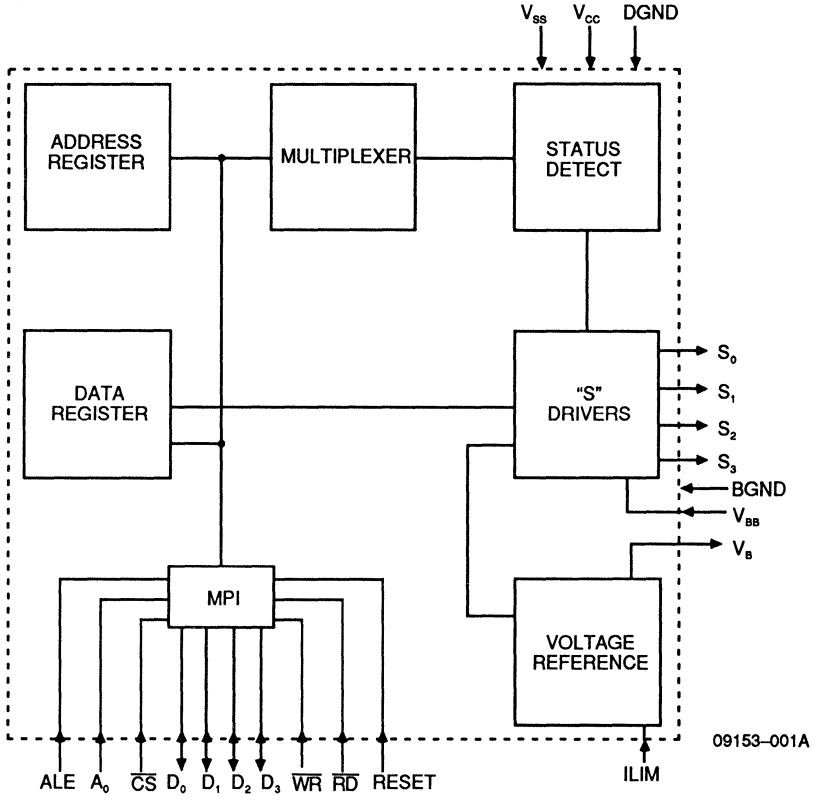
Quad Exchange Power Controller (QEPC)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Supplies power for up to four digital telephone lines
- Conforms to the CCITT recommendations for power feed at the "S"or "T" reference point
- Applications for intelligent NTs and PABX/Central Office line cards
- Supports point-to-point and point-to-multipoint configurations
- Built-in battery control circuit for –40 V operation
- Each of the four lines is individually controlled
- Status detectors for each line driver; open-loop, current-overload, low output voltage, thermal overload
- Programmable current limiting
- Automatic shutdown of overloaded lines
- Automatic thermal shutdown
- Microprocessor-compatible interface
- High-voltage bipolar technology allows battery voltages up to 65 V
- Output current up to 150 mA per driver

BLOCK DIAGRAM



Publication # 09153 Rev. C Amendment /0
Issue Date: April 1989

GENERAL DESCRIPTION

The Am7938 Quad Exchange Power Controller (QEPC) provides a power source for up to four line interfaces. The power source to the Am7938 is a local battery or a centralized regulated power supply. The Am7938 can reside in intelligent NTs or PABX/Central Office line cards. It can operate in point-to-point and point-to-multipoint configurations. Via the Am7938's microprocessor interface, each powered line is individually controlled and monitored. The power to each line can be switched off independently. Hence, overloads and faults are easy to detect and localize even in a large system. The status conditions detected by the Am7938 on each line which may be read by the microprocessor are: low output voltage, open-loop, current overload, thermal overload, and normal line conditions.

Current limit and thermal shutdown circuits protect the Am7938 against overload conditions. However, certain applications may require additional external protection circuitry.

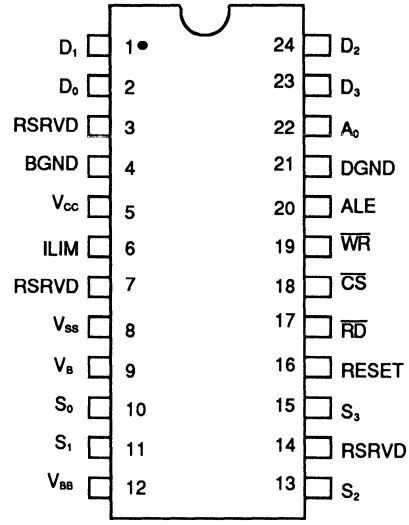
The Am7938 has been developed specifically for CCITT-compatible ISDN configurations. It should be recognized, however, that due to its versatile design, the Am7938 can be used in numerous other applications.

GLOSSARY OF ABBREVIATIONS

A_0	Address Bit
ALE	Address Latch Enable
BGND	Battery Ground (Battery refers to Telephone Line Supply)
CCITT	Consultative Committee for International Telegraph and Telephone
CO	Central Office
COD	Current-Overload Detector
$\overline{CS}$	Chip Select
D_0–D_3	Data Lines 0 through 3
DGND	Digital Ground
IAR	Indirect Address Register
ILIM	Current Limit
ISDN	Integrated Services Digital Network
LER	Line Enable Register
LVD	Low Voltage Detector
MPI	Microprocessor Interface
NT	Network Terminator
OLD	Open-Loop Detector
PABX	Private Automatic Branch Exchange
RESET	Reset
$\overline{RD}$	Read
RLIM	Current Limit Programming Resistor
RSRVD	Reserved
S	"S" Reference Point
S_0–S_3	S-Driver Lines 0 through 3
TE	Terminal Equipment
TOR	Thermal Overload Register
U	"U" Reference Point
V_B	Battery Control Voltage
V_{BB}	Battery Supply
$\overline{WR}$	Write

CONNECTION DIAGRAM
Top View

24-Pin DIP



Note: 1. Pin 1 is marked for orientation.
2. Reserved (RSRVD) pins should not be connected externally to any signal or supply.

2

ORDERING INFORMATION

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing

AM7938 **D** **C** **B**

e. OPTIONAL PROCESSING
Blank = Standard Processing
B = Burn-in

d. TEMPERATURE RANGE
C = Commercial (0 to + 70°C)

c. PACKAGE TYPE
D = 24-Pin Ceramic DIP (CD 024)

b. SPEED OPTION
Not Applicable

a. DEVICE NUMBER/DESCRIPTION
Am7938
Quad Exchange Power Controller (QEPC)

Valid Combinations	
AM7938	DC, DCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

A_0
Address Line (Input)

A_0 selects source and destination locations for read and write operations on the data bus. A_0 must be valid on the falling edge of ALE or during $\overline{RD}$ and $\overline{WR}$ if ALE is tied HIGH.

ALE
Address Latch Enable (Input; Active HIGH)

ALE is an input control pulse used to strobe the address on the A_0 line into the address latch. This signal is active HIGH to admit the input address. The address is latched on the HIGH-to-LOW transition of ALE. While ALE is HIGH, the address latch is transparent. For an unmultiplexed microprocessor bus, ALE must be tied HIGH.

BGND
Ground, Battery

Regulated output on V_B is referenced to this ground.

$\overline{CS}$
Chip Select (Input; Active LOW)

$\overline{CS}$ must be LOW to enable the read or write operations of the Am7938. Data transfer occurs over the D_3–D_0 lines. The interaction of $\overline{CS}$, $\overline{RD}$, $\overline{WR}$, and D_3–D_0 is described below.

D_3–D_0
Data Bus (Input/Output; Three State, Active HIGH)

The four bidirectional data bus lines are to exchange information with a microprocessor. D_0 is the least significant bit and D_3 is the most significant bit. A HIGH on the data bus corresponds to a logical "0". These lines act as inputs when $\overline{WR}$ and $\overline{CS}$ are active and as outputs when $\overline{RD}$ and $\overline{CS}$ are active. When $\overline{CS}$ is inactive, the D_3–D_0 pins are placed in a high-impedance state.

DGND
Ground, Digital

Digital Ground to logic.

ILIM
Current Limit Programming (Input)

ILIM programs the current limit of the "S" drivers using an external resistor connected between ILIM and V_{SS}. The ILIM pin is 1.25 V more positive than V_{SS}. The current limit is 5 mA plus 1000 times the current in the external resistor to ground. The programmed current limit applies to each "S" driver.

$\overline{RD}$
Read (Input; Active LOW)

The active-LOW read signal is conditioned by $\overline{CS}$ and transfers internal information to the data bus. If A_0 is a logical "0", logic levels of the Indirect Address Register (IAR) and Thermal Shutdown Status bit will be transferred to D_0–D_3. If A_0 is a logical "1", the data addressed by the IAR will be transferred to D_0–D_3.

RESET
Reset (Input; Active HIGH)

RESET initializes the registers in the Am7938, leaving the "S" drivers switched off.

S_3–S_0
"S" Drivers (Output)

S_3–S_0 each supply power to one line. The outputs can sink up to 150 mA each. The voltage at the line is connected to V_{BB} through a saturated transistor switch.

V_B
Battery Reference Voltage (Output)

V_B provides an output proportional to the deviation of V_{BB} from an internal –40 V reference with respect to BGND. V_B can be used as a driver for an external PNP Darlington power transistor supplying power from the battery to V_{BB}. See Figure 2.

V_{BB}
Supply Voltage "S" Driver (Input)

V_{BB} supplies power to the "S" drivers.

V_{CC}
+5 V Power supply (Input)

V_{SS}
Substrate Voltage (Input)

V_{SS} is the internal negative supply voltage. V_{SS} must always be connected to the most negative supply voltage.

$\overline{WR}$
Write (Input; Active LOW)

The active-LOW write signal is conditioned by $\overline{CS}$ and transfers information from the data bus to an internal register selected by A_0. If A_0 is a logical "1", D_0–D_3 is written into the Line Enable Register (LER). If A_0 is a logical "0", D_0–D_2 is written into the IAR. LER and IAR are the only two writable registers in the Am7938.

FUNCTIONAL DESCRIPTION

The Am7938 is divided into two sections, the Analog section and the Microprocessor Interface (MPI) section. The analog section provides power and detects the status of the "S" lines. This status information is available to the microprocessor via the MPI.

Initialization

The Am7938 is initialized when reset by an external signal applied to the RESET pin. In the initialized state the analog drivers are switched off.

–40 V Power

The voltage at the "S" drivers is approximately V_{BB} (less VSAT). A regulated –40 V "S" output can be achieved by using the V_B pin to drive external Darlington power transistor.

Analog Section

The major functions of the Analog section are the four line drivers, which are saturated Darlington transistor switches capable of sinking up to 150 mA each. The power to the drivers is derived from the negative supply voltage (V_{BB}) to the Am7938. The output voltage to each line is slaved to V_{BB}, and the voltage drop in each driver is approximately 1.5 V.

Line driver protection is provided through the integration of current limit and over-temperature shut-off. The current limit is hardware-programmable via an external resistor (RLIM) connected between ILIM and V_{SS}. The ILIM pin is 1.25 V more positive than V_{SS}. The output limit is: 5 mA + 1000 x 1.25 V/RLIM.

The thermal shut-off is internally set at approximately 150° C. At this temperature all the drivers are unconditionally switched off. However, at approximately 140° C, only the drivers that are in the current-overload condition will be turned off.

Status detectors, associated with each of the line drivers, monitor the load conditions on each line by comparing an electrical parameter (e.g., current and voltage at the line) with a reference level. The output of each detector can be read by the microprocessor. In addition to these status detectors, the temperature of the Am7938 is monitored via integrated temperature detectors. The detectors respond at approximately 140° C and 150° C, as defined above, and the 150° C detector can be monitored by the microprocessor via the MPI. The status detectors provide the following information from each of the lines (all detectors have built-in hysteresis):

Low Output Voltage Detection

The low output voltage status bit becomes active when the output transistor is pulled out of saturation.

Open-Loop Detection

The open-loop status bit becomes active when the current on the line drops below a minimum value.

Current-Overload Detection

The current-overload status bit becomes active when the current on the line nears the current limit.

Thermal Overload Detection

If the Am7938 device temperature reaches 140° C, then all the line drivers in the current-overload condition will be switched off and the corresponding bits in the Thermal Overload Register will be cleared. If the device temperature increases to 150° C, all the line drivers will be turned off, and all the bits in the Thermal Overload Register will be cleared. The T-bit will be set, and it can be read along with the Indirect Address Register (IAR) to indicate that all the drivers have been turned off. To initialize any of the bits in the Thermal Overload Register, the microprocessor must first turn off on the line drivers via the Line Enable Register (LER) (see MPI definition). The line drivers must not be reactivated until the T-bit in the address register is cleared by the temperature detector in the Am7938.

MPI Section

The MPI allows the user to access the detectors defined in the Analog section. The line driver's status bits are grouped by function. Bits 0 through 3 of the detectors correspond to lines 0 through 3, respectively. The status groups are:

Low Voltage Detector (LVD)
Open-Loop Detector (OLD)
Current-Overload Detector (COD)
Thermal Overload Register (TOR)

The data is not latched in these status groups except in the TOR. Thus, the user should filter (multiple samples) the received data to ensure its integrity. There are two other registers in the MPI: the Indirect Address Register (IAR), and Line Enable Register (LER).

The IAR contains 3 bits which address the desired status group or the LER. The IAR is read along with the T-bit defined in the Analog section. The microprocessor can read the IAR to check the validity of the address.

The LER is used to enable or disable the individual line drivers. The line drivers will only become active if the corresponding bit in the TOR is inactive. The LER is a read/write register. The MPI contains the interface to the following pins:

D_3–D_0	Bidirectional	Data Bus
A_0	Input	Address Line
ALE	Input	Address Latch Enable
$\overline{RD}$	Input	Read Enable
$\overline{WR}$	Input	Write Enable
$\overline{CS}$	Input	Chip Select

The 4-bit bidirectional data bus (D_3–D_0) is used to communicate with the registers. Access to the registers is controlled by $\overline{CS}$, $\overline{RD}$, $\overline{WR}$, ALE, and A_0 as shown below. A read or write cycle must be preceded by a valid A_0. A_0 is latched internally in a transparent latch by ALE. The selection of the status group or the LER is determined by the content of the IAR. Following a write to the IAR, a delay of at least 1 µs must be provided for settling before the next read operation. Subsequent reads from the status group or the LER do not have this restriction.

The truth table for the MPI control is shown below:

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_0	
0	1	0	0	Write IAR (T bit is read only)
0	0	1	0	Read IAR and T bit
0	1	0	1	Write LER
0	0	1	1	Read status groups or LER
1	X	X	X	No access

Indirect Address Register (IAR) and T bit

The IAR is 3 bits wide and accessible through the data port, D_2–D_0. The content of the Indirect Address Register (IAR_2–IAR_0) determines the selection of the status groups or the LER. The thermal overload bit "T" is read at the same time as IAR and occupies D_3. This register has the following format:

Bit	Symbol	
0	IAR0	Bit 0 of the IAR
1	IAR1	Bit 1 of the IAR
2	IAR2	Bit 2 of the IAR
3	T	T bit (Read only) Logical "0": temperature normal (default value) Logical "1": temperature above 150° C (all drivers shut off)

IAR_2–IAR_0 address the status groups and the LER as shown below:

IAR2	IAR1	IAR0	Select
0	0	0	LVD
0	0	1	OLD
0	1	0	COD
0	1	1	Reserved
1	0	0	Reserved
1	0	1	Reserved
1	1	0	LER
1	1	1	TOR

The contents and format of the status groups and the LER are as follows:

LVD:

Bit	Logical "1"	Logical "0" (default value)
0	"S0" low voltage	"S0" voltage normal
1	"S1" low voltage	"S1" voltage normal
2	"S2" low voltage	"S2" voltage normal
3	"S3" low voltage	"S3" voltage normal

The Low Voltage Detector (LVD) indicates the voltage level on the "S" lines. The low voltage condition becomes active (logical "1") if the output transistor is pulled out of saturation (VSOL).

OLD:

Bit	Logical "1"	Logical "0" (default value)
0	"S0" open-loop	"S0" current normal
1	"S1" open-loop	"S1" current normal
2	"S2" open-loop	"S2" current normal
3	"S3" open-loop	"S3" current normal

The Open-Loop Detector (OLD) indicates the open-loop condition on the "S" lines. The open-loop condition becomes active (logical "1") if the current on the line drops below the threshold value ISOC.

COD:

Bit	Logical "1"	Logical "0" (default value)
0	"S0" current-overload	"S0" current normal
1	"S1" current-overload	"S1" current normal
2	"S2" current-overload	"S2" current normal
3	"S3" current-overload	"S3" current normal

The Current-Overload Detector (COD) indicates the current-overload condition on the "S" lines. The overload condition becomes active (logical "1") if the output current approaches the value programmed by an external resistor between ILIM and V_{SS}.

TOR:

Bit	Logical "1" (default value)	Logical "0"
0	"S0" operational	"S0" off
1	"S1" operational	"S1" off
2	"S2" operational	"S2" off
3	"S3" operational	"S3" off

The Thermal Overload Register (TOR) contains the overload status of the "S" line drivers. If the Am7938 device temperature reaches 140° C, then the "S" line drivers which are in the current-overload condition will be switched off. The corresponding bits in the TOR will be set to a logical "0". To initialize any of the bits in the TOR, the microprocessor must first turn off the "S" line drivers via the LER. However, the TOR bits cannot be deactivated if the 150° C detector is active. The µP may re-enable the "S" drivers via the LER after the TOR condition is removed. The TOR is a read-only register.

LER:

Bit	Logical "1"	Logical "0" (default value)
0	"S0" on	"S0" off
1	"S1" on	"S1" off
2	"S2" on	"S2" off
3	"S3" on	"S3" off

The Line Enable Register (LER) is used to enable or disable the individual "S" line drivers. The "S" line will only become active if the corresponding bit in the TOR is set to a logical "1". The LER can be written directly and read indirectly.

APPLICATIONS

The Am7938 major applications are in intelligent NTs, and in PABX and Central Office line cards. The applications show the Am7938 interfaced to the CCITT "S" interface point; however, the Am7938 can be used to control the power feed for any transformer coupled system, i.e., the CCITT "U" interface point.

For inductive loads it is advisable to connect .01 µF between each output pin and V_{SS}, and connect diodes as shown in Figure 5 .

The MPI Registers will not function properly when the battery power is disconnected, i.e., when V_{SS} is floating or grounded.

The Am7938 Used With a Regulated Supply (Figure 1)

This diagram shows the Am7938 used with a regulated supply voltage of $V_{BB} = -40$ V ±5%. The output voltage from the "S" drivers (because of the voltage drop across the internal transistors) will be VSAT more positive than V_{BB}. The V_B pin is unused; V_{SS} is tied to V_{BB}. Other battery voltages can be used. The Am7938 can power up to eight TEs per "S" line provided the total current per line does not exceed 150 mA.

Recommended Decoupling:
0.1 µF from V_{CC} to DGND
1.0 µF from V_{SS} to DGND
V_{SS} shorted to V_{BB}
DGND shorted to BGND

The Am7938 Used With an External Power Transistor (Figure 2)

Power to the Am7938 is supplied from the local battery with an external power transistor to dissipate power from the Am7938. The Am7938 allows battery voltages up to 65 V provided the external power transistor can handle the power dissipation.

Recommended Decoupling:
0.1 µF from V_{CC} to DGND
1.0 µF from V_{SS} to DGND
0.1 µF from V_{BB} to DGND
0.1 µF from V_B to V_{SS} (across transistor B–C)
DGND shorted to BGND

The Am7938 Employing a Relay to Provide Polarity Reversal (Figure 3)

The CCITT recommends that polarity reversal be used to resolve the power contention in a point-to-multipoint configuration when local power is lost. In Figure 3, polarity reversal is implemented using the Am7938 and one additional relay. The coil and contacts of the relay are connected so that the relay is activated when local power is available, and is deactivated when local power is lost. Hence, power consumption is minimized when local power is lost.

An alternative solution for power contention resolution is to power only a few lines when local power is lost. In the Am7938, each line is independently controlled via the microprocessor through the LER, hence, the microprocessor can turn off non-priority lines as required.

Four Am7938s Interfaced to a Microprocessor Bus (Figure 4)

This diagram illustrates four Am7938s interfaced to a microprocessor. Only the lower four data lines from the microprocessor's 8-bit data bus are used. These are connected to the Am7938s D_3–D_0 pins. In a non-multiplexed microprocessor bus system ALE must be tied to a logical "1".

Protection of the QEPC Against Overvoltage (Figure 5)

In Central Office (CO) or long-line PABX applications, the QEPC requires additional external protection against lightning and voltages induced from the power lines.

Figure 5 shows how the QEPC and the system is protected via an overvoltage-limiting device and fuse resistors.

The voltage-limiting device limits the voltage on the "S" line to a safe value, and the fuse resistors break the circuit if a continuous high-power source is connected to the "S" line. The diode in the battery line prevents the line surges from going below the substrate voltage.

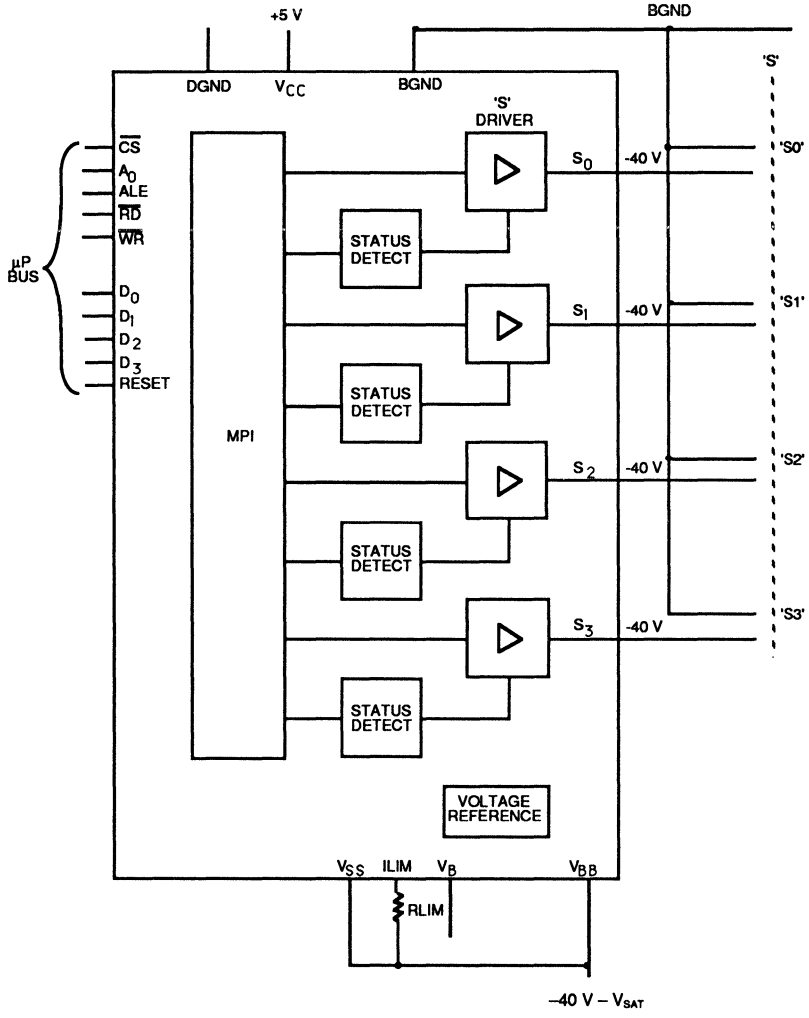


Figure 1. The Am7938 Used With a –40 V Regulated Supply
(other battery/S_{out} voltages can be used)

09153–003A

2

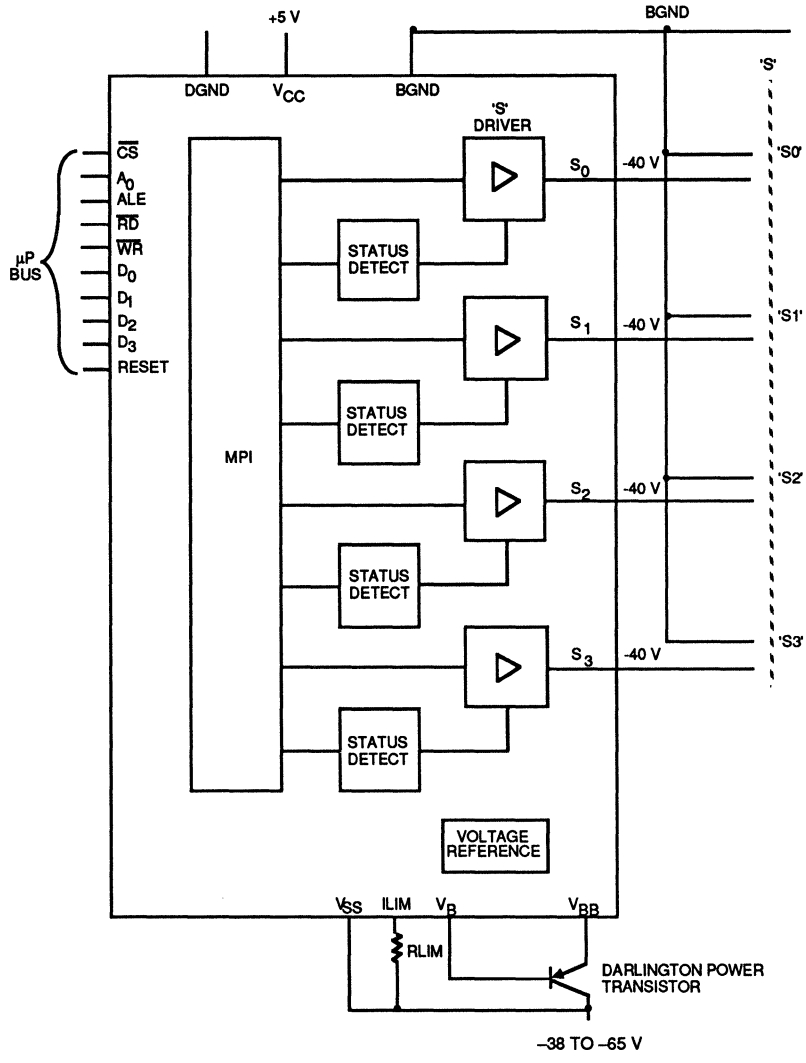


Figure 2. The Am7938 Used With an External Power Transistor 09153-004A

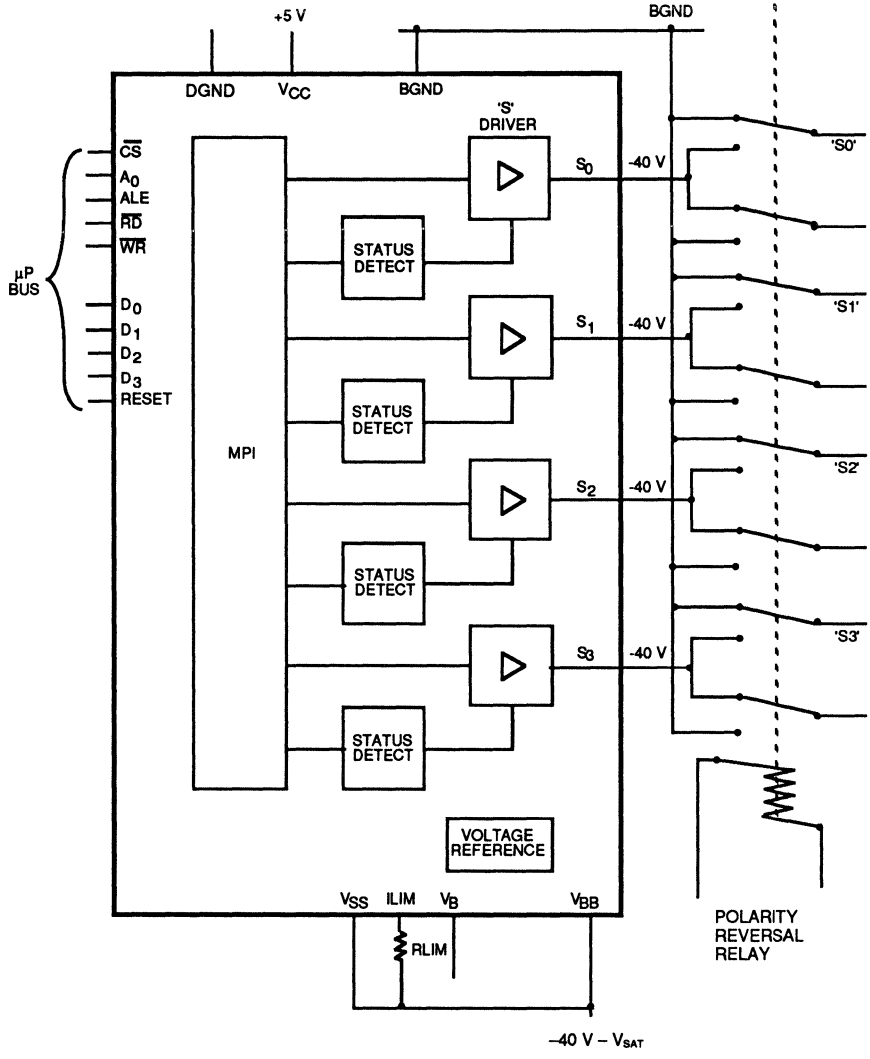


Figure 3. The Am7938 Employing a Relay to Provide Polarity Reversal

09153-005A

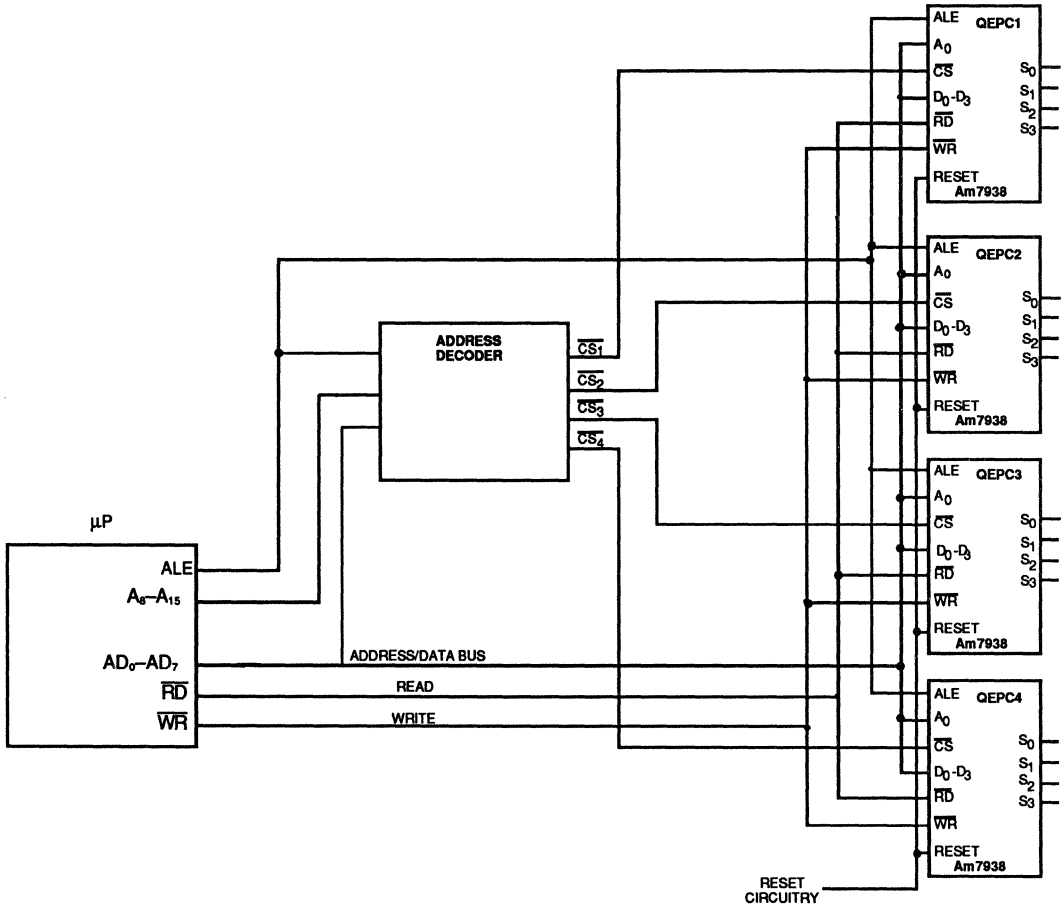


Figure 4. Four Am7938s Interfaced to a Microprocessor Bus 09153–006A

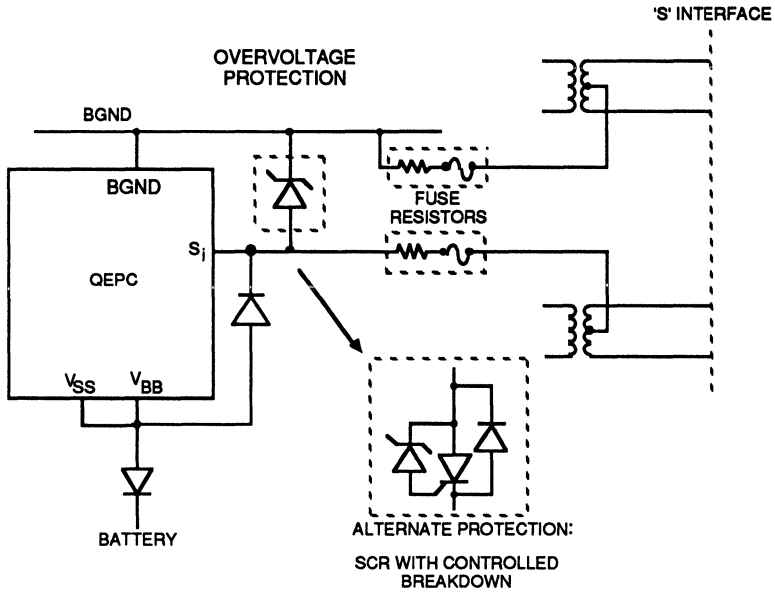
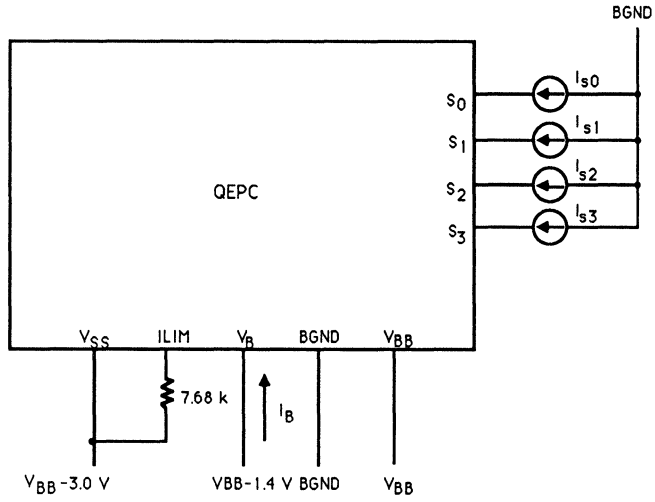


Figure 5. Protection of the QEPC Against Overvoltage

09153-007A



I_{si}= Current at 'S' Output Driver i, where i = 0–3

Figure 6. Current Into V_B

09153-008A

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... –60°C to +150°C
Voltage from Digital Input
to DGND ..... –0.4 V to V_{CC}
Voltage from V_{CC}
to DGND ..... –0.4 V to 7 V
Voltage from V_{SS}
to DGND ..... –70 V to +0.4 V
Voltage from V_{BB}
to DGND ..... (V_{SS} –0.4 V) to +0.4 V
..... and (V_{CC} –70 V) to +0.4 V

100 ns Pulse Voltage from Si
to DGND (Note 1 and 3) ..... –90 V to + 2 V
Voltage from BGND
to DGND ..... V_{SS} to V_{CC}

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES (Note 2)

Commercial (C) Devices
Ambient Temperature(T_A) ..... 0°C to +70°C
DGND ..... 0 V
BGND Voltage ..... –2 V to + 0.5 V
V_{CC} Voltage ..... +5 V ± 5%
V_{BB} Voltage ..... –65 V to –38 V
V_{SS} Voltage, V_B Unused ..... V_{BB}
V_{SS} Voltage, V_B Used ..... –65 V to (V_{BB} –3 V)

Operating ranges define those limits between which the functionality of the device is guaranteed.

Notes:
1. Si stands for S_0, S_1, S_2, or S_3 outputs.

2. Operation at smaller V_{BB} magnitudes is possible, but parameters are not guaranteed.

3. The test condition is specified with a diode in series with V_{SS} as shown in Figure 5.

DC CHARACTERISTICS OVER COMMERCIAL OPERATING RANGE

Parameter Symbol	Parameter Description	Test Conditions	Min.	Typ	Max.	Unit
V_{IH}	Logic Input HIGH Level		2.0			V
V_{IL}	Logic Input LOW Level				0.8	V
I_{OH}	Logic Output HIGH Current	(V_{OH} = 2.4 V)	400			µA
I_{OL}	Logic Output LOW Current	(V_{OL} = 0.4 V)	2.0			mA
I_{IH}	Logic Input HIGH Current	(V_{IH} = 2.0 V)			100	µA
I_{IL}	Logic Input LOW Current	(V_{IL} = 0.8 V)			250	µA
I_{OZH}	Output Hi-Z Current HIGH	(2.4V < V_{OZ} < V_{CC})			100	µA
I_{OZL}	Output Hi-Z Current LOW	(0V < V_{OZ} < 0.4 V)			250	µA
I_{CC}	VCC Supply Current			12	TBD	mA
C_L	Logic Input/Output Capacitance			10		pF
V_{SAT}	Saturation Voltage	(V_{Si} - V_{BB}),				
		ISi = 150 mA			2.0	V
I_B	Current Into V_B (see Figure 6)	V_{BB} = –42 V			50	µA
		V_{BB} = –38 V	1200			µA
I_{SS}	V_{SS} Supply Current,	V_{SS} = –65 V,				
		V_{BB} = –40 V		2.8	TBD	mA
I_{BB}	V_{BB} Supply Current, V_{BB} = –42 V	(Outputs Disabled)		1.2	TBD	mA
I_{SLIM}	Limit Current	RLIM = 8.62K	TBD	150	TBD	mA
I_{SLIM}	Limit Current	RLIM = 27.8K	TBD	50	TBD	mA
V_{SOL}	Voltage Overload Detection					
	Relative to V_{BB}			3	TBD	V
I_{SOL}	Current Overload Detection/ILIM		TBD	90	TBD	%
I_{SOC}	Current at Open-Circuit Detection		TBD	5	TBD	mA
I_{SZ}	Si Current to ground, Si disabled	Si is i-th output)			TBD	µA

SWITCHING CHARACTERISTICS OVER OPERATING RANGE

Parameter Number	Parameter Symbol	Parameter Description	Min.	Max.	Unit
Microprocessor Read/Write Timing					
1	t_{RLRH}	$\overline{RD}$, $\overline{CS}$ Pulse Width	200		ns
2	t_{RHRL}	$\overline{RD}$ Recovery Time	200		ns
3	t_{RLDA}	$\overline{RD}$, $\overline{CS}$ LOW to Data Availabe		200	ns
4	t_{RHDZ}	$\overline{RD}$ or $\overline{CS}$ HIGH to Data Hi Z		110	ns
5	t_{AHAL}	ALE Pulse Width	60		ns
6	t_{ADAL}	Address Setup Time	35		ns
7	t_{ADAZ}	Address Hold Time	35		ns
8	t_{AZRL}	Address Hi Z to $\overline{RD}$ LOW	0		ns
9	t_{WLWH}	$\overline{WR}$ or $\overline{CS}$ Pulse Width	200		ns
10	t_{WHWL}	Write Recovery Time	200		ns
11	t_{DAWH}	Data Available to $\overline{WR}$ or $\overline{CS}$ HIGH	100		ns
12	t_{WHDZ}	$\overline{WR}$ or $\overline{CS}$ HIGH to Data Hi Z	100		ns
13	t_{MUX}	IAR Write to Detector Read	1000		ns
Reset Timing					
14	t_{RES}	Reset Pulse Width	200		ns
15	t_{PHRL}	Power Stable to Reset LOW	1		µs

Note:
All switching characteristics tests are performed with a 100 pF test load.

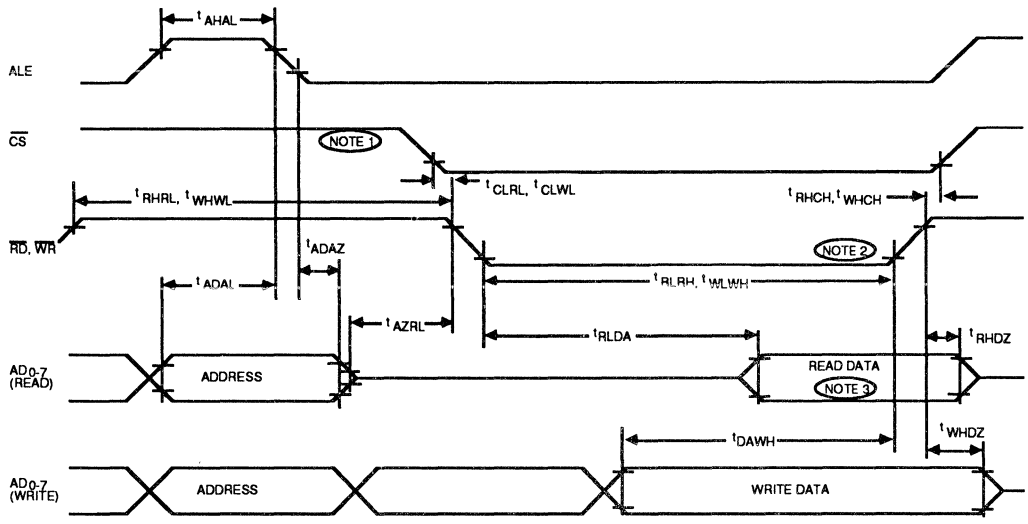


Figure 7. Microprocessor Read/Write Timing 09153–009A

Notes:

1. If t_{CLRL} is negative, t_{RHRL}, t_{RLRH}, t_{AZRL} and t_{RLDA} are measured from $\overline{CS}$ rather than $\overline{RD}$. If t_{CLWL} is negative, t_{WHWL} and t_{WLWH} are measured from $\overline{CS}$ rather than $\overline{WR}$.
2. If t_{RHCH} is negative, t_{RHRL}, t_{RLRH} and t_{RHDZ} are measured from $\overline{CS}$ rather than $\overline{RD}$. If t_{WHCH} is negative, t_{WHWL} and t_{WLWH}, t_{DAWH} and t_{WHDZ} are measured from $\overline{CS}$ rather than $\overline{WR}$.
3. When a read from a status group or the LER immediately follows a write to the IAR, a minimum of 1 μs is required between these operations. This is to allow the internal buffers in the Am7938 to settle. Subsequent reads from the status group or the LER do not have this restriction.

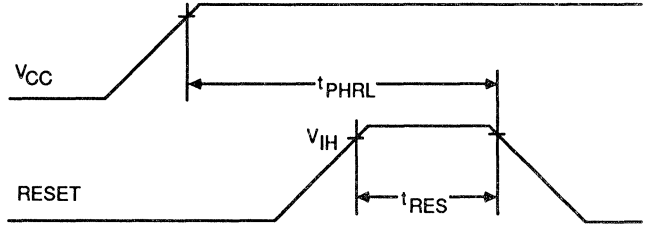
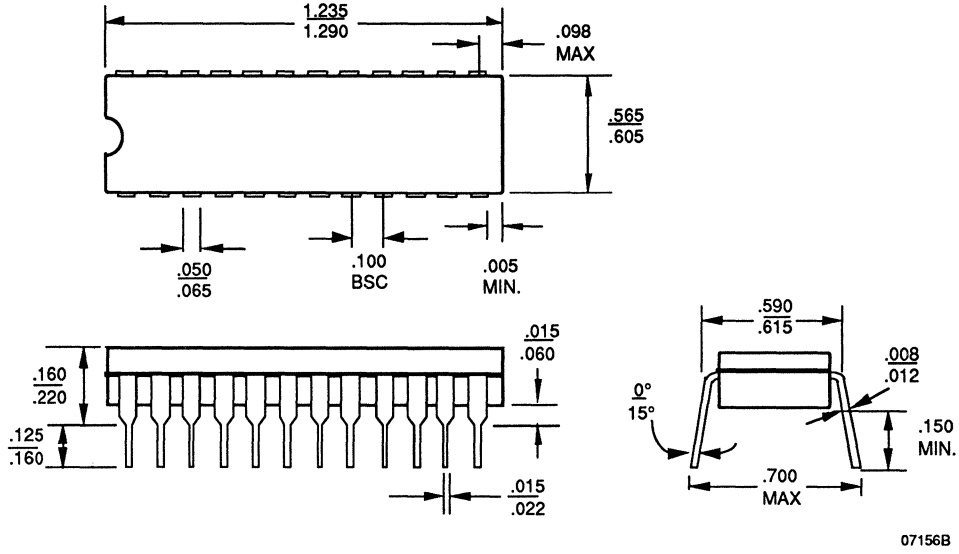


Figure 8. Reset Timing 09153–010A

PHYSICAL DIMEMSIONS
CD 024



Preliminary

Am82525

High-Level Serial Communications Controller Extended (HSCX)

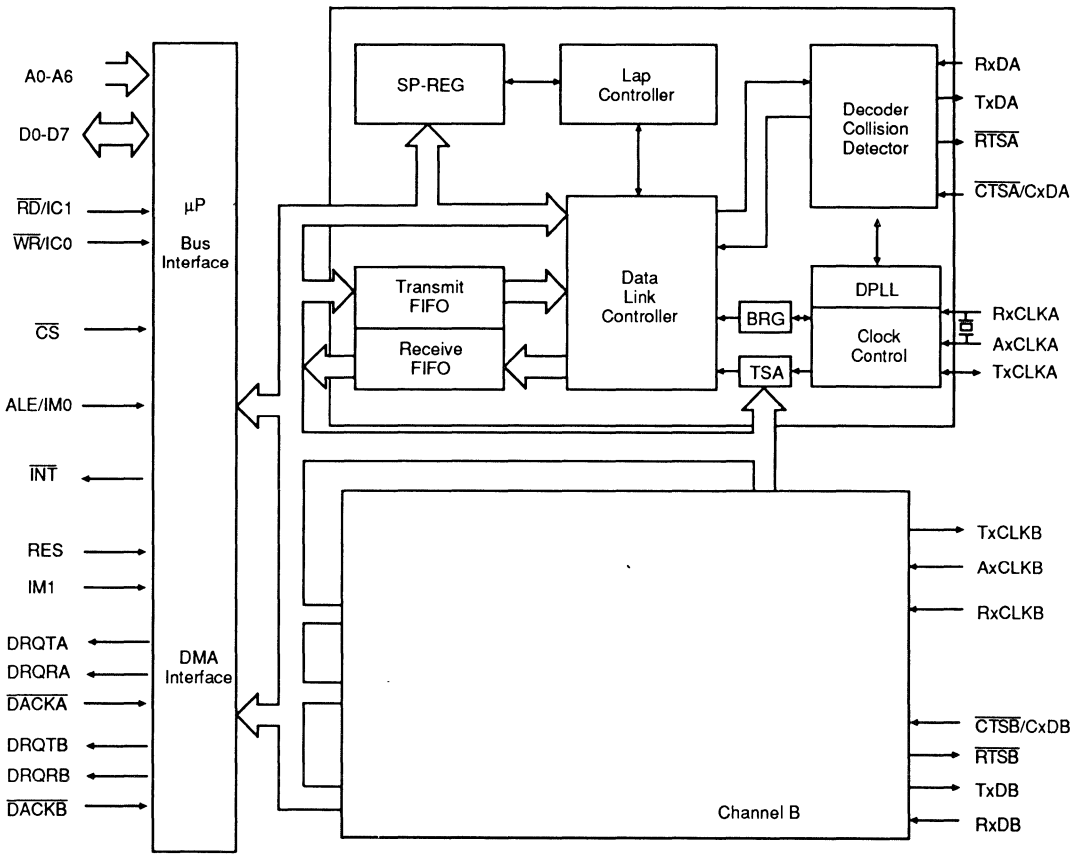
Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

Serial Interface

- Two independent full-duplex HDLC channels
- On-chip clock generation or external clock source
- On-chip DPLL for clock recovery for each channel
- On-chip baud rate generators for each channel
- Independent time slot assignment for each channel with programmable time slot length (1–256 bit)
- Different modes of data encoding (NRZI, NRZ)
- Modem control lines ($\overline{RTS}$, $\overline{CTS}$, $\overline{CD}$)
- Support of bus configuration by collision resolution

BLOCK DIAGRAM



11139-001B

Publication # 11139 Rev. B Amendment /0
Issue Date: May 1989

2

DISTINCTIVE CHARACTERISTICS (Continued)

- Programmable bit inversion
- Transparent receive/transmit of data without HDLC framing
- Cyclic transmission mode (1 to 32 bytes possible)
- Data rate up to 4 Mbps

Protocol Support

- Auto mode
- Non-auto mode
- Transparent mode
- Handling of bit-oriented functions in all modes
- LAPB/LAPD/SDLC/HDLC procedural support in auto mode (I- and S-frame handling)
- Modulo 8 or modulo 128 operation
- Programmable timeout and retry conditions
- Programmable maximum packet size checking

Microprocessor Interface

- 64-byte FIFOs per channel and direction
- Storage capacity of up to 17 short frames in receive direction
- Efficient transfer of data blocks from/to system memory by DMA or interrupt request
- 8-bit demultiplexed or multiplexed bus interface
- Intel or Motorola type microprocessor interface

General

- Compatible to Am82520 (HSCC)
- Advanced CMOS technology
- Low power consumption
  - active 25 mW at 4 MHz
  - standby 4 mW

GENERAL DESCRIPTION

The Am82525 HSCX is a High-Level Serial Communications Controller with extended features and functionality. The HSCX is compatible to the Am82520 HSCC.

The HSCX has been designed to implement high-speed communication links using HDLC protocols and to reduce the hardware and software overhead needed for serial synchronous communications.

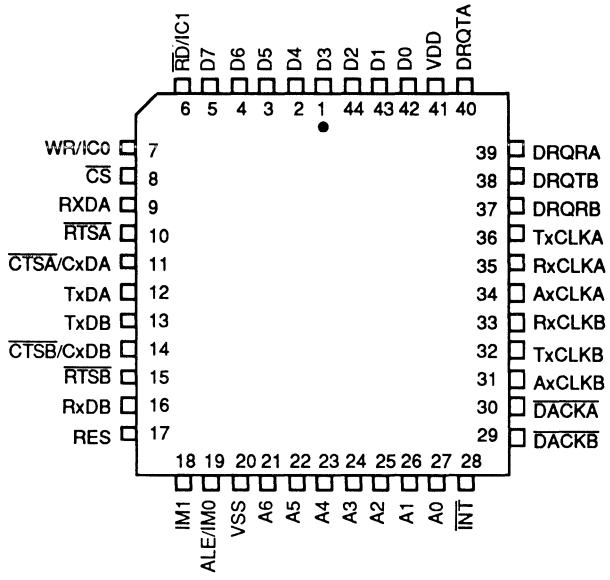
Due to its 8-bit adaptive bus interface it fits into every AMD/Siemens/Intel or Motorola 8- or 16-bit microcontroller or microprocessor system. The data throughput from/to system memory is optimized for transferring blocks of data (up to 32 bytes) by means of DMA or interrupt request. Together with the storing capacity of up to 64 bytes in on-chip FIFOs, the serial interfaces are effectively decoupled from the system bus which drastically reduces the dynamic load and reaction time of the CPU.

The HSCX directly supports the X.25 LAPB, the ISDN LAPD, and SDLC (Normal Response Mode) protocols and is capable of handling a large set of OSI layer 2 protocol functions independently from the host processor. Furthermore, the HSCX opens a wide area for applications which use time division multiplex methods; e.g, time-slot oriented PCM systems, systems designed for packet switching, and ISDN applications, by its programmable telecom-specific features.

The HSCX is available in a 44-pin PLCC package.

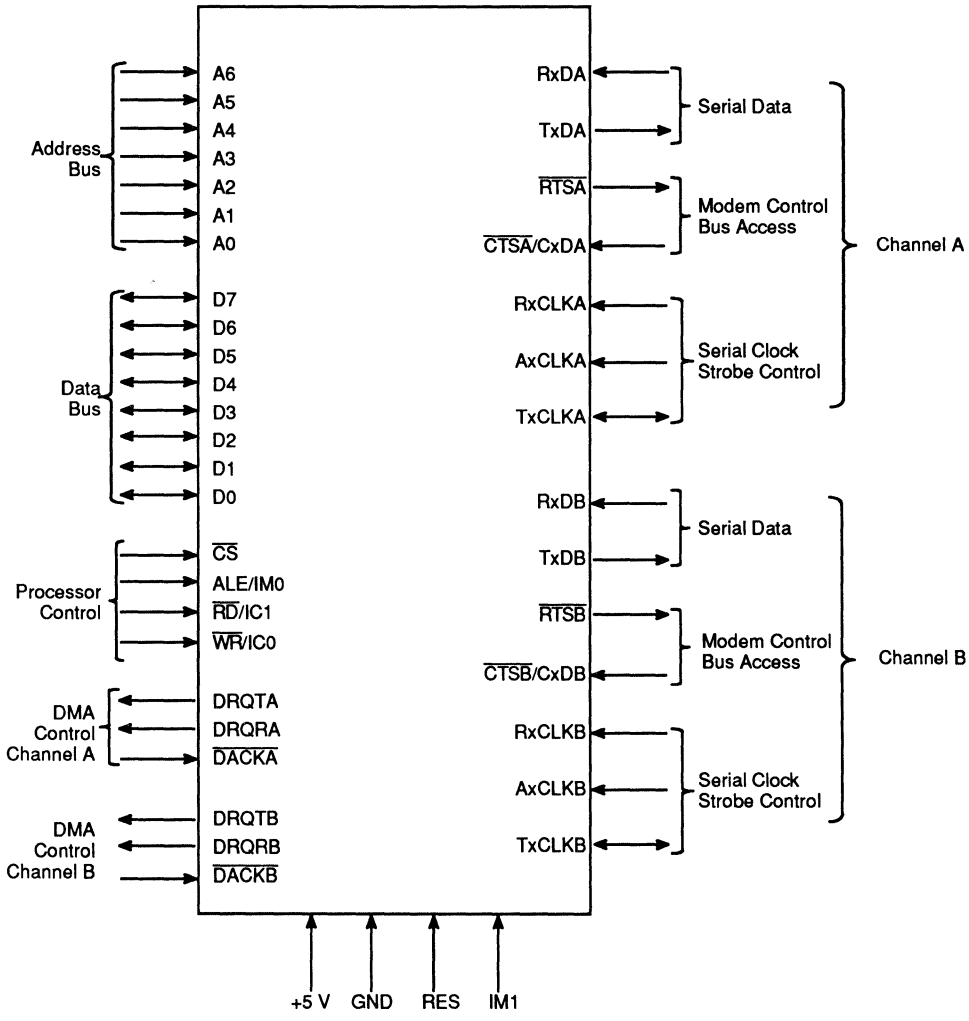
CONNECTION DIAGRAM
Top View

44-Pin PLCC



Note: Pin is marked for orientation.

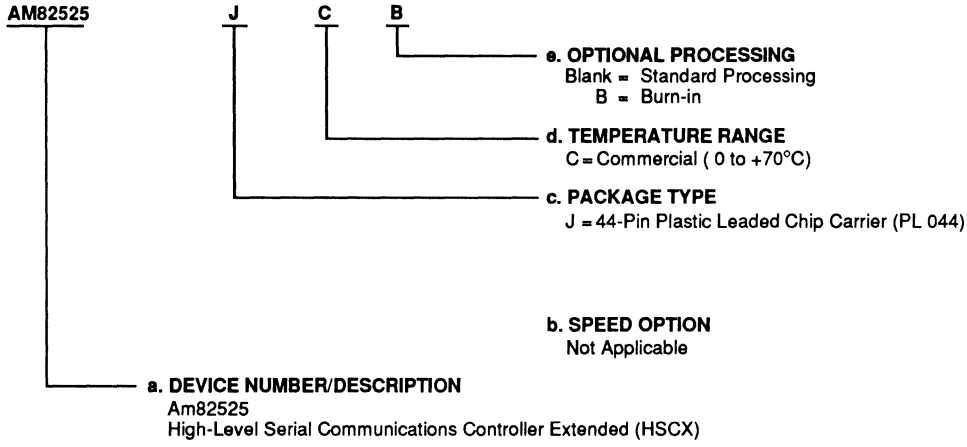
LOGIC SYMBOL



ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM82525	JC, JCB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

D0–D7

Data Bus (Input/Output)

The Data Bus lines are bi-directional three-state lines which interface with the system's Data Bus. These lines carry data and commands status to and from the HSCX.

$\overline{RD}$/IC1

Read, Intel Bus Mode (IM1 connected to LOW) (Input)

This signal indicates a Read operation. When the HSCX is selected via $\overline{CS}$, the $\overline{RD}$ enables the bus driver to put data from an internal register addressed by A0–A6 on the Data Bus.

When the HSCX is selected for DMA transfers with $\overline{DACK}$, the $\overline{RD}$ signal enables the bus drivers to put data from the respective Receive FIFO on the Data Bus, and Inputs to A0–A6 are ignored.

INPUT CONTROL 1, Motorola Bus Mode (IM1 connected to HIGH.)

If Motorola Bus Mode has been selected, this pin serves as one of the following inputs (depending on the selection with IM0) to control Read/Write operations:
E = Enable, active HIGH (IM0 tied LOW) or
$\overline{DS}$ = Data Strobe, active LOW (IM0 tied HIGH)

$\overline{WR}$/IC0

Write, Intel Bus Mode (Input)

This signal indicates a Write operation. When $\overline{CS}$ is active the HSCX loads an internal register with data provided from the Data Bus. When $\overline{DACK}$ is active for DMA transfers, the HSCX loads data from the Data Bus on the top of the respective transmit FIFO.

INPUT CONTROL 0, Motorola Bus Mode

In Motorola Bus Mode, this pin serves as the R/W input to distinguish between Read or Write operations.

$\overline{CS}$

Chip Select (Input)

A LOW signal selects the HSCX for a Read/Write operation.

RXDA, RXDB

Receive Data (Channel A/Channel B) (Input)

Serial data is received on these pins at standard TTL or CMOS levels.

$\overline{RTSA}$, $\overline{RTSB}$

Request to Send (Channel A/Channel B) (Input)

When the $\overline{RTS}$ bit in the MODE register is set, the $\overline{RTS}$ pin goes LOW. When the $\overline{RTS}$ bit is reset, $\overline{RTS}$ goes HIGH if the transmitter has finished and there is no further request for transmission.

In the bus configuration, this pin can be programmed using CCR2 to:

–go LOW during the actual transmission of a frame shifted by one clock period, excluding collision bits.
–go LOW during the reception of data frame.
–stay always HIGH ($\overline{RTS}$ disabled).

$\overline{CTSA}$/CXDA

Clear to Send (Channel A/Channel B) (Input)

A LOW on the $\overline{CTS}$ input pin enables the respective transmitter. Additionally, an interrupt may be issued if a state transition occurs at the $\overline{CTS}$ pin. If no "Clear to Send" function is required, the $\overline{CTS}$ pins should be connected directly to GND.

$\overline{CTSB}$/CXDB

Collison Data (Channel A/Channel B) (Input)

In a bus configuration, the external serial bus must be connected to the respective CxD pin for collision detection.

TXDA, TXDB

Transmit Data (Channel A/Channel B) (Output)

Transmit Data is shifted out through these pins at standard TTL or CMOS levels. These pins can be programmed to work either as push-pull, or open drain outputs supporting bus configurations.

RES

Reset (Input)

A HIGH signal on this input forces the HSCX into the reset state. The HSCX is in power-up mode during reset and in power-down mode after reset. The minimum pulse which is 1.8 µs.

IM1

Input Mode 1 (Input)

By connecting this pin to either V_{SS} or V_{DD}, the bus interface can be adapted to either an AMD/Siemens/Intel or Motorola environment.
IM1 = LOW: Intel Bus Mode
IM1 = HIGH: Motorola Bus Mode

ALE/IM0

Address Latch Enable (Intel Bus Mode) (Input)

A HIGH on this line indicates an address on the external address/data bus, which will select one of the HSCX's internal registers. The address is latched by the HSCX with the falling edge of ALE. This allows the HSCX to be directly connected to a CPU with a multiplexed address/data bus compatible to Am82520 HSCC. The address input pins A0–A6 must be exernally connected to a CPU with a multiplexed address/data bus compatible to Am82520 HSCC. The address input pins A0–A6 must be externally connected to the data bus pins (D0–D6 for 8-bit CPUs, D1–D7 for 16-bit CPUs; that is, multiply all internal register addresses by 2).

Input Mode 0, Motorola Bus Mode

In Motorola bus mode, the level at this pin determines the function of the IC1 pin (see description of pin 6).

V_{SS}

Ground (Input)

0 V

A0–A6

Address Bus (Input)

These inputs interface with seven bits of the system's address bus A3 to select one of the internal registers for read or write. They are usually connected at A0–A6 in 8-bit systems or at A1–A7 in 16-bit systems.

$\overline{INT}$

Interrupt Request (Output)

This signal is activated when the HSCX requests an interrupt. The CPU may determine the particular source and cause of the interrupt by reading the HSCX interrupt status registers (ISTA, EXIR).

$\overline{INT}$ is an open drain output; therefore, the interrupt request outputs of several HSCXs can be connected to one interrupt input in a "wired-or" combination. This pin must be connected to a pull-up register.

$\overline{DACKA}$/$\overline{DACKB}$

Acknowledge (Channel A/Channel B) (Input)

When LOW, this input signal from the DMA controller notifies the HSCX that the requested DMA cycle controlled via DRQxx (pins 37–40) is in progress; that is, the DMA controller has achieved bus mastership from the CPU and will start data transfer cycles (either Read or Write).

Together with $\overline{RD}$, if DMA has been requested from the receiver or with $\overline{WR}$, if DMA has been requested from the transmitter, this input works like $\overline{CS}$ to enable a data byte to be read from or written to the top of the receive or transmit FIFO of the specified channel.

IF $\overline{DACK}$n is active, the input pin A0–A6 is ignored and the FIFOs are implicitly selected. If the $\overline{DACK}$n signals are not used, these pins must be connected to V_{DD}.

AxCLKA, AxCLKB

Alternate Clock (Channel A/Channel B) (Input)

These pins realize several input functions. Depending on the selected clock mode, they may supply either a:

–CD (carrier detect) modem control or general purpose input. This pin can be programmed to function as receiver enable if the "auto start" feature is selected (CAS bit in XBCH set). The state at this pin can be read from the VSTR register, or

–a receive strobe signal (clock mode 1), or

–a frame synchronization signal in time-slot oriented operation mode (clock mode 5), or

–together with RxCLK, a crystal connection for the internal oscillator (clock mode 4, 6, 7).

TxCLKA, TxCLKB

Transmit Clock (Channel A/Channel B) (Input/Output)

The functions of these pins depend on the programmed clock clock, provided that the TSS bit in the CCR2 register is reset. Programmed as inputs (if the TIO bit in CCR2 is reset), they may supply either:

–the transmit clock for the respective channel (clock mode 0, 2, 6), or

–a transmit strobe signal (clock mode 1).

Programmed as outputs (if the TIO bit in CCR2 is set), the TxCLK pins supply either the:

–transmit clock of the respective channel which is generated either

- from the baud rate generator (clock mode 2, 6; TSS bit in CCR2 set), or
- from the DPLL circuit (clock mode 3, 7), or
- from the crystal oscillator (clock mode 4), or

–a tri-state control signal indicating the programmed transmit time slot (clock mode 5).

RxCLKA, RxCLKB

Receive Clock (Channel A/Channel B) (Input)

The function of these pins also depend on the programmed clock mode. In each channel, RxCLK may supply either:

–the receive clock (clock mode 0), or

–the receive and transmit clock (clock mode 1, 5), or

–the clock for the baud rate generator (clock mode 2, 3), or

–crystal connection for the internal oscillator (clock mode 4, 6, 7, together with AxCLKA).

DRQRA, DRQRB

DMA Request Receiver (Channel A/Channel B) (Output)

The receiver of the HSCX requests a DMA data transfer by activating this line. DRQRn remains HIGH as long as the receive FIFO requires data transfers, thus blocks of data (32, 16, 8, or 4 bytes) are transferred.

DRQRn is activated immediately following the falling edge of the last read cycle.

DRQTA, DRQTB

DMA Request Transmitter (Channel A/Channel B) (Output)

The transmitter of the HSCX requests a DMA data transfer by activating this line.

The DRQTn remains HIGH as long as the transmit FIFO requires data transfers.

The amount of data bytes to be transferred from system memory to the HSCX (byte count) must be written first to the XBCH, XBCL registers.

Blocks of data (n · 32 bytes + REST, n = 0, 1, . . .) are transferred until the byte count is reached.

DRQTn is deactivated immediately following the falling edge of the last $\overline{WR}$ cycle.

V_{DD}

Power (Input)

+5 V power supply.

FUNCTIONAL DESCRIPTION

The HSCX comprises two completely independent full-duplex HDLC channels, Channel A and Channel B, supporting various layer 1 functions by means of an internal oscillator, baud rate generator (BRG), digital phase locked loop (DPLL), and time slot assignment (TSA) circuits.

Furthermore, layer 2 functions are performed by an on-chip LAP (link access procedure) data link controller.

The data link controller handles all functions necessary to establish and maintain an HDLC data link, such as:

- Flag insertion and detection
- Bit stuffing (zero-bit insertion/deletion)
- CRC generation and checking
- Address field recognition

Associated with each serial channel is a set of independent command and status registers (SP-REG) and 64-byte-deep FIFOs for transmit and receive direction.

DMA capability has been added to the HSCX by means of a 4-channel DMA interface with one DMA request line for each transmitter and receiver of both channels.

General

The HSCX distinguishes from other low-level HDLC devices by its advanced characteristics. The most important are:

- Support of different link configurations.

  Beyond the point-to-point configurations, the HSCX directly enables point-to-multipoint or multimaster configurations without additional hardware or software expense.

  In point-to-multipoint configurations, the HSCX can be used as a master as well as a slave station. Even when working as a slave station, the HSCX can initiate the transmission of data at any time. An internal function block provides the means of idle and collision detection, and collision resolution, which are necessary if several stations start transmitting simultaneously. A multimaster configuration is also possible.

- Support of layer 2 functions by HSCX.

  Besides those bit-oriented functions usually supported with the HDLC protocol such as bit stuffing, CRC check, flag and address recognition, the HSCX provides a high degree of procedural support.

  In a special operating mode (auto mode), the HSCX processes the information transfer and the procedure handshaking (I-, and S-frames of HDLC protocol) autonomously. The only restriction is that the window size (number of outstanding unacknowledged frames) is limited to 1. This will be sufficient in most applications. The communication procedures are mainly processed between the communication controllers and not between the processors; therefore, the dynamic load of the CPU and the software expense is largely reduced.

  The CPU is informed about the status of the procedure and has mainly to manage the receive and transmit data. In order to maintain cost effectiveness and flexibility, such functions as link setup/disconnection and error recovery in case of protocol errors (U-frames of HDLC protocols) are not implemented in hardware and must be done by user's software.

- Telecom-specific features.

  In a special operating mode, the HSCX can transmit or receive data packets in one of up to 64 time slots of programmable width (clock mode 5). Furthermore, the HSCX can transmit or receive variable data portions within a defined window of one or more clock cycles, which has to be selected by an external strobe signal (clock mode 1). These features make the HSCX especially suitable for all applications using time division multiplex methods such as time slot oriented PCM systems, systems designed for packet switching, or in ISDN applications.

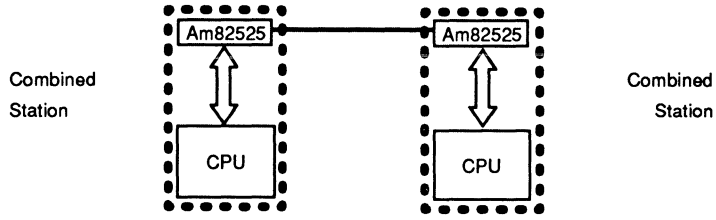
- FIFO buffers for efficient transfer of data packets.

  The FIFO buffers used for the temporary storage of data packets transferred between the serial communications interface and the parallel system bus are another feature of the HSCX. Also because of the overlapping input/output operation (dual-port behavior), the maximum message length is not limited by the size of the buffer. Together with the DMA capability, the dynamic load of the CPU is drastically reduced by transferring the data packets block by block via direct memory access. The CPU only has to initiate the data transmission by the HSCX and determine the status in case of completely received frames but is not involved in data transfers.

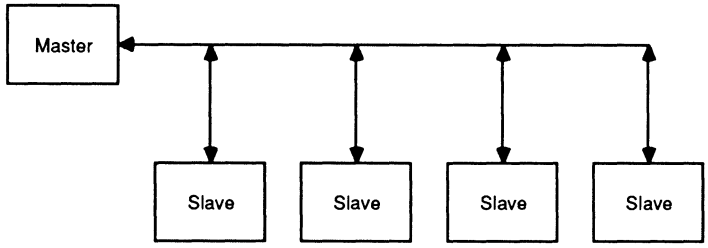
Operating Modes

The HDLC controller of each channel can be programmed to operate in various modes, which are different in the treatment of the HDLC frame in receive direction. Thus, the receive data flow and the address recognition features can be effected in a very flexible way, which satisfies almost every requirement. There are six different operating modes which can be set by the mode register.

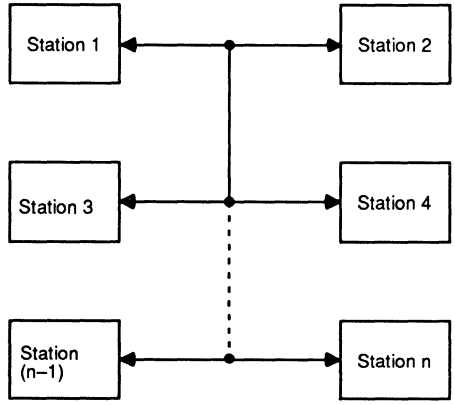
Auto Mode (MODE: MDS1, MDS0 = 00) — Characteristics: window size 1, random message length, address recognition.



Point-to-Point Configuration



Point-to-Multipoint Configuration



Multimaster Configuration

11139–004B

Figure 4. Link Configurations

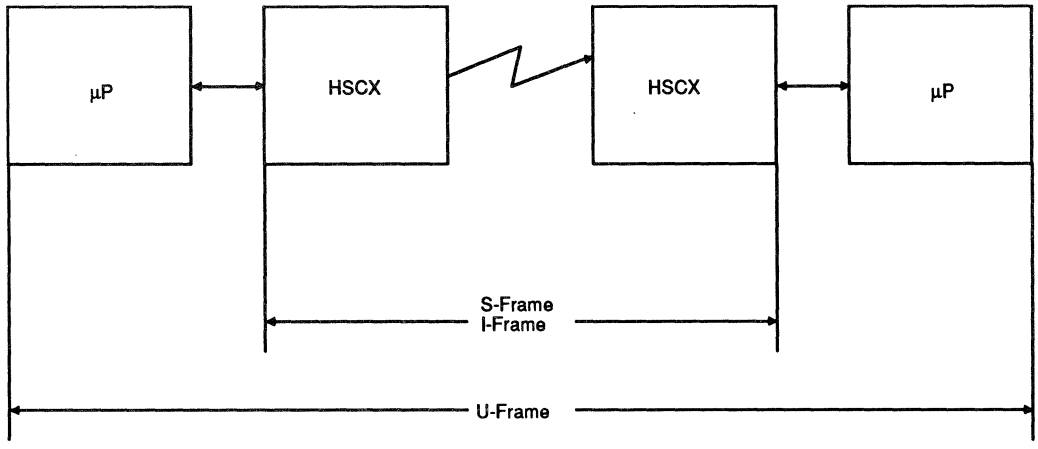


Figure 5. Procedural Support in Auto Mode

The HSCX processes autonomously all numbered frames (S-, I-frames) of an HDLC procedure. The HDLC control field, data in the I-field of the frames, and an additional status byte is temporarily stored in the RFIFO. The HDLC control field as well as additional information can also be read from special registers (RHCR, RSTA).

According to the selected address mode, the HSCX can perform a 2-byte or 1-byte address recognition. If a 2-byte address field is selected, the high address byte is compared with the fixed value FEH or FCH (group address) as well as with two individually programmable values in RAH1 and RAH2 registers. According to the ISDN LAPD protocol, bit 1 of the high byte address will be interpreted as command/response bit (C/R), depending on the setting of the CRI bit in RAH1, and will be excluded from the address comparison.

Similarly, two compare values can be programmed in special registers (RAL1, RAL2) for the low address byte. A valid address will be recognized in case the high and low byte of the address field correspond to one of the compare values. The HSCX can be called (addressed) with six different address combinations; however, only the logical connection identified through the address combination RAH1, RAL1 will be processed in the auto mode. All others will be processed in the non-auto mode. HDLC frames with address fields that do not match with any of the address combinations are ignored by the HSCX.

In case of a 1-byte address, RAL1 and RAL2 will be used as compare registers. According to the X.25 LAPB protocol, the value in RAL1 will be interpreted as command and the value in RAL2 as response.

Non-Auto Mode (MODE: MDS1, MDS0 = 01)—Characteristics: address recognition, random window size.

All frames with valid addresses (address recognition identical to auto mode) are forwarded directly to the system memory. The HDLC control field, data in the I-field and an additional status byte are temporarily stored in the RFIFO. The HDLC control field and additional information can also be read from special registers (RHCR, RSTA). In non-auto mode, all frames are treated similarly.

Transparent Mode 1 (MODE: MDS1, MDS0, ADM = 101)—Characteristics: address recognition high byte.

Only the high byte of a 2-byte address field will be compared. The whole frame except the first address byte will be stored in RFIFO. RAL1 contains the second and RHCR the third byte following the opening flag.

Transparent Mode 0 (Mode: MDS1, MDS0, ADM = 100)—Characteristics: no address recognition.

No address recognition is performed and each frame will be stored in the RFIFO. RAL 1 contains the first and RHCR the second byte following the opening flag.

Extended Transparent Modes 0, 1 (Mode: MDS1, MDS0 = 11)—Characteristics: fully transparent.

In extended transparent modes, fully transparent data transmission/reception without HDLC framing is performed; that is, without flag generation/recognition, CRC generation/check, bit-stuffing mechanism. This allows user-specific protocol variations or the usage of character oriented protocols (such as IBM BISYNC).

Data transmission is always performed out of the XFIFO. In extended transparent mode 0 (ADM = 0), data reception is done by the RAL1 register, which always contains the actual data byte assembled at the RxD pin. In extended transparent mode 1 (ADM = 1), the receive data is additionally shifted into the RFIFO.

SYSTEM INTEGRATION

General Aspects

Figure 6 gives a general overview of the system integration of HSCX.

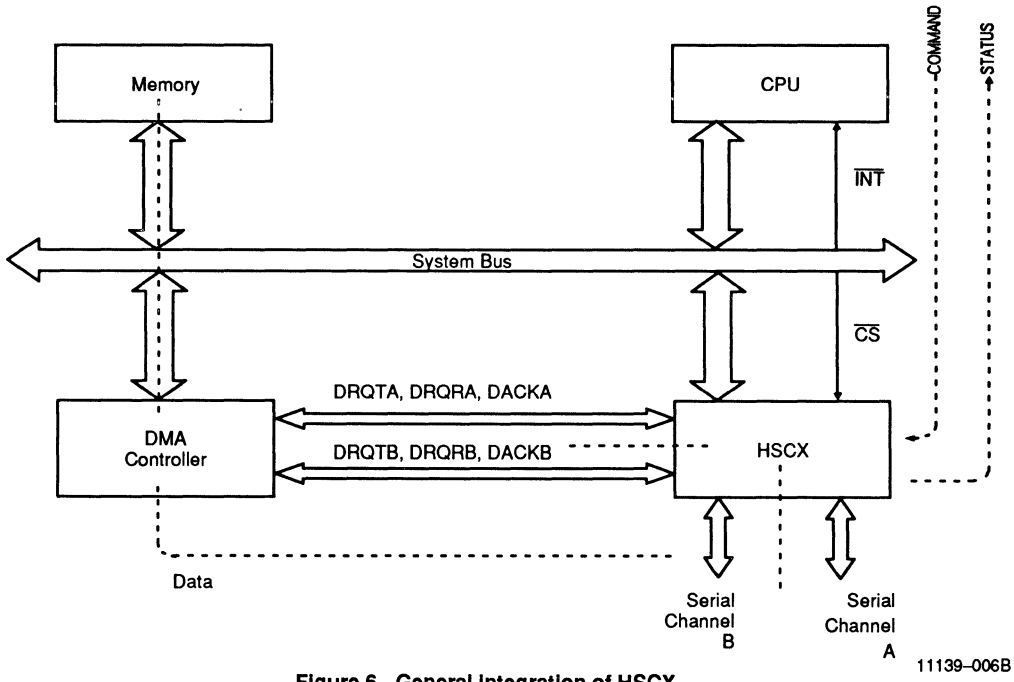


Figure 6. General Integration of HSCX

The HSCX's bus interface consists of an 8-bit bidirectional data bus (D0–D7), seven address line inputs (A0–A6), three control inputs ($\overline{RD}$ / DS, $\overline{WR}$ / R/W,$\overline{CS}$), one interrupt request output ($\overline{INT}$) and a 4-channel DMA interface (DRQTA, DRQRA, $\overline{DACKA}$, DRQTB, DRQRB, DACKB). Mode input pins (strapping options) allow the bus interface to be configured for either an AMD/ Siemens/Intel, or Motorola environment.

Generally, there are two types of transfers occurring on the system bus:

- Command/Status transfers, which are always controlled by the CPU. The CPU sets the operation mode (initialization), controls function sequences and gets status information by writing or reading the HSCX's registers (using $\overline{CS}$, $\overline{WR}$, or $\overline{RD}$, and register address using A0–A6).
- Data Transfers, which are effectively performed by DMA without CPU interaction using the HSCX DMA interface (DMA Mode). Optionally, interrupt controlled data transfer can be done by the CPU (Interrupt Mode).

Specific Applications

HSCX with Am8051 Microcontroller

For cost-sensitive applications, the HSCX can be interfaced with an Am8051 microcontroller system (without DMA support) very easily as shown in Figure 7.

Although the HSCX provides a demultiplexed bus interface, it optionally can be connected directly to the local multiplexed bus of Am8051 because of the internal address latch function (using ALE, compatible to Am82520 HSCC).

The address lines A0–A6 must be wired externally to the data lines D0–D6 (direct connection) in this case.

Intel Bus Mode is selected connecting IM1 pin to LOW (GROUND). Since data transfer is controlled by interrupt, the DMA acknowledge inputs ($\overline{DACKA}$, $\overline{DACKB}$) are connected to V_{DD} (+5 V).

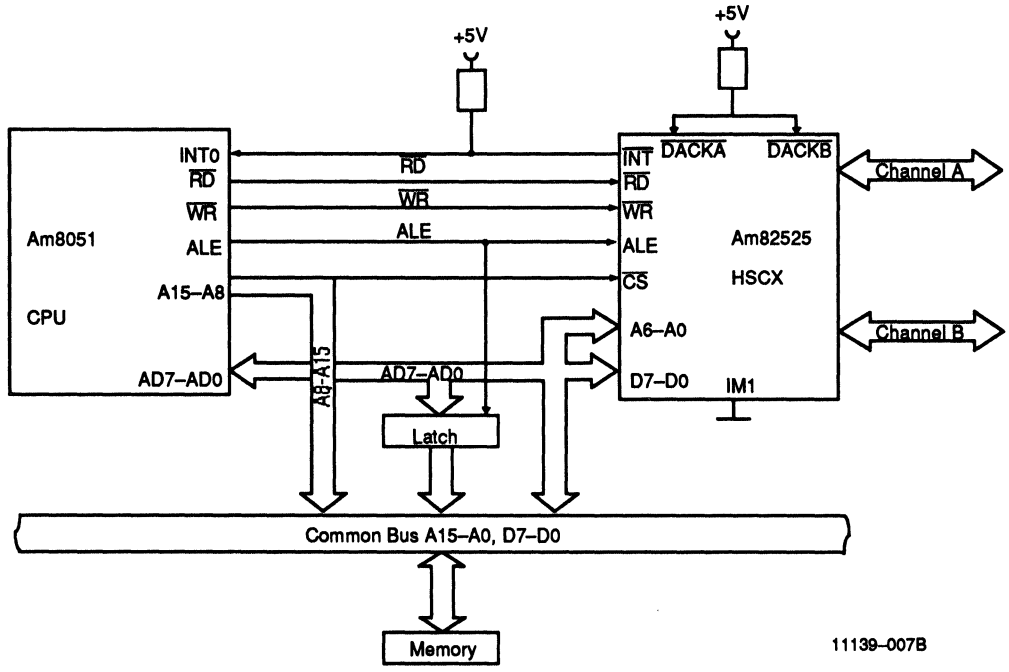


Figure 7. HSCX with Am8051 Microcontroller

HSCX with an Am80188 Microprocessor

A system with minimized additional hardware expense can be built up with an Am80188 microprocessor as shown in Figure 8.

The HSCX is connected to the demultiplexed system bus. Data transfer for one serial channel can be done by the 2-channel on-chip DMA controller of the Am80188; the other channel is serviced by interrupt. Since the Am80188 does not provide DMA acknowledge outputs, data transfer from/to HSCX is controlled with $\overline{CS}$, $\overline{RD}$, or $\overline{WR}$ address information (A0–A6), and the $\overline{DACKA}$, $\overline{DACKB}$ inputs are not used.

- This solution supports applications with a high speed data rate in one serial channel with minimum hardware expense making use of the on-chip peripheral functions of the Am80188 (chip select logic, interrupt controller, DMA controller).

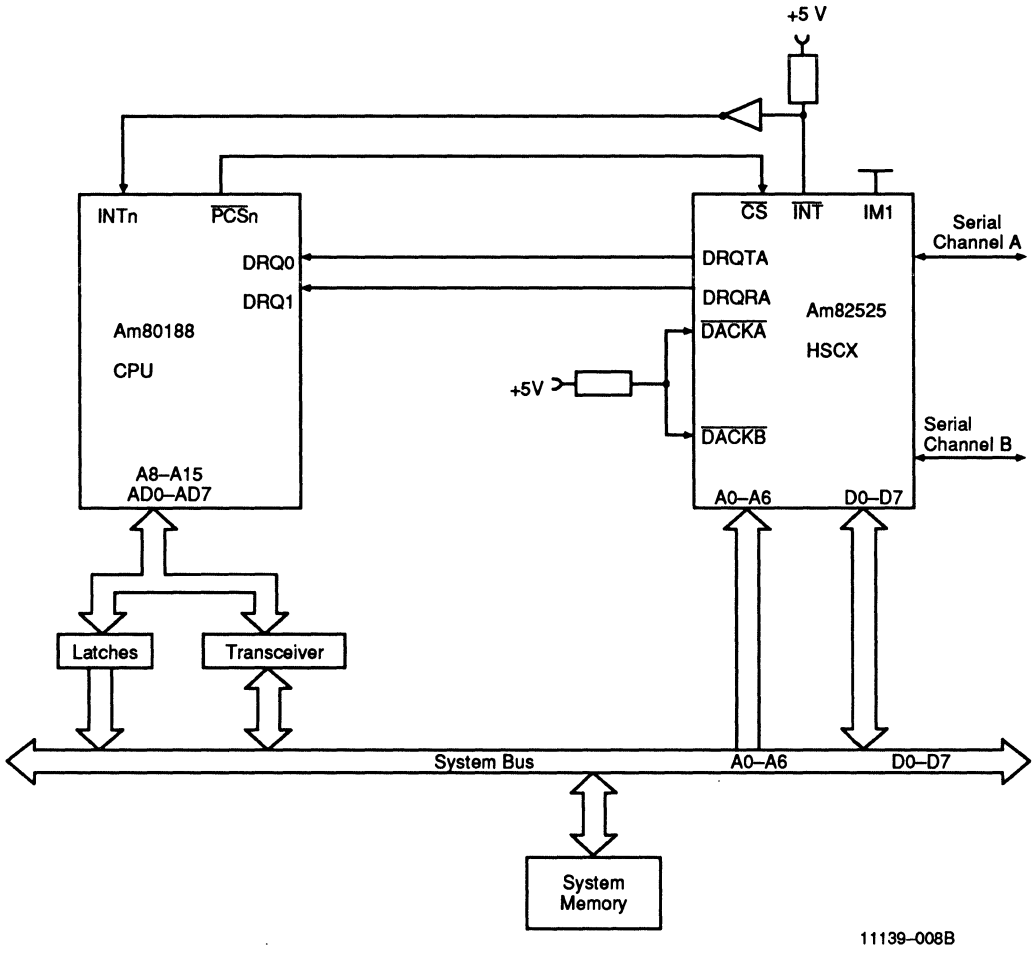


Figure 8. HSCX with 80188 CPU

OPERATIONAL DESCRIPTION

Reset

The HSCX is forced into the reset state if a high signal is input to the RES pin for a minimum period of 1.8 µs. During RESET, the HSCX is temporarily in the power-up mode, and a subset of the registers is initialized with defined values.

After RESET, the HSCX is in power-down mode, and the following registers contain defined values:

Register	Reset Value	Meaning
CCR1	00 H	– power down mode Serial port configuration: pt-pt, NRZ coding, transmit data pins are open drain outputs
CCR2	00 H	– clock mode 0 $\overline{RTS}$ pin normal function
MODE	00 H	– $\overline{CTS}$ and $\overline{RTS}$ interrupts disabled no data inversion auto-mode 1 byte address field external timer mode
	48 H	– receivers inactive $\overline{RTS}$ output controlled by HSCX, timer resolution: k = 32.768, no testloop
	00 H	XFIFO write enable receive line inactive no commands executing
	00 H	– no interrupts masked
	00 H	no commands
	00 H	– interrupt controlled data transfer (DMA disabled) – full-duplex LAPB/LAPD operation of LAP controller – carrier detect auto start of receiver disabled 1-bit timeslots

Operational Phase

After having performed the initialization, the CPU switches each individual channel of the HSCX into operational phase by setting the PU bit in the CCR1 register (power-up, if not already done during initialization).

Initially, the CPU should bring the transmitter and receiver to a defined state by issuing an XRES (transmitter reset) and RHR (receiver reset) command from the CMDR register. If data reception should be performed, the receivers must be activated by setting the RAC bit in Mode to 1.

If no "Clear to send" function is provided with a modem, the $\overline{CTS}$ pins of the HSCX must be connected directly to ground in order to enable data transmission.

Now the HSCX is ready to transmit and receive data. The control of the data transfer phase is done mainly by commands from CPU to HSCX via the CMDR register and by interrupt indications from HSCX to CPU.

Additional status information, which does not trigger an interrupt, is available in the STAR register. A complete description of every register is provided in the following paragraph. Additionally, the address of the respective registers is noted in the form: (ADDR Channel A/ADDR Channel B).

ADDRESS (A0–A6) Channel		REGISTER		Comment Meaning
A	B	Read	Write	
00	40			
•	•	RIFO	XFIFO	Receive/Transmit FIFO
•	•			
1F	5F			
20	60	ISTA	MASK	Interrupt STAtus/MASK
21	61	STAR	CMDR	STAtus/CoMmanD
22	62	MODE		MODE
23	63	TIMR		TIMeR
24	64	EXIR	XAD1	EXtended Interrupt/Transmit ADdress 1
25	65	RBCL	XAD2	Receive Byte Count Low/Transmit ADdress 2
26	66	—	RAH1	Receive Address HIGH 1
27	67	RSTA	RAH2	Receive STAtus/Receive Address HIGH 2
28	68	RAL1		Receive Address LOW 1
29	69	RHCR	RAL2	Receive HDLC Control/Receive Address LOW 2
2A	6A	—	XBCL	Transmit Byte Count LOW
2B	6B	—	BGR	Baud Rate Generator Register
2C	6C	CCR2		Channel Configuration Register 2
2D	6D	RBCH	XBCH	Receive/Transmit Byte Count HIGH
2E	6E	VSTR	RLCR	Version STAtus/Receive Frame Length Check
2F	6F	CCR1		Channel Configuration Register 1
30	70	—	TSAX	Time Slot Assignment Transmit
31	71	—	TSAR	Time Slot Assignment Receive
32	72	—	XCCR	Transmit Channel Capacity
33	73	—	RCCR	Receive Channel Capacity

11139–009B

Figure 9. Layout of Register Addresses

REGISTER DEFINITIONS

Receive FIFO (Read) RFIFO (00 . . . 1F/40 . . . 5F)

- Interrupt Controlled Data Transfer (Interrupt Mode). Selected if DMA bit in XBCH is reset. Up to 32 bytes of receive data can be read from the RFIFO following an RPF or an RME interrupt.

  RPF Interrupt: Exactly 32 bytes to be read.

  RME Interrupt: Number of bytes to be determined by reading the RBCL, RBCH registers.

- DMA Controlled Data Transfer (DMA Mode). Selected if DMA bit in XBCH is set.

  If the RFIFO contains 32 bytes, the HSCX autonomously requests a block data transfer by DMA activating the DRQR line as long as the start of the 32nd read cycle. This forces the DMA controller to continuously perform bus cycles until 32 bytes are transferred from the HSCX to the system memory (level triggered, demand transfer mode of DMA controller).

  If the RFIFO contains less than 32 bytes (one short frame or the last part of a long frame) the HSCX requests a block data transfer depending on the contents of the RFIFO according to the following table:

RFIFO Contents (Bytes)	DMA Request (Actual Bytes Transferred)
(1) 2,3	4
4–7	8
8–15	6
16–32	32

  Additionally, an RME interrupt is issued after the last byte has been transferred.

  As a result, the DMA controller may transfer more bytes as actually valid in the current received frame. The valid byte count must therefore be determined reading the RBCH, RBCL registers following the RME interrupt.

Transmit FIFO (WRITE) XFIFO (00 . . . 1F/40 . . . 5F)

- Interrupt Mode. Selected if DMA bit in XBCH is reset.

  Up to 32 bytes of transmit data can be written to the XFIFO following an XPR interrupt.

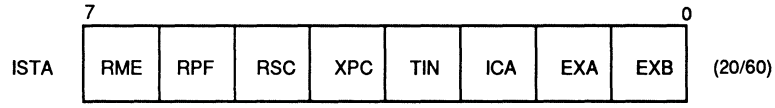
- DMA Mode. Selected if DMA bit in XBCH is set.

  Prior to any data transfer, the actual byte count of the frame to be transmitted must be written to the XBCH, XBCL registers by the user.

  If data transfer is then initiated via the CMDR register (command XTF or XIF), the HSCX autonomously requests the correct amount of block data transfers ($n \cdot 32 + REST$, $n = 0,1,\ldots$).

Note: Addresses within the address space of the FIFOs are interpreted equally; that is, the actual data byte can be accessed with any address within the valid scope.

Interrupt Status Register (READ)–Value after RESET: 00H



Receive Message End (RME)—one message up to 32 bytes or the last part of a message greater than 32 bytes has been received and is now available in the RFIFO. The message is complete.

The actual message length can be determined reading the RBCH, RBCL registers. Additional information is available in the RSTA register.

Receive Pool Full (RPF)—a block of 32 bytes of a message is stored in the RFIFO. The message is not yet complete.

Note: This interrupt is generated only in Interrupt mode.

Receive Status Change (RSC) significant in auto mode only—a status change (receiver ready/receiver not ready) of the opposite station has been detected in auto mode; that is, the HSCX has received an RR/RNR supervisory frame according to the HDLC protocol. The current status can be read from the STAR register (RRNR bit).

Transmit Pool Ready (XPR)—a data block of up to 32 bytes can be written to the transmit FIFO.

Timer Interrupt (TIN)—the internal timer and repeat counter has expired. (See also description of TIMR register.)

Interrupt of Channel A (ICA) Channel B only—indicates that an interrupt is caused by Channel A and the interrupt source is indicated in the ISTA register of Channel A; that is, at least one bit of the ISTA register of Channel A is set.

Extended Interrupt of Channel A (EXA) Channel B only—an interrupt is caused by Channel A and the source is indicated in the EXIR register of Channel A.

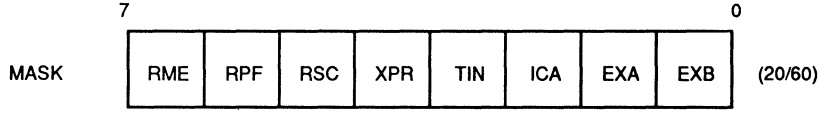
Extended Interrupt of Channel B (EXB) Channel B only—an interrupt is caused by Channel B and the source is indicated in the EXIR register of Channel B.

Note: The ICA, EXA, and EXB bits are present in Channel B only and point to the ISTA (Channel A), EXIR (Channel A), and EXIR (Channel B) registers.

After the HSCX has requested an interrupt by asserting its $\overline{\text{INT}}$ pin LOW, the CPU must first read the ISTA register of Channel B and check the state of these bits in order to determine which interrupt source of which channel has caused the interrupt. More than one interrupt source may be indicated by a single interrupt request.

After the respective register has been read, EXA, and EXB are reset. All other bits will be reset after reading ISTA. To prevent malfunctions, each bit is individually monitored and reset.

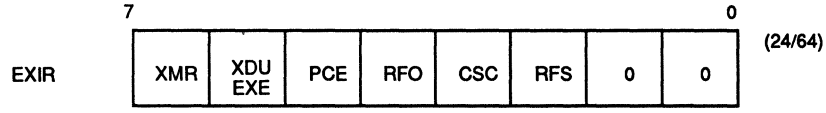
Mask Register (WRITE)—Value after RESET: 00H (all interrupts enabled)



Each interrupt source can be selectively masked by setting the respective bit in the Mask (bit positions corresponding to ISTA register). Masked interrupts are not indicated when reading ISTA. Instead, they remain internally stored and will be indicated after the respective Mask bit is reset.

Note: In the event of an extended interrupt, no interrupt request will be generated with a masked EXA, EXB bit, although this bit is set in ISTA.

Extended Interrupt Register (READ)—Value after RESET: 00H



Transmit Message Repeat (XMR)—the transmission of the last message has to be repeated because:

- the HSCX has received a negative acknowledg ment in auto mode, or
- a collision has occurred after sending the 32nd data byte of a message in a bus configuration, or
- $\overline{CTS}$ (transmission enable) was withdrawn after sending the 32nd data byte of a message in a point-to-point configuration.

Transmit Data Underrun/Extended Transmission End (XDU/EXE) — the actual frame has been aborted with Idle, because the XFIFO holds no further data, but the frame is not yet complete. In extended transparent mode, this bit indicates the transmission-end condition.

Note: It is not possible to send transparent- or I-frames when an XMR or XDU interrupt is indicated.

Protocol Error (PCE) significant in auto mode only—the HSCX has detected a protocol error; that is, it has received:

- an S-, or I-frame with incorrect N (R)
- an S-frame containing an I-field

Receive Frame Overflow (RFO)—one frame could not be stored due to occupied RFIFO; that is, a whole frame has been lost. This interrupt can be used for statistical purposes and indicates that the CPU does not respond quickly enough to an incoming RPF or RME interrupt.

Clear To Send Status Change (CSC)—indicates that a state transition has occurred at the $\overline{CTS}$ pin. The actual state can be read from STAR register ($\overline{CTS}$ bit).

This interrupt must be enabled setting the CIE bit in CCR2.

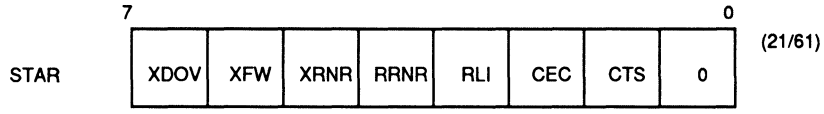
Receive Frame Start (RFS)—this is an early receiver interrupt activated after the start of a valid frame has been detected; that is, after a valid address check in operation mode providing address recognition, otherwise after the opening flag (transparent mode 0), delayed by two bytes.

After an RFS interrupt, the contents of the following are valid and can be read by the CPU.

- RHCR
- RAL1
- RSTA—bit 3–0

This interrupt must be enabled setting the RIE bit in CCR2.

Status Register (READ)—Value after RESET: 48H



(21/61)

Transmit Data Overflow (XDOV)—more than 32 bytes have been written to the XFIFO.

Transmit FIFO Write Enable (XFW)—data can be written to the XFIFO.

Transmit RNR (XRNR) significant in auto mode only —indicates the status of the HSCX.

0—receiver ready

1—receiver not ready

Received RNR (RRNR) significant in auto mode only — indicates the status of the remote station.

0—receiver ready

1—receiver not ready

Receive Line Inactive (RLI) — Neither flags as inter-frame time fill nor frames are received via the receive line.

Note: Significant only in point-to-point configurations.

Command Executing (CEC)

0—no command is currently executed; the CMDR register can be written to.

1—a command (written previously to CMDR) is currently executed, no further command can be temporarily written via CMDR register.

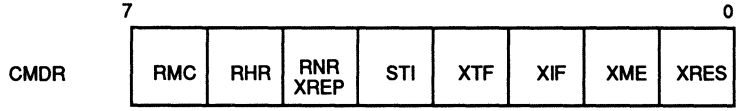
Clear To Send State ($\overline{CTS}$)—if the CIE bit in CCR2 is set, this bit indicates the state of the $\overline{CTS}$ pin.

0—$\overline{CTS}$ is inactive (HIGH signal at $\overline{CTS}$)

1—$\overline{CTS}$ is active (LOW signal at $\overline{CTS}$)

2

Command Register (WRITE)—Value after RESET: 00H.



(21/61)

Receive Message Complete (RMC)—confirmation from CPU to HSCX that the actual frame or data block has been fetched following an RPF or RME interrupt, thus the occupied space in the RFIFO can be released.

Note: In DMA mode, this command is issued only once after an RME interrupt. The HSCX does not generate further DMA requests prior to the reception of this command.

Reset HDLC Receiver (RHR) all data in the RFIFO and the HDLC receiver is deleted.

In auto mode, the transmit and receive sequence number counters additionally are reset.

Receiver Not Ready/Transmission Repeat (RNR/XREP)—the function of this command depends on the selected operation mode (MDS1, MDS0, ADM bit in Mode):

- Auto mode: RNR

The status of the HSCX receiver is set. Determines whether a received frame is acknowledged via an RR or RNR supervisory frame in auto mode.

0—receiver ready (RR)

1—receiver not ready (RNR)

- Extended Transparent mode 0,1: XREP

Together with XTF and XME set (write 2AH to CMDR), the HSCX repeatedly transmits the contents of the XFIFO (1 . . . 32 bytes) without HDLC framing fully transparent, that is, without Flag, CRC insertion or Bit Stuffing.

The cyclic transmission is stopped with an XRES command.

Start Timer (STI)—the internal timer is started.

Note: The timer is stopped by rewriting the TIMR register after start.

Transmit Transparent Frame (XTF)

- Interrupt mode

After having written up to 32 bytes to the XFIFO, this command initiates the transmission of a transparent frame. An opening flag sequence is automatically added to the data by the HSCX.

- DMA mode

After having written the length of the frame to be transmitted to the XBCH, XBCL registers, this command initiates the data transfer from system memory to HSCX by DMA. Serial data transmission starts as soon as 32 bytes are stored in the XFIFO.

Transmit I-Frame (XIF) used in auto mode only—initiates the transmission of an I-frame in auto mode. In addition to the opening flag sequence, the address and control field of the frame are automatically added by HSCX.

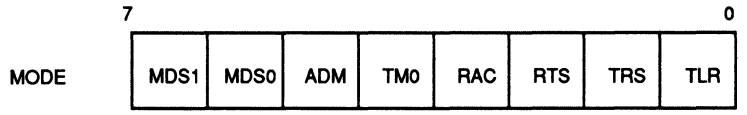
Transmit Message End (XME) used in interrupt mode only—indicates that the data block written last to the transmit FIFO completes the actual frame. The HSCX can terminate the transmission operation properly by appending the CRC and the closing flag sequence to the data.

In DMA mode, the end of the frame is determined by the transmit byte count in XBCH, XBCL.

Transmit Reset (XRES)—the contents of the XFIFO are deleted and Idle is transmitted. This command can be used by the CPU to abort a frame currently in transmission.

Note: The maximum time between writing to the CMDR register and the execution of the command is 2,5 clock cycles. Therefore, if the CPU operates with a very high clock in comparison with the HSCX's clock, it's recommended that the CEC bit of the STAR register be checked before writing to the CMDR register to avoid any loss of commands.

Mode Register (READ/WRITE)—Value after RESET: 00H



(22/62)

Mode Select (MDS1, MDS0)—the operating mode of the HDLC controller is selected.

00 — auto mode

01 — non-auto mode

10 — transparent mode

11 — extended transparent mode

Address Mode (ADM)—the meaning of this bit varies depending on the selected operating mode:

- Auto mode, non-auto mode

  Defines the length of the HDLC address field.

  0 — 8-bit address field

  1 — 16-bit address field

In transparent modes, this bit differentiates between two sub-modes:

- Transparent mode

  0 — transparent mode 0; no address recognition.

  1 — transparent mode 1; high byte address recognition.

- Extended transparent mode; without HDLC framing.

  0—extended transparent mode 0

  1—extended transparent mode 1

Note: In extended transparent modes, the RAC bit must be reset to enable fully transparent reception.

Timer Mode (TMD)—the operation mode of the internal timer is set.

0—external mode

The timer is controlled by the CPU and can be started at any time setting the STI bit in CMDR.

1—internal mode

The timer is used internally by the HSCX for timeout and retry conditions in auto mode (refer to the description of the TIMR register).

Receiver Active (RAC)—switches the receiver to operational or inoperational state.

0—receiver inactive

1—receiver active

In extended transparent modes this bit must be reset to enable fully transparent reception.

Request To Send (RTS)—defines the state and control of the $\overline{RTS}$ pin.

0—The $\overline{RTS}$ pin is controlled by the HSCX autonomously.

$\overline{RTS}$ is activated when a frame transmission starts and is deactivated after the transmission operation is complete.

1—The $\overline{RTS}$ pin is controlled by the CPU.

If this bit is set, the $\overline{RTS}$ pin is activated immediately and remains active until this bit is reset.

Timer Resolution (TRS)—the resolution of the internal timer (factor k, see description of TIMR register) is selected.

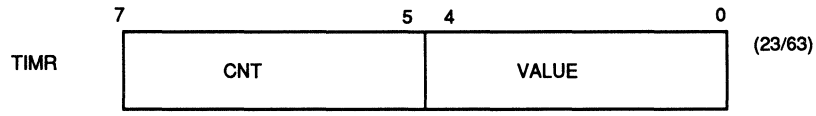
0 — k = 32.768

1 — k = 512

Testloop (TLP)—input and output of the HDLC channels are internally connected. (transmitter channel A — receiver channel A/transmitter channel B — receiver channel B).

2

Timer Register (READ/WRITE)



VALUE—Sets the time period t1 as follows:

$$t1 = k \cdot (VALUE + 1) \cdot TCP$$

where k is the timer resolution factor which is either 32.768 or 512 clock cycles depending on the programming of TRS bit in Mode.

–TCP is the clock period of transmit data.

Interpreted differently depending on the selected timer mode (CNT) Bit TMD in MODE.

- Internal timer mode (MODE.TMD = 1)

  –Retry Counter (in HDLC known as N2)

CNT indicates the number of S-commands (maximum six) which are transmitted autonomously by the HSCC after expiration of time period t1, in case an I-frame is not acknowledged by the opposite station.

If CTN is set to 7, the number of S-commands is unlimited.

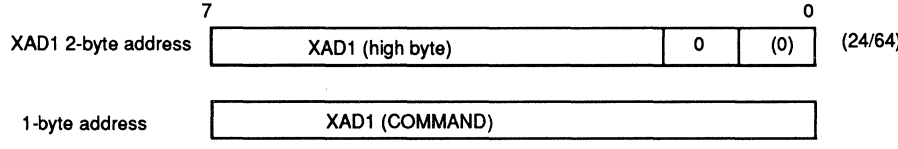
- External timer mode (MODE, TMD = 0)

CNT plus Value indicates the time period t2 after which a timer interrupt will be generated. The time period t2 is

$$t2 = 32 \cdot k \cdot CTN \cdot TCP + t1$$

If CTN is set to 7, a timer interrupt is periodically generated after the expiration of t1.

Transmit Address Byte 1 (WRITE)



XAD1 (and XAD2) can be programmed with one individual address byte which is appended automatically to the frame by HSCX in auto mode. The function depends on the selected address mode (bit ADM in Mode).

2-byte address field (MODE.ADM = 1)

XAD1 builds up the high byte of the 2-byte address field. Bit 1 must be set to 0. According to the ISDN LAPD protocol, bit 1 is interpreted as the C/R (Command/Response) bit. This bit is manipulated automatically by the HSCX depending on the setting of the CRI bit in RAH1:

4R Meaning	bit 1 (C/R)	
Commands transmit	1	0
Responses transmit	0	1
	CRI = 1	CRI = 1

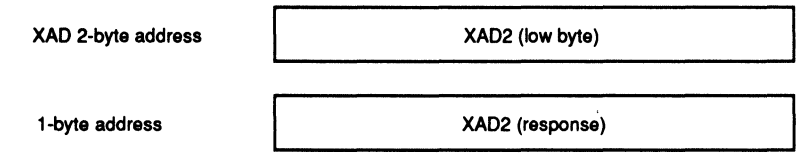
(In the ISDN, the high address byte is known as SAPI.)

In accordance with the HDLC protocol, bit 0 should be set to 0, indicating the extension of the address field to two bytes.

1-byte address field (MODE.ADM = 0)

In accordance with the X.25 LAPB protocol, XAD1 indicates a command.

Transmit Address Byte 2 (WRITE)



Second individually programmable address byte.

2-byte address (MODE.ADM = 1)

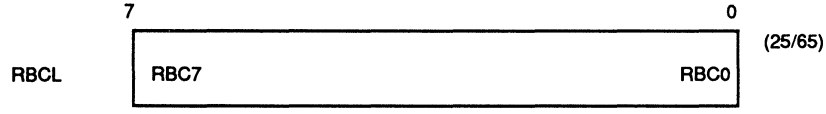
XAD2 builds up the low byte of the 2-byte address field. (In the ISDN, the low address byte is known as TEI.)

1-byte address (MODE.ADM = 0)

In accordance with the X.25 LAPB protocol, XAD2 indicates a response.

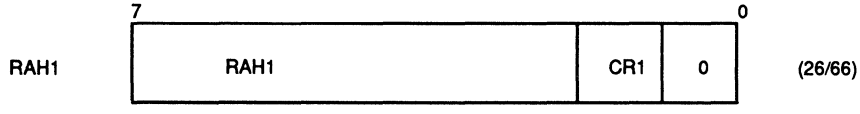
Note: XAD1, XAD2 registers are used only if the HSCX is operated in auto mode.

Receive Byte Count LOW (READ)



Together with RBCH (bits RBC11–RBC8), the length of the actual received frame (1...4095 bytes) can be determined. These registers must be read by the CPU following an RME interrupt.

Receive Address Byte High Register 1 (WRITE)



In operating modes that provide high byte address recognition, the high byte of the received address is compared with the individual programmable values in RAH1 or RAH2.

RAH1—Value of the first individual high address byte

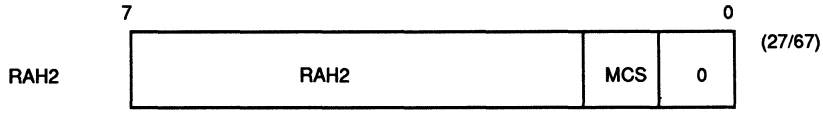
CRI—Command/Response Interpretation

The setting of the CRI bit affects the meaning of the C/R bit in RSTA as follows:

C/R Meaning	(C/R) Value	
Commands Received	0	1
Responses Received	1	0
	CRI = 1	CRI = 1

Important: If the 1-byte address field is selected in auto mode, RAH1 must be set to 00H.

Receive Address Byte High Register 2 (WRITE)



RAH2 — Value of second individual programmable high address byte.

MCS — Modulo Count Select, valid in auto mode only.

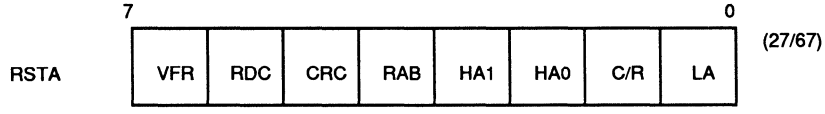
The MCS bit adjusts the control field format according to the HDLC (ISDN/LAPD).

0 – basic operation (modulo 8)

1 – extended operation (modulo 128)

Note: When modulo 128 is selected, in auto mode the "RHCR" register contains compressed information of the extended control field (see RHCR, register description). RAH1, RAH2 registers are used in auto and non-auto operating modes when a 2-byte address field has been selected (MODE.ADM = 1) and in the transparent mode 0.

Receive Status Register (READ)



VFR—Valid Frame—determines whether a valid frame has been received.

1. Valid

0. Invalid

An invalid frame is either

– a frame which is not an integer number of 8 bits (n · 8 bits) in length (e.g., 25 bit), or

– a frame which is too short depending on the selected operation mode via MODE (MDS1, MDS0, ADM) as follows:

- auto-/non-auto mode (16-bit address): 4 bytes
- auto-/non-auto mode (8-bit address): 3 bytes
- transparent mode 1: 3 bytes
- transparent mode 0: 2 bytes

Note: Shorter frames are not reported.

Receive Data Overflow (RDO)—a data overflow has occurred within the actual frame.

CRC Compare/Check (CRC)

0—CRC check failed; received frame contains errors.

1—CRC check okay; received frame is error-free.

Receive Message Aborted (RAB)—the received frame was aborted from the transmitting station.

According to the HDLC protocol, this frame must be discarded by the CPU.

High Byte Address Compare (HA1, HA0), significant only if 2-byte address mode has been selected—in operating modes that provide high byte address recognition, the HSCX compares the high byte of a 2-byte address with the contents of two individual programmable registers (RAH1, RAH2) and the fixed values FEH and FCH (group address).

Depending on the result of this comparison, the following bit combinations are possible:

10 — RAH1 has been recognized

00—RAH2 has been recognized

01—group address has been recognized

Note: If RAH1, RAH2 contain the identical values, the combination 00 will be omitted.

Command/Response (C/R), significant only if 2-byte address mode has been selected—value of the C/R bit (bit of high address byte) in the received frame. The interpretation depends on the setting of the CRI bit in the RAH1 register. Refer also to the description of RAH1 register.

Low Byte Address Compare (LA), not significant in transparent and extended transparent operating mode—the low byte address of a 2-byte address field, or the single address byte of a 1-byte address field is compared with two individual programmable registers (RAL1, RAL2).

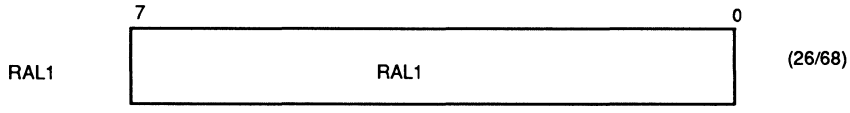
0—RAL2 has been recognized

1—RAL1 has been recognized

According to the X.25 LAPB protocol, RAL1 is interpreted as Command and RAL2 is interpreted as Response.

Note: RSTA corresponds to the last received HDLC frame; it is duplicated into RFIFO for every frame (last byte of frame).

Receive Address Byte Low Register 1 (READ/WRITE)



(26/68)

The general function (READ/WRITE) and the meaning or contents of this register depend on the selected operating mode:

–auto-/non-auto mode (16-bit address)—WRITE:

RAL1 can be programmed with the value of the first individual low address byte.

– auto-/non-auto mode (8-bit address)—WRITE:

According to X.25 LAPB protocol, the address in RAL1 is recognized as command address.

– transparent mode 1 (high byte address recognition) —READ:

RAL1 contains the byte following the high byte of the address in the receive frame (that is, the second byte after the opening flag).

–transparent mode 0 (no address recognition)— READ:

RAL1 contains the first byte after the opening flag (first byte of received frame).

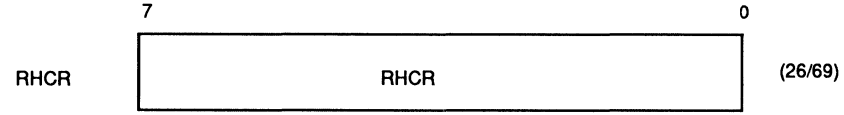
– extended transparent modes 0,1—READ:

RAL1 contains the actual data byte currently assembled at the RxD pin, by passing the HDLC receiver (fully transparent reception without HDLC framing).

Receive Address Byte Low Register 2 (WRITE)

Value of the second individual programmable low address byte. If a one-byte address field is selected, RAL2 is recognized as Response according to X.25 LAPB protocol.

Receive HDLC Control Register (READ)

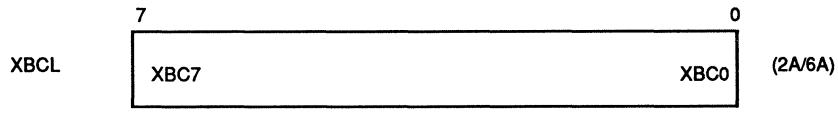


(26/69)

Value of the HDLC control field of the last received frame.

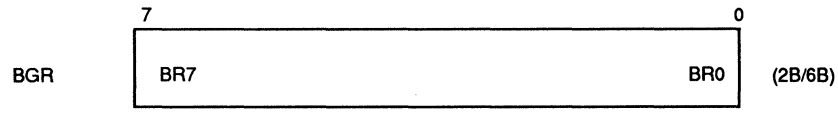
Note: RHCR is duplicated into RFIFO for every frame.

Transmit Byte Count Low (WRITE)



Together with XBCH (bits XBC11–XBC8) this register is used in DMA mode only to program the length (1–4095 bytes) of the next frame to be transmitted. This allows the HSCX to request the correct amount of DMA cycles after an XTF or XIF command via CMDR.

Baud Rate Generator Register (WRITE)



BR7–BR0 — Baud Rate, bit 7–0

Together with bits BR9, BR8 of CCR2, the division factor of the baud rate generator is adjusted. Depending on the programmed value N in BR9–BR0 (N = 0 – 1023), the division factor k results as follows:

$$k = (N + 1)\ 2$$

Channel Configuration Register 2 (READ/WRITE)—Value after RESET: 00H.

The meaning of the individual bits in CCR2 depends on the selected clock mode via CCR1 as follows:

CCR2 clock mode 0, 1	BCS1	BCS0	0	0	0	CIE	RIE	DIV	(2C/6C)
clock mode 2, 6	BR9	BR8	BDF	TSS	TIO	CIE	RIE	DIV	
clock mode 3, 4, 7	BR9	BR8	BDF	0	TIO	CIE	RIE	DIV	
clock mode 5	BCS1	BCS0	XCS0	RCS0	TIO	CIE	RIE	DIV	

BCS1, BCS0 — Bus Control Signal Selection—valid only in a bus configuration (selected via CCR1).

0 X $\overline{\text{RTS}}$ output is activated during the transmission of a frame, see also Mode register.

1 0 $\overline{\text{RTS}}$ output is always HIGH ($\overline{\text{RTS}}$ disabled).

1 1 $\overline{\text{RTS}}$ indicates the reception of a data frame (active LOW).

BR9, BR8—Baud Rate, Bit 9–8 (higher significant bits, refer to description of BGR register).

BD—Baud Rate Division Factor

0—The division factor of the baud rate generator is set to 1 (constant).

1—The division factor is adjusted with BR9–BR0 bits of CCR2 and BRG register.

TSS—Transmit Clock Source Select

0—The transmit clock is input to the TxCLKA/TxCLKB pins.

1—The transmit clock is derived from the baud rate generator output clock divided by 16.

TIO—Transmit Clock Input Output Switch

0—TxCLKA, TxCLKB pins are inputs

1—TxCLKA, TxCLKB pins are outputs

CIE—Clear To Send Interrupt Enable—any state transition at the $\overline{CTS}$ input pin may cause an interrupt which is indicated in the EXIR register (CSC bit). The actual state at the $\overline{CTS}$ pin can be determined reading the $\overline{CTS}$ bit of the STAR register.

0—disable

1—enable

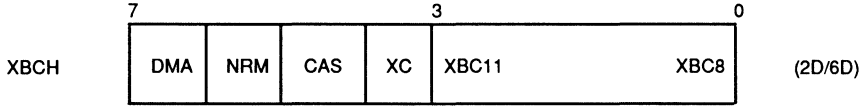
RIE—Receive Frame Start Interrupt Enable—When set, the RFS interrupt (via EXIR) is enabled.

DIV—Data Inversion—Valid only if NRZ data encoding is selected. Data is transmitted and received inverted.

XCS0, RCS0—Transmit/Receive Clock Shift, Bit 0—together with bits XCS2, XCS1 (RCS2, RCS1) in TSAX (TSAR) the clock shift relative to the frame synchronization signal of the transmit (receive) time slot can be adjusted.

A clock shift of 0–7 bits is programmable (clock mode 5 only).

Transmit Byte Count High (WRITE)—Value after RESET: 000xxxxx



DMA—DMA Mode—selects the data transfer mode of HSCX to System Memory.

0—Interrupt controlled data transfer (interrupt mode)

1—DMA controlled data transfer (DMA mode)

NRM—Normal Response Mode—valid in auto mode only. Determines the function of the LAP Controller:

0—full-duplex LAPB/LAPD operation

1—half-duplex NRM operation

CAS—Carrier Detect Auto Start—when set, a HIGH at the CD (AxCLK) pin enables the respective receiver, and data reception is started.

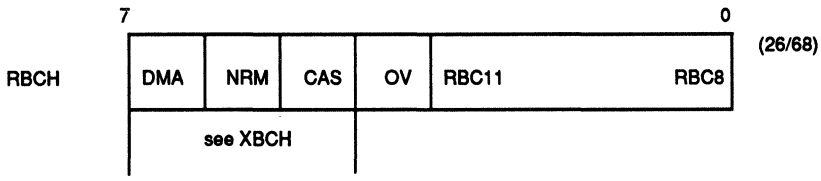
XC—Transmit Continuously—valid only if DMA mode is selected.

If the XC bit is set, the HSCX continuously requests for transmit data ignoring the transmit byte count programmed via XBCX, XBCL.

XBC11–XBC8—Transmit Byte Count (most significant bits)—valid only if DMA mode is selected.

Together with XBCL (bits XBC7–BC0) the length of the frame to be transmitted is programmed.

Received Byte Count High (READ)—Value after RESET: 000xxxxx

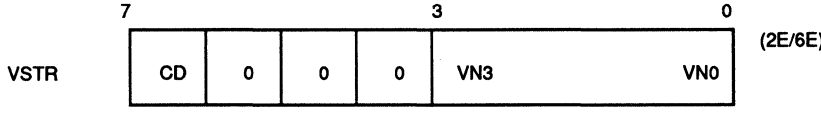


DMA, NRM, CAS—these bits represent the read-back value programmed in XBCH (see XBCH).

OV—Counter Overflow—more than 4095 bytes received. The received frame exceeded the byte count in RBC11 . . . RBC0.

RBC11–RBC8—Receive Byte Count (most significant bits)—together with RBCL (bits RBC7–RBC0) the length of the received frame can be determined.

Version Status Register (READ)



CD—Carrier Detect—this bit represents the inverted state at the CD (AxCLK) pin.

1—CD active (LOW)

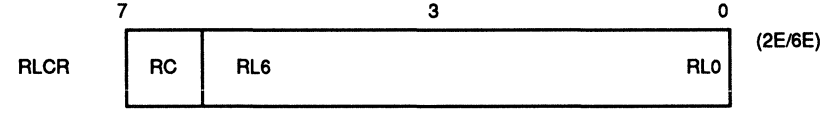
0—CD inactive (HIGH)

VN3 . . . VN0 . . . Version Number of Chip

0—Version A1

2—Version A2

Receive Length Check Register (WRITE)



RC—Receive Check (on/off)

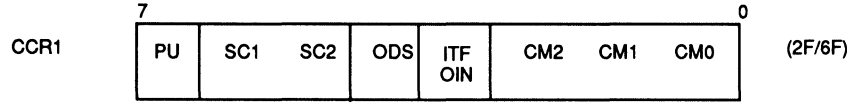
0—Receive length check feature disabled

1—Receive length check feature enabled

RL—Receive Length—the maximum receive length after which data reception is suspended can be programmed here. Depending on the value RL programmed via RL6–RL0, the receive length is (RL + 1) · 32 bytes.

A frame exceeding this length is treated as if it were aborted by the opposite station (RME Interrupt, RAB bit set). In this case, the receive byte count (RBCH, RBCL) is greater than the programmed receive length.

Channel Configuration Register 1 (READ/WRITE)—Value after RESET: 00H



PU—Switches Between Power Up and Power Down Mode

0—power down (standby)

1—power up (active)

SC1, SC0... Serial Port Configuration

00—NRZ data encoding

10—NRZI data encoding

01—bus configuration, timing mode 1

11—bus configuration, timing mode 2

Note: If bus configuration is selected, only NRZ coding is supported.

ODS... Output Driver Select—defines the function of the transmit data pins (TxDA, TxDB)

0—TxD pins are open drain outputs

1—TxD pins are push-pull outputs

ITF/OIN—Interframe Time Fill/One Insertion—the function of this bit depends on the selected serial port configuration (bit SC1):

- Point-to-point configurations: ITF

  Determines the idle (= no data to send) state of the transmit data pins (TxDA, TxDB)

0 — Continuous idle sequences are output (TxD pins remain in the "1" state)

1—Continuous flag sequences are output ("01111110" bit patterns)

- Bus configurations: OIN

  In bus configurations, the ITF is implicitly set to "0"; that is, continuous "1"s are transmitted, and data encoding is NRZ.

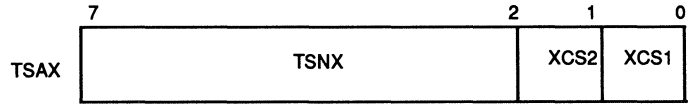
  When this bit is set, a "ONE" insertion (deletion) mechanism is activated, inserting a "1" after seven consecutive "0"s in the transmit data stream or deleting a "1" in the receive data stream.

  Similar to the HDLC's bit-stuffing mechanism (inserting a "0" after five consecutive "1"s), this method proves to be advantageous when the receive clock is recovered from the receive data stream by means of DPLL, because it is guaranteed that at least after seven bits a transition occurs in the receive data as in case of long "0" sequences.

CM2, CM1, CMO—Clock Mode—selects one of the eight different clock modes:

000	clock mode 0
.	.
.	.
.	.
111	clock mode 7

Time Slot Assignment Register Transmit (WRITE)



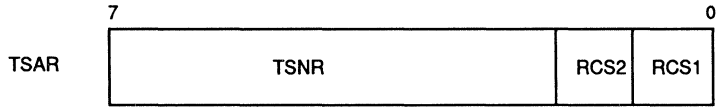
(30/70)

This register is used only in clock mode 5.

TSNX—Time Slot Number Transmit—selects one of up to 64 possible time slots (00H–3FH) in which data is transmitted. The number of bits per time slot can be programmed via XCCR.

XCS2, XCS1—Transmit Clock Shift, Bit 2–1—together with bit XCS0 in CCR2, the transmit clock shift can be adjusted.

Time Slot Assignment Register Receive (WRITE)



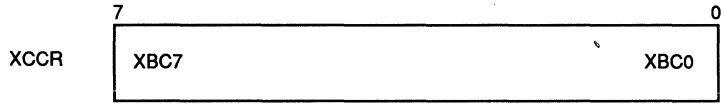
(31/71)

This register is used only in clock mode

TSNR—Time Slot Number Receive—defines one of up to 64 possible time slots (00H–3FH) in which data is received. The number of bits per time slot can be programmed via RCCR.

RCS2, RCS1—Receive Clock Shift, Bit 2–1—together with bit RCS0 in CCR2, the receive clock shift can be adjusted.

Transmit Channel Capacity Register (WRITE)—Value after RESET: 00H

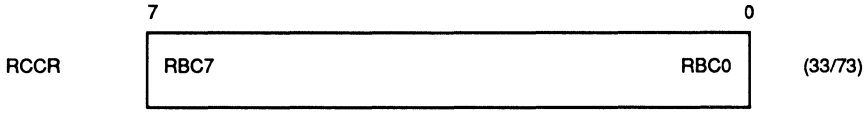


(32/72)

XBC7–XBC0—Transmit Bit Count, Bit 7–0—defines the number of bits to be transmitted with a time slot:

Number of bits = XBC + 1 (1–256 bits/time slot)

Receive Channel Capacity Register (WRITE)—Value after RESET: 00H



RBC7–RBC0—Receive Bit Count, Bit 7–0—defines the number of bits to be received within a time slot:

Number of bits = RBC + 1 (1–256 bits/time slot)

ABSOLUTE MAXIMUM RATINGS

Ambient temperature under bias . . . . . –55 to +125°C

Voltage on any pin with
respect to ground . . . . . –0.25 to V_{DD} +0.25 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Ambient Temperature (T_A) . . . . . 0 to +70°C

Supply Voltage (V_{CC}) . . . . . +4.75 to +5.25 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS

$T_A = 0$ to $+70°C$; $V_{DD} = 5\ V \pm 5\%$, $V_{SS} = 0\ V$.

Table 1. DC Characteristics

Symbol	Parameter	Limit Values Min.	Max.	Unit	Test Condition
V_{IL}	Input low voltage	–0.4	0.8	V	
V_{IH}	Input high voltage	2.0	$V_{CC} + 0.4$	V	
V_{OL}	Output low voltage		0.45	V	$I_{OL} = 2$ mA
V_{OH}	Output high voltage	2.4		V	$I_{OH} = -400\ \mu A$
V_{OH}	Output high voltage		$V_{DD} - 0.5$	V	$I_{OH} = -100\ \mu A$
I_{CC}	Power supply current operational		8	mA	$V_{DD} = 5$ V inputs at 0 V/V_{DD}, no output loads
	Power supply current power down		1.5	mA	
I_{LI}	Input leakage current		10 µa		0 V < V_{IN} < V_{DD} to 0 V
I_{LO}	Output leakage current				0 V < V_{OUT} < V_{DD} to 0 V

CAPACITANCES

$T_A = 25°C$, $V_{DD} = 5\ V \pm 5\%$, $V_{SS} = 0\ V$.

Table 2. Capacitances

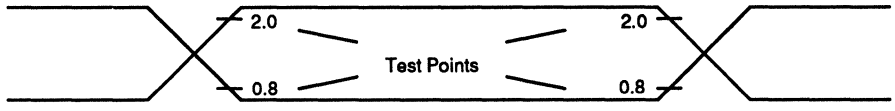
Symbol	Parameter	Type	Limit Values Max	Unit	Test Condition
C_{IN}	Input capacitance	5	10	pF	
C_{OUT}	Output capacitance	10	20	pF	
C_{IO}	I/O	8	15	pF	

SWITCHING CHARACTERISTICS

T_A = 0 to +70°C, V_{DD} = 5 V ±5%.

Inputs are driven to 2.4 V for a logical "1" and to 0.4 V for a logical '0'. Timing measurements are made at 2.0 V for a logical "1" and at 0.8 V for a logical "0".

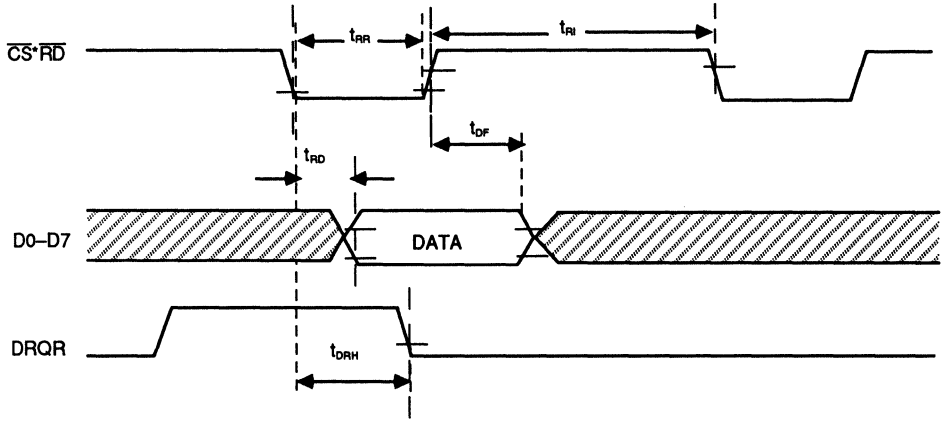
The AC testing input/output waveforms are shown below.



11139-010B

Figure 10. Input/Output Waveform for AC Tests

Microcontroller Interface Timing



11139-011B

Figure 11. µP Read Cycle

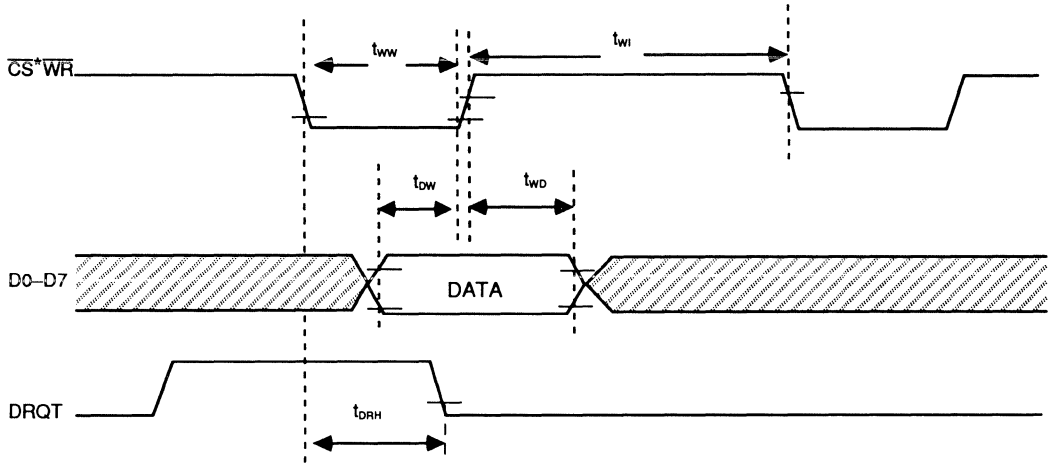


Figure 12. µP WriteCycle

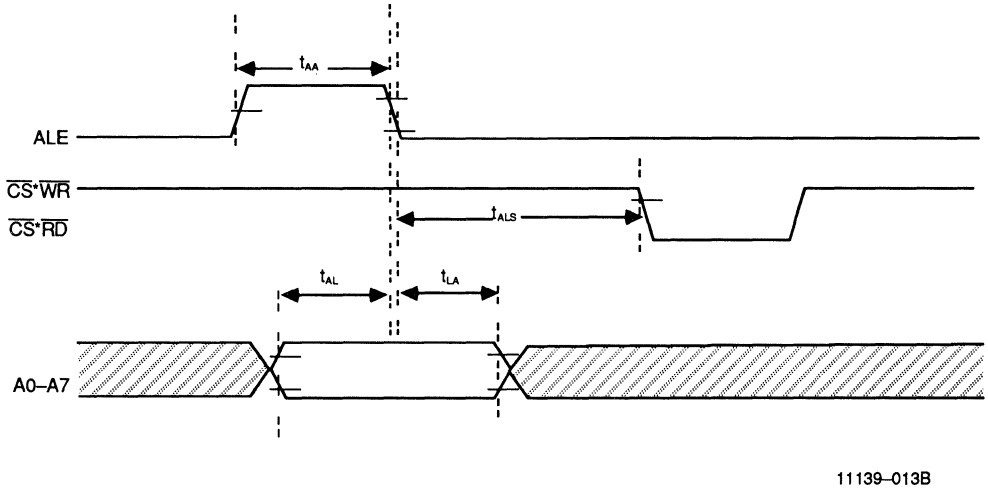
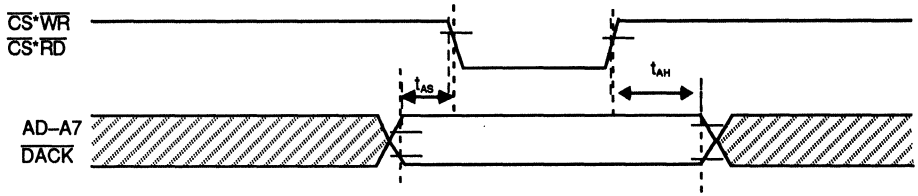


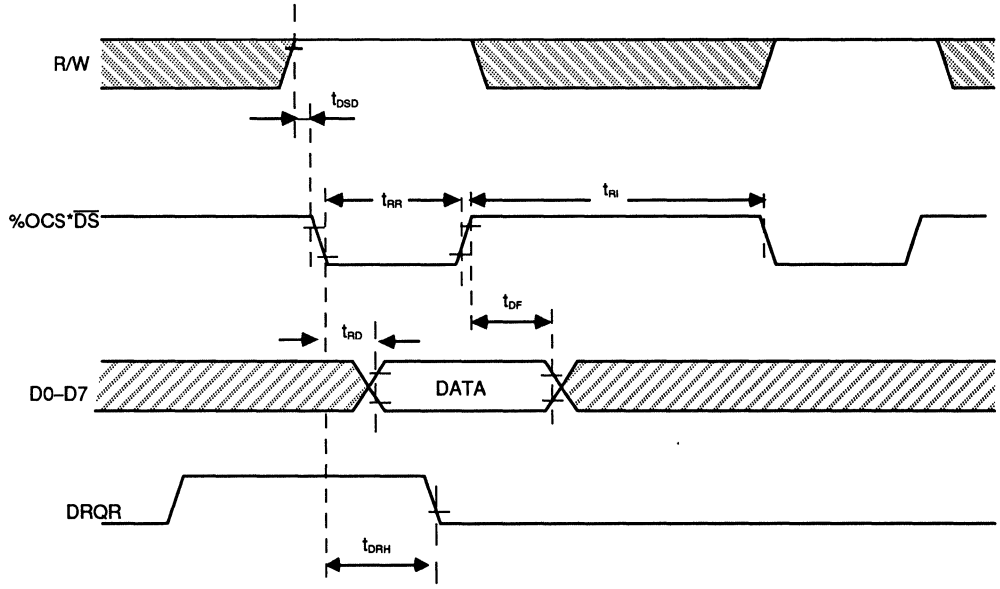
Figure 13. Multiplexed Address Timing



11139–014B

Figure 14. Address Timing

Motorola Bus Mode



11139–015B

Figure 15. µP Read Cycle

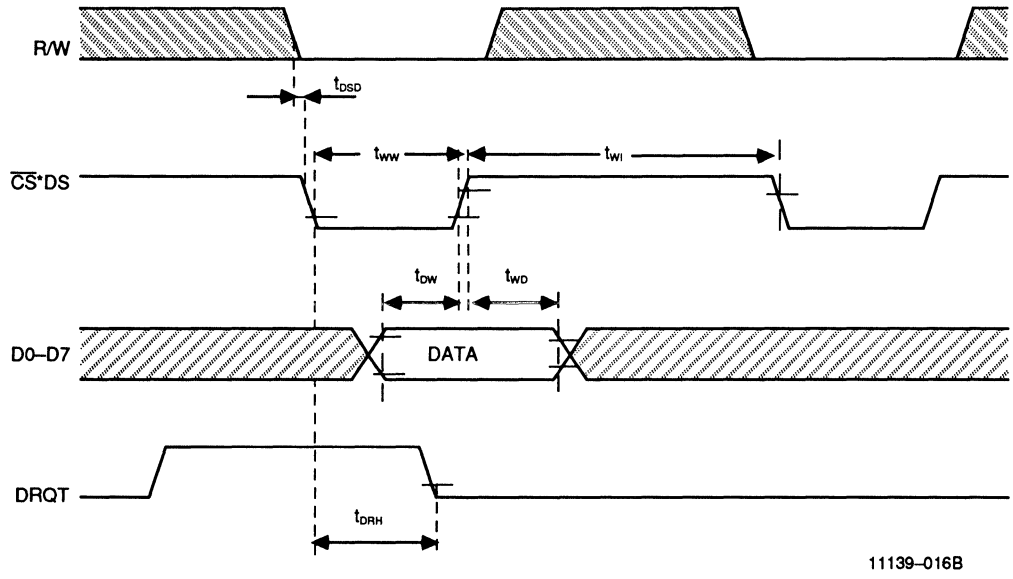


Figure 16. µP Write Cycle

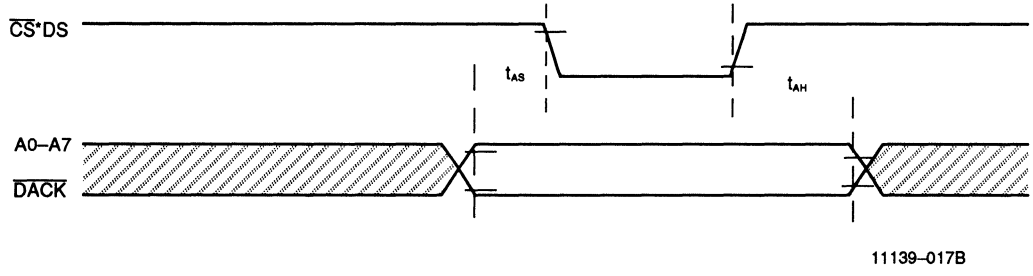


Figure 17. Address Timing

Table 3. Microcontroller Interface Timing Characateristics

Parameter	Symbol	Limit Values Min.	Max.	Unit	Test Condition
ALE pulse width	t_{AA}	30	–	ns	–
Address setup time to ALE	t_{AL}	10	–	ns	–
Address hold time from ALE	t_{LA}	20	–	ns	–
Address latch setup time to $\overline{WR}$, $\overline{RD}$	t_{ALS}	0	–	ns	–
Address setup time to $\overline{WR}$, $\overline{RD}$	t_{AS}	10	–	ns	–
Address hold time from $\overline{WR}$, $\overline{RD}$	t_{AH}	20	–	ns	–
DMA request delay	t_{DRH}	–	75	ns	–
DS delay after R/W setup	t_{DSD}	0	–	ns	–
$\overline{RD}$ pulse width	t_{RR}	120	–	ns	–
Data output delay from $\overline{RD}$	t_{RD}	–	120	ns	–
Data float delay from $\overline{RD}$	t_{DF}	–	25	ns	–
$\overline{RD}$ control interval	t_{RI}	70	–	ns	–
$\overline{WR}$ pulse width	t_{WW}	60	–	ns	–
Data setup time to $\overline{WR}$+$\overline{CS}$	t_{DW}	30	–	ns	–
Data hold time from $\overline{WR}$+$\overline{CS}$	t_{WD}	10	–	ns	–
$\overline{WR}$ control interval	t_{WI}	70	–	ns	–

Table 4. Serial Interface Timing

Parameter	Symbol	Limit Values Min.	Max.	Unit	Test Condition
Receive data setup	t_{RDS}	5		ns	
Receive data hold	t_{RDH}	30		ns	
Collision data setup	t_{CDS}	0		ns	
Data hold	t_{CDH}	20		ns	
Transmit data delay	t_{XDD}	20	70	ns	
Request to send delay 1	t_{RTD1}	30	120	ns	
Request to send delay 2	t_{RTD2}	20	85	ns	
Clock period	t_{CP}	240		ns	
Clock period Low	t_{CPL}	90		ns	
Clock period High	t_{CPH}	90		ns	

Table 5. Clock Mode 1

Parameter	Symbol	Limit Values Min.	Max.	Unit	Test Condition
Receive strobe delay	t_{RSD}	30		ns	
Receive strobe setup	t_{RSS}	60		ns	
Receive strobe hold	t_{RSH}	30		ns	
Transmit strobe delay	t_{XSD}	30		ns	
Transmit strobe setup	t_{XSS}	60		ns	
Transmit strobe hold	t_{XSH}	30		ns	
Transmit data delay	t_{XDD}		70	ns	
Strobe data delay	t_{SDD}		90	ns	
High impedance from clock	t_{XCZ}		50	ns	
High impedance from strobe	t_{XSZ}		50	ns	

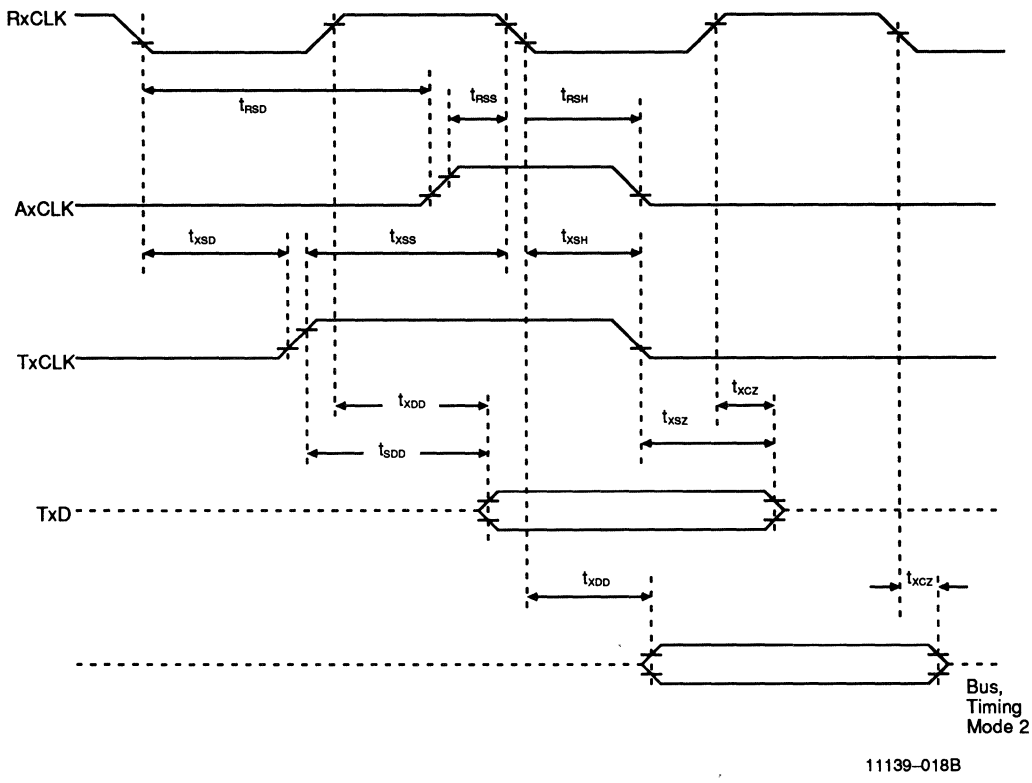
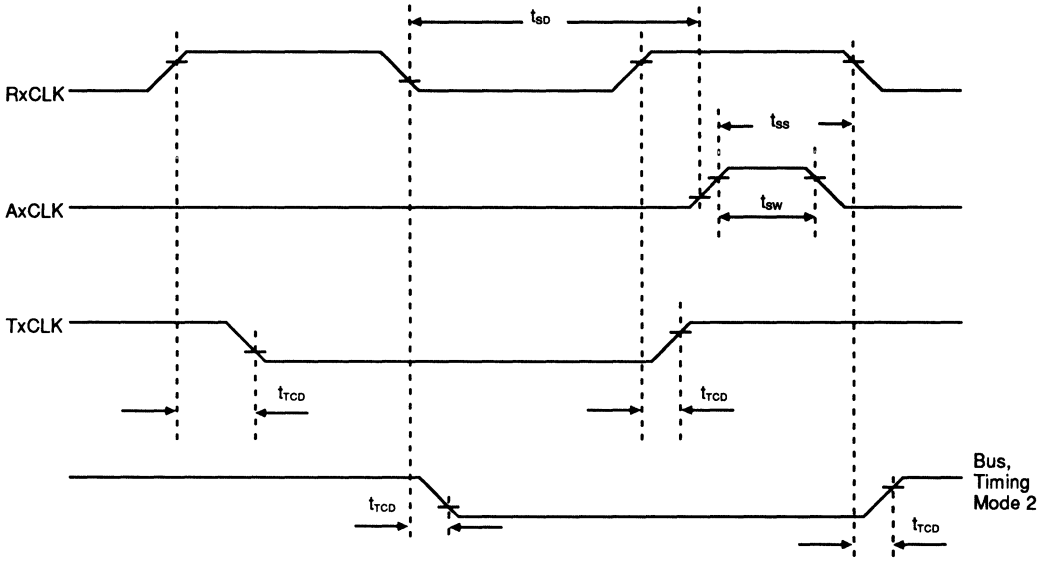


Figure 18. Strobe Timing

Clock Mode 5



11139-019B

Figure 19. Synchronization Timing

Table 6. Clock Mode 5 Timing

Parameter	Symbol	Limit Values Min.	Max.	Unit
Sync pulse start early *	t_{SSE}	30		ns
Sync pulse start late **	t_{SSL}	0	30	ns
Sync pulse width	t_{SW}	40		ns
Time-slot control delay	t_{CD}	20	75	ns

* If sync pulse starts before ↓ edge of RxCLK A/B first bit transmitted occurs on ↓ edge of RxCLK A/B.

** If sync pulse occurs after ↓ edge of RxCLK A/B first bit transmitted occurs on ↓ edge of sync pulse.

Clock Mode 2, 3, 6, 7

Table 7. Internal Clocking

Description	Min.	Max.	Unit	Condition
Clock frequency		12	MHz	Baud rate generator used
Clock frequency		19.3	MHz	Baud rate generator not used

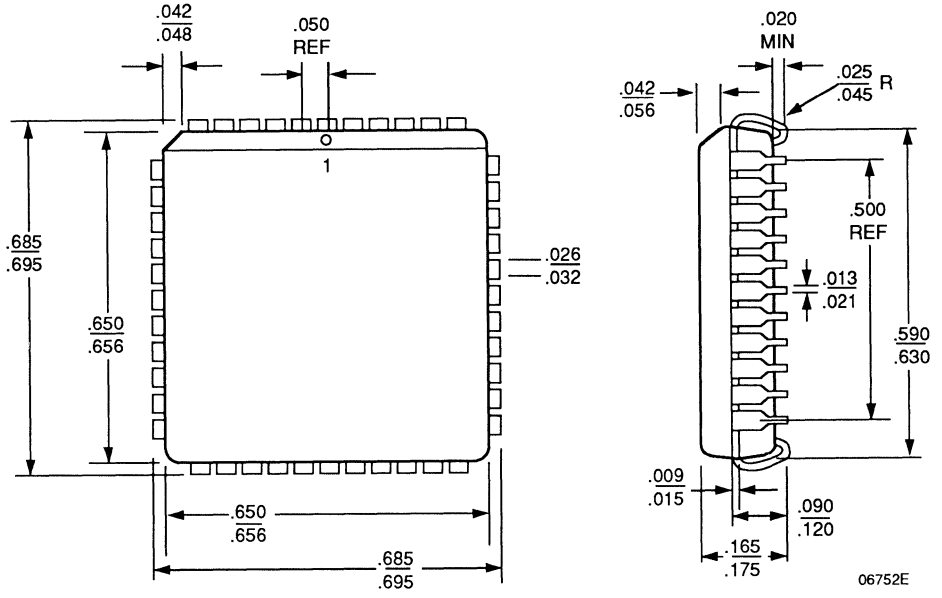
Reset Timing

Table 8. RES Characteristics

Symbol	Description	Min.	Max.	Unit	Condition
t_{RWH}	RES High	1800		ns	

PHYSICAL DIMENSIONS

Plastic package, PLCC, 44 pins



2

Advanced
Micro
Devices

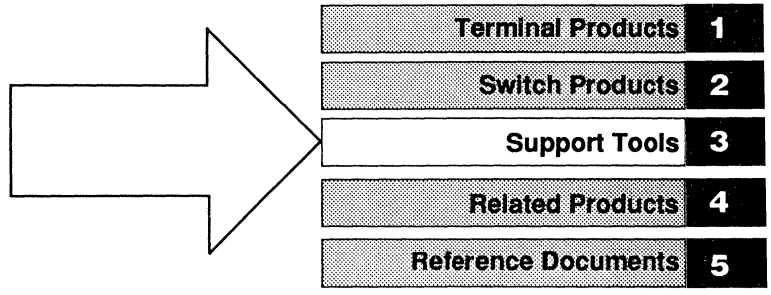


Table of Contents

ISDN AmLink
Interface Reference Guide

June 1989

3

Publication # 09529 | Rev. B | Amendment /0
Issue Date: June 1989

DISTINCTIVE CHARACTERISTICS

The following list describes the distinctive characteristics of AMD's AmLink™ LAPD/LAPB:

- Full implementation of the CCITT Q.921 recommendations
- Primitive-driven software interface
- Multi-packet data queues between Layers 2 and 3 expediting data transmissions between layers
- Supporting both D- and B-channels simultaneously
- Configurable link parameters
- Modular design
- State table implementation allowing flexibility, structured design, and ease of maintenance
- Device hardware independent
- Operating system independent
- Supports multiple TEI's per each physical channel
- Supports multiple logical links per each TEI
- May be implemented in firmware
- Portable source code in "C"

Chapter 2

GENERAL DESCRIPTION

This document describes the AmLink LAPD/LAPB interface between the LAPD/LAPB and the Layer_3 entity, and the LAPD/LAPB and the Layer_2 Management Entity (ME). Layer_3 and Layer_2 are defined by the Open Systems Interconnection (OSI) model of the International Standards Organization. In this document, all references made to LAPD equally apply to LAPB.

2.1 Purpose

The purpose of the Link Access Procedure for the D-channel (LAPD) is to transfer information between Layer_3 entities across the ISDN network using the D-channel.

The purpose of the Link Access Procedure Balanced (LAPB) is to transfer information between Layer_3 entities using the B-channel.

The details of LAPD are provided in Recommendation Q.921 (I.441 ISDN user-network interface data link layer specification). Layer_3 is defined in Recommendations Q.930 (I.450 ISDN user network interface Layer_3 general aspects) and Q.931 (I.451 ISDN user-network interface Layer_3 specification).

2.2 System Requirements

The AmLink LAPD is designed so that it requires minimum operating system assistance for its proper operation. It contains the required mechanism to support multiple logical links for both D- and B-channels. However, it can be altered to take advantage of any multitasking features that the local operating system may have. For timer and queue services, it generates proper MDL primitive requests as described in the following sections.

The system requirements of AmLink are:

- 25 Kbytes of RAM/ROM
- 1000 bytes of private RAM
- 512 minimum bytes of RAM per Logical Link
- Memory allocation/de-allocation from the Management Entity
- Timer services from the Management Entity
- Low-Level Device Drivers
- Shared memory per physical channel is 48 bytes plus user-supplied buffers

2.3 Architecture

Figure 1 shows the block diagram of the interface between Layers 1 through 3. The Am79C401 Integrated Data Protocol Controller (IDPC™) and Am79C30 Digital Subscriber Controller (DSC™) components provide the physical connection or Layer_1 of the OSI model. The DSC provides the D-channel connections, whereas the IDPC resides above the B-channel. The low-level drivers provide an identical interface between the link layer and the physical layer. The interface between the link layer and these drivers is referred to as the Layer_2 to Layer_2⁻ interface. The Management Entity (ME) provides the necessary support for all layers, and is in charge of providing and maintaining different timers and storage areas as requested by the various layers.

The design of the Management Entity is system dependent and is generally accomplished by the host. In this implementation, however, a sample management entity is developed which operates in

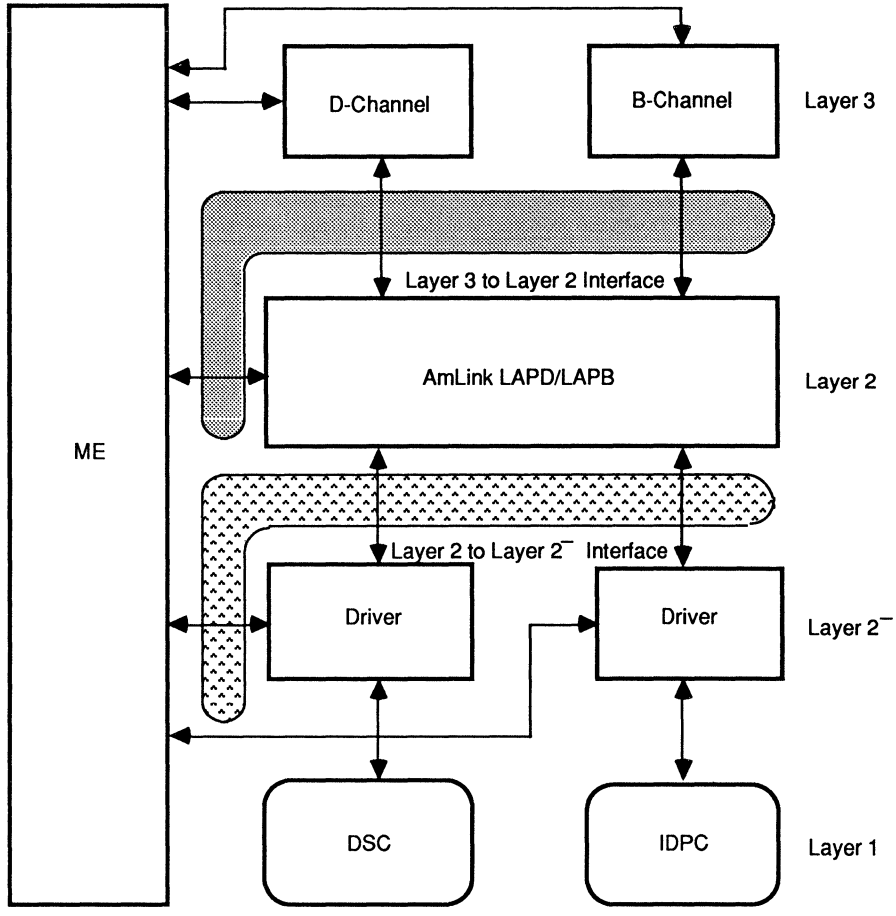


Figure 1. OSI Layers 1 Through 3

the foreground, whereas drivers, LAPD, and Layer_3 will be primarily executed as interrupt-driven tasks.

2.4 Development Environment

The AmLink LAPD is implemented using Microsoft "C" compiler version 5.1 and the Microsoft Macro Assembler version 5.1.

FUNCTIONAL DESCRIPTION

This section describes the communication interface between LAPD and Layer_3, and between LAPD and the Management Entity.

3.1 POR Configuration and Initialization

Prior to using this code, several initialization tasks must be carried out as indicated below:

1) Installation of all input_service_handler vectors. These are entry points to the three input processors inside LAPD: one from the Low-Level Driver (LLD) to LAPD, one from Layer_3 to the LAPD, and one from the Management Entity to the LAPD.
2) Installation of all output_service_handler vectors. These are the entry points to the interrupt handlers outside LAPD: one that processes LAPD to LLD commands, one that processes LAPD to ME events, and one that processes LAPD to Layer_3 events.
3) Executing the initlap() routine.

After executing the initlap() routine, the LAPD will be able to respond to input commands and responses.

3.2 Event/Command Mailboxes

The communication interface between Layer_2 and Layer_3 and between Layer_2 and the Management_Entity takes place via a set of communication mailboxes, where each mailbox has the following structure:

Event/Command Identifier Field
Receipt_Acknowledge_Field
Parameters

The entity initiating the communication has the task of:

1) Loading the event/command identifier field with the code associated with each particular primitive.
2) Placing FF into the Receipt_Acknowledge_Field.
3) Setting the parameters associated with each primitive.
4) Generating a software (or hardware if implemented on separate processors) interrupt for the recipient, who removes the information from the mailbox and acknowledges the receipt by loading a valid acknowledgment byte into the Receipt_Acknowledge_Field.

In case of 2-byte-long parameters, the least significant byte is loaded first. For example, if LENGTH is a 2-byte-long parameter, the parameter field will have the following format:

Octet	1	Low order	8 bits of LENGTH (LSB)
	2	High order	8 bits of LENGTH (MSB)

In case of address parameters, the parameter will occupy 4 bytes and will have the format as described below:

Octet	1	Low order	8 bits of the OFFSET
	2	High order	8 bits of the OFFSET
	3	Low order	8 bits of the SEGMENT
	4	High order	8 bits of the SEGMENT

3.3 Layer_3 and Layer_2 Communication Interface

The Layer_2 and Layer_3 exchange information via command and event mailboxes is shown in Figure 2. There are separate sets of command and event mailboxes for each of the B- and D-channels. All primitive exchanges between Layer_2 and Layer_3 have the DL prefix.

3.3.1 Layer_3 to Layer_2 Interface

The Layer_3 to Layer_2 (LAPD) commands are communicated via the L3_L2_Command_Mail_Box. Layer_3 must provide a primitive code that signifies one of the Layer_3 to Layer_2 primitives and should also indicate the Connection Endpoint Identifiers (CEI) associated with its peer Layer_3, which is the intended final recipient of the information as far as Layer_3 initiating the primitive is concerned.

Each CEI is composed of a Service_Access_Point_Identifier (SAPI) and a Connection_Endpoint_Suffix (CES). After loading the L3_L2_Command_Mail_Box with the appropriate data, Layer_3

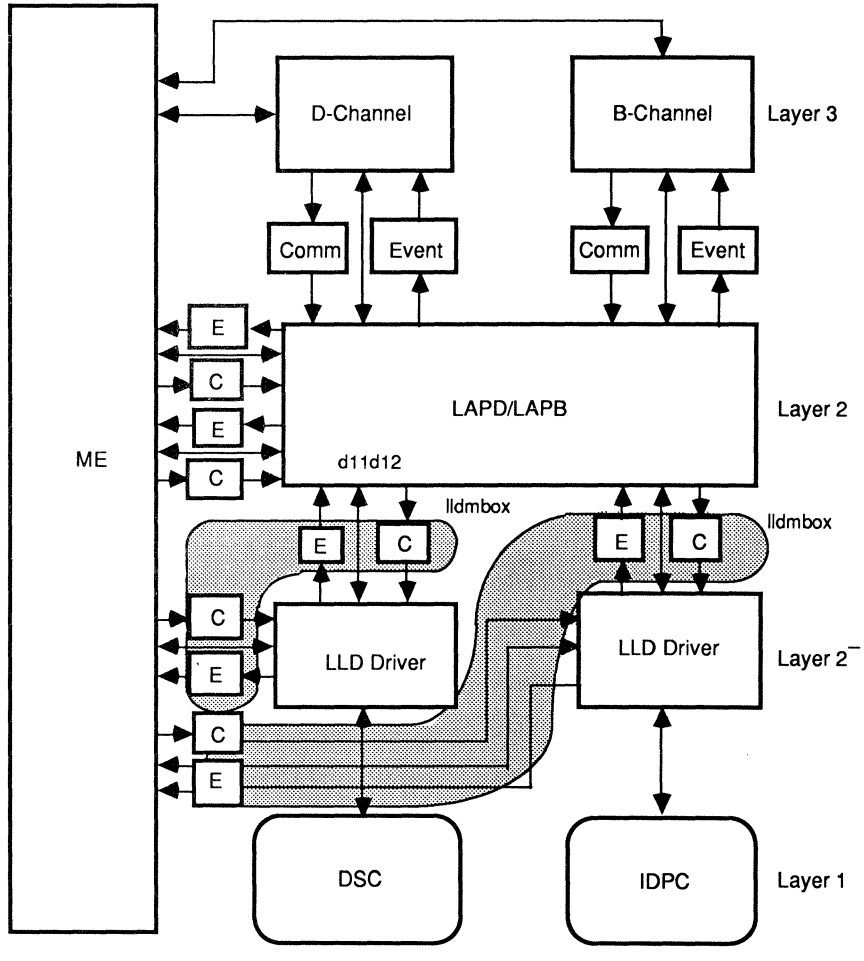


Figure 2. Command and Event Mailbox Structures

should generate the il3l2 interrupt in case of B-channel data transfers and the dl3l2 interrupt for the D-channel communication to activate Layer_2. The letter "i" preceding the interrupt name signifies that the interrupt is for the IDPC device that resides above the B-channel; similarly, "d" stands for DSC device controlling the D-channel. The direction for the interrupt is shown by the order of the layer numbers; in this case, l3l2 implies that the direction is from Layer_3 to Layer_2.

The Layer_3 to Layer_2 primitives, along with their command codes and parameters, are described below:

Code	Primitive	Parameters	Byte
DL_EST_RQ	DL_ESTABLISH_REQUEST	SAPI	1
		CES	1
DL_REL_RQ	DL_RELEASE REQUEST	SAPI	1
		CES	
DL_DA_RQ	DL_DATA_REQUEST	SAPI	1
		CES	1
		I_Message_Address	4
		I_Message_Length	2
		Refno	2
		Available_Slots	1
DL_U_DA_RQ	DL_UNIT_DATA_REQUEST	SAPI	1
		CES	1
		UI_Message_Address	4
		UI_Message_Length	2
		Refno	2

The code is a constant declaration that identifies a primitive.

The UI_Message_Address field points to the buffer containing the Unacknowledged Information that is being transmitted, and follows the format used for address pointers as described in Section 3.2.

The Available_Slots parameter indicates to Layer_3 the number of available buffer slots it has for subsequent DL_DATA_REQUESTs.

The UI_Message_Length field specifies the length of the UI message and it follows the format used for 2-byte-long parameters. The Refno indicates the reference number of the buffer that contains this UI message. This reference number is used to release the memory space whenever the message is successfully transmitted by the low-level driver.

The allowed values for the Receipt_Acknowledge_Field are:

Code	Definition
FF	Set by Layer_3
00	Set by LAPD to indicate successful reception
11	Set by LAPD to indicate error in the command

The actions to be taken by the LAPD in response to the commands received from Layer_3 are described in Q.921 CCITT Recommendations (I.441).

3.3.2 Layer_2 to Layer_3 Interface

The Layer_2 to Layer_3 communication is provided via the L2_L3_Event _Mail_Box. The structure of this mailbox is described in Section 3.2. After loading the appropriate parameters into the L2_L3_Event_Mail_Box, the LAPD should generate the il2il3 interrupt for the B-channel communication and the dl2l3 interrupt for the D-channel transfers to pass control to the Layer_3 entity.

The Layer_2 to Layer_3 primitives, along with the event codes and parameters associated with each primitive, are described below:

Code	Primitive	Parameters	Byte
DL_EST_IN	DL_ESTABLISH_INDICATION	SAPI	1
		CES	1
DL_REL_IN	DL_RELEASE_INDICATION	SAPI	1
		CES	1
DL_U_DA_IN	DL_UNIT_DATA_INDICATION	SAPI	1
		CES	1
		UI_Message_Address	4
		UI_Message_Length	2
		Refno	2
DL_DA_IN	DL_DATA_INDICATION	SAPI	1
		CES	1
		I_Message_Address	4
		I_Message_Length	2
		Refno	2
DL_EST_CON	DL_ESTABLISH_CONFIRM	SAPI	1
		CES	1
DL_REL_CON	DL_RELEASE_CONFIRM	SAPI	1
		CES	1

The code is a constant declaration that uniquely identifies a primitive.

The 2-byte message length parameter and the 4-byte message address parameter follow the formats described in Section 3.2.

The allowed values for the Receipt_Acknowledge_Field are:

Code	Definition
FF	Set by LAPD
00	Set by Layer_3 to indicate successful receipt
11	Set by Layer_3 to indicate error in the event

The actions to be taken by the LAPD before providing the indication/confirmation to Layer_3 are described in Q.921 CCITT Recommendations (I.441).

3.4 Layer_2 to Management Entity Interface

The Layer_2 and Layer_2 Management Entity exchange information in a set of command/event mailboxes as shown in Figure 2. There is a separate command/event mailbox for each of the

B- and D-channels. The primitives exchanged between Layer_2 and the Management Entity have the MDL prefix.

3.4.1 Management Entity to LAPD Interface

The Management Entity to LAPD commands are communicated via the ME_L2_Command_Mail_Box. The structure of this mailbox is described in Section 3.2. To start the communication, the ME should load the mailbox with the primitive code and parameters, and generate an appropriate interrupt (imel2 for the B-channel and dmel2 for the D-channel) to transfer control to the LAPD.

The Management Entity to Layer_2 primitives, along with their command codes and parameters lists, are shown below:

Code	Primitive	Parameters	Byte
MDL_AS_RQ	MDL_ASSIGN_REQUEST	TEI	1
		CES	1
MDL_R_RQ	MDL_REMOVE_REQUEST	TEI	1
MDL_U_DA_RQ	MDL_UNIT_DATA_REQUEST	UI_Message_Address	4
		UI_Message_Length	2
		Refno	2
MDL_X_RQ	MDL_XID_REQUEST	SAPI	1
		CES	1
		XID_Message_Address	4
		XID_Message_Length	2
		Refno	2
MDL_X_RS	MDL_XID_RESPONSE	SAPI	1
		CES	1
		XID_Message_Address	4
		XID_Message_Length	2
		Refno	2
MDL_ER_RS	MDL_ERROR_RESPONSE	SAPI	1
		CES	1
		Code	1

The code is a constant declaration that uniquely identifies a primitive.

The message length, reference number (Refno), and address parameters follow the format described in Section 3.2.

The error codes used in the MDL_Error_Response primitive are defined in Q.921 Specifications Appendix II.

The receipt codes allowed for the Receipt_Acknowledge_Field are:

Code	Definition
FF	Set by ME
	Set by LAPD to indicate successful receipt
	Set by LAPD to indicate error

The actions to be taken by the LAPD in response to the commands received from the ME are described in Q.921 CCITT Recommendations (I.441).

3.4.2 Layer_2 to Management Entity Interface

The Layer_2 to ME communication is provided via the L2_ME_Event_Mail_Box. The format for this mailbox is described in Section 3.2. To start the communication, the LAPD will place the event code and its associated parameters into the mailbox, and generate an appropriate interrupt (il2m3 for the B-channel and dl2me for the D-channel) to transfer control of the Management Entity.

The Layer_2 to Management Entity primitives, along with their codes and parameters, are shown below:

Code	Primitive	Parameters	Byte
MDL_AS_IN	MDL_ASSIGN_INDICATION	CES	1
MDL_U_DA_IN	MDL_UNIT_DATA_INDICATION	UI_Message_Address	4
		UI_Message_Length	2
		Refno	2
MDL_X_IN	MDL_XID_INDICATION	SAPI	1
		CES	1
		XID_Message_Address	4
		XID_Message_Length	2
		Refno	2
MDL_X_CON	MDL_XID_CONFIRM	SAPI	1
		CES	1
		XID_Message_Address	4
		XID_Message_Length	2
		Refno	2
MDL_ER_IN	MDL_ERROR_INDICATION	SAPI	1
		CES	1
		Error_code	1

The message length parameters, reference number (Refno), and message address parameters follow the formats shown in Section 3.2.

The error codes used in the MDL_Error_Indication primitive are defined in Q.921 Specifications Appendix II.

The receipt codes allowed for the Receipt_Acknowledge_Field are:

Code	Definition
FF	Set by LAPD
00	Set by ME to indicate successful receipt
11	Set by ME to indicate error

The actions to be taken by the LAPD before providing the indication/confirmation to the ME are described in Q.921 CCITT Recommendations (I.441).

3.5 Timer and Queue Services Required by LAPD

In addition to the services that are specified in Q.921 specifications, the Layer_2 Management Entity must also provide timer and storage services for Layer_2.

The two new primitives added to provide these new services for the LAPD are:

MDL_TIMER_SERVICE and
MDL_BUFFER_SERVICE

3.5.1 MDL Timer Services

The Layer_2 Management Entity must provide the following timer services for the LAPD:

MDL_TIMER_SERVICE_INDICATION and
MDL_TIMER_SERVICE_RESPONSE

3.5.1.1 MDL_TIMER_SERVICE_INDICATION

The MDL_TIMER_SERVICE_REQUEST is generated by the LAPD to start, stop, or restart a specific timer. To initiate this service, the LAPD should place the MDL primitive code with its associated parameters into the L2_ME_Event_Mail_Box and generate an interrupt to transfer control to the Layer_2 Management Entity.

The parameters associated with the MDL_Timer_Service primitive are shown below:

Code	Parameter	Length (byte)
MDL_TM_IN	pppppimm	1
	Reference number	2
	Time_out value	2
	Pointer to var	4

The code is a constant declaration that uniquely identifies a primitive.

The "ppppp" field within the first parameter indicates a priority number ranging from 0 (low) to 31 (high), and will be implemented at a later date. Currently, priority will be set to 0 for all timers.

If the "i" field in the first parameter is set to "0," it indicates that an interrupt should be generated when the timer times-out. If this field is set to "1," a global variable is set when the timer times-out.

The "mm" fields in the first parameter modify the command such that:

mm = 00—start timer
= 01—stop timer
= 10—restart timer
= 11—reserved

The reference number is a 16-bit-long parameter which identifies a unique timer.

The pointer to var parameter is used to indicate the global variable that will be set when the timer times-out (if i = 1).

All 2- and 4-byte parameters follow the specification format presented in Section 3.2.

The receipt codes allowed for the Receipt_Acknowledge_Field are:

Code	Definition
FF	Set by LAPD
00	Set by ME to indicate successful receipt
11	Set by ME to indicate error

3.5.1.2 MDL_TIMER_SERVICE_RESPONSE

The MDL_TIMER_SERVICE_RESPONSE is generated by the Layer_2 Management Entity whenever any one of the timers initiated by a previous MDL_TIMER_SERVICE_INDICATION primitive time out.

To initiate this primitive, the Layer_2 Management Entity should load the MDL_TM_RS primitive code along with its associated parameters into the ME_L2_Command_Mail_Box and generate an interrupt to activate Layer_2. The parameters used for this primitive are shown below:

Code	Parameters	Length (byte)
MDL_TM_RS	Reference number	2

The code is a constant declaration that identifies the primitive.

The reference number identifies the timer that has timed out. This 2-byte number follows the format used for representing 2-byte-long parameters as shown in Section 3.2.

The receipt codes allowed for the Receipt_Acknowledge_Field are:

Code	Definition
FF	Set by ME
00	Set by LAPD to indicate successful receipt
11	Set by LAPD to indicate error

3.5.2 Memory Management Services

The Management Entity has the tasks of providing the various queues for the different entities. To allow for this the following primitive is used:

MDL_BUFFER_SERVICE_INDICATION

3.5.2.1 MDL_BUFFER_SERVICE_INDICATION

The MDL_BUFFER_SERVICE_INDICATION is generated by the LAPD to request a block of memory area. Because LAPD must immediately receive a response to its buffer service request, the same request mailbox is used to pass back the parameters that identify the designated buffer area (if the response is granted).

To initiate this request, LAPD should load the L2_ME_Response_Mail_Box with the MDL_BF_IN primitive code and its associated parameters, and generate an interrupt for the Layer_2 Management Entity. The Layer_2 Management Entity will either deny or grant this request. If the request is granted, then the Management Entity will load the address parameter with the address of the granted buffer.

The primitive code and parameters for this request are listed below:

Code	Parameters	Length
MDL_BF_IN	——— m	1
	Buffer size in byte	2
	Buffer address	4
	Refno	2

The code is a constant declaration identifying the primitive.

The "m" field in the first parameter is set by LAPD, and if it is equal to 0, then the request is for allocation; otherwise, it is a de-allocation request.

In case of an allocation request, the size field (loaded by LAPD) indicates the minimum size of the requested buffer. The format of this parameter is described in Section 3.2.

The address parameter is loaded by the Management Entity if the request is granted. This parameter points to the granted buffer. The format of this parameter is described in Section 3.2.

For an allocation request (m = 0), the Refno field is loaded by the Management Entity and it uniquely identifies a buffer. During de-allocation (m = 1), the Refno parameter (loaded by LAPD) indicates the buffer that is to be de-allocated.

The allowed values for the_Receipt_Acknowledge_Field are:

Code	Definition
FF	Set by LAPD
00	Set by ME if request is granted
11	Set by ME if request is denied

INTERFACE SPECIFICATION BETWEEN LAYER_2 AND LAYER_2⁻

As shown in Figure 2, Layer_2 and Layer_2⁻ communicate via a set of event/command mailboxes. Layer_2⁻, which is also referred to as the Low-Level Driver (LLD), provides a common interface between LAPD and Layer_1 for all the devices that might provide the physical connection. The Physical Layer has the task of transmitting and receiving packets for LAPD and also providing status information reflecting the connection. In addition, it has the task of providing the necessary interface between the physical device and the Layer_1 Management Entity. The details of the Low-Level Driver (LLD) interface specification with Layer_2 and the Layer_1 Management Entity is described in a separate document. This document describes the packet reception and transmission services of this driver as seen from LAPD.

4.1 Layer_2 to Layer_2⁻ Interface

The Layer_2 to Layer_2⁻ communication is provided via the L2_LLD_Command_Mail_Box, which follows the general format for the mailboxes as described in Section 3.2.

In order to transmit a packet, LAPD must provide the following primitive:

Description	Parameter	Byte
Transmit Buffer	Buffer RAM Address	4
	Packet Length	2
	Buffer Length	2

This command will instruct the Low-Level Driver to transmit one or more packets of the size equal to the "Packet Length." The packet starts at the memory address equal to the "Buffer RAM Address." The packets are assumed to be contiguous in RAM. The total length (in bytes) of all packets in the transmit buffer is equal to the "Buffer Length" parameter.

All 2-byte and 4-byte parameters follow the format described in Section 3.2.

The allowed values for the Receipt_Acknowledge_Field are:

Code	Definition
00	Set by LLD to acknowledge successful receipt
01	Set by LLD to indicate illegal command code
02	Set by LLD to indicate illegal parameters
FF	Set by LAPD to initialize the field

4.2 Layer_2⁻ to Layer_2 Interface

The Layer_2⁻ to Layer_2 communication is provided via the LLD_L2_Event_Mail_Box, which follows the general format for the mailboxes as described in Section 3.2 with one exception.

The "Buffer Transmitted Without Error" and "Packet Received Without Error" are the primitives that are generally used for packet reception/transmission services for LAPD. The format for these events are:

Description	Parameters	Byte
Buffer Transmitted Without Error	None	
Packet Received Without Error	Packet Address	4
	Packet Length	2
	Buffer Reference Number	2
Buffer Transmit Exception	None	

The "Buffer Transmitter Without Error" is a response from the LLD to LAPD indicating the successful transmission of a packet that was initiated by a previous "Transmit Buffer" command from LAPD.

The "Packet Received Without Error" is an indication from the LLD to LAPD identifying the reception of a packet. The "Packet Address" parameter describes the starting address of the packet, the "Packet Length" parameter identifies the length of the received packet, and the "Buffer Reference Number" is returned to the higher layer for its use in de-allocating the received buffer after processing the packet.

All 2- and 4-byte parameters follow the format described in Section 3.2.

The "Buffer Transmit Exception" is an indication from the LLD to LAPD identifying that an error has occurred in a previously initiated "Transmit Buffer" command from LAPD.

The allowed values for the Receipt_Acknowledge_Field are:

Code	Definition
FF	Set by LLD to initialize the field
00	Set by LAPD to indicate successful receipt
11	Set by LAPD to indicate error in the code

DETAILED FUNCTIONAL SPECIFICATION

In this section, the DL and MDL primitives and their associated command/event codes and parameters are described in more depth.

The DL primitives are used for communication interface between Layers 2 and 3, whereas the MDL primitives are used for communication interface between Layer_2 and the Layer_2 Management Entity.

In the case of 2-byte-long parameters, the least significant byte is loaded first. For example, if LENGTH is a 2-byte-long parameter, the parameter field will have the following format:

Octet	1	Low order	8 bits of LENGTH (LSB)
	2	High order	8 bits of LENGTH (MSB)

In the case of address parameters, the parameter will occupy 4 bytes and will have the format as described below:

Octet	1	Low order	8 bits of the OFFSET
	2	High order	8 bits of the OFFSET
	3	Low order	8 bits of the SEGMENT
	4	High order	8 bits of the SEGMENT

Each primitive description contains the following information:

PRIMITIVE:	The name of the primitive
CODE:	Command/Event code representing the primitive
MAILBOX:	Mailbox used for this primitive
INPUT:	Input parameters required for this primitive
OUTPUT:	Output parameters returned (if any)
RECEIPT CODE:	The allowed receipt codes
DESCRIPTION:	The function provided by this primitive
NOTES:	Special considerations or characteristics related for this primitive

5.1 Layer_3 to Layer_2 Interface

The detailed functional specifications for the primitives directed from Layer_3 to Layer_2 are described in this section.

PRIMITIVE: DL_ESTABLISH_REQUEST

CODE: DL_EST_RQ

MAILBOX: L3_L2_COMMAND

INPUTS:

SAPI	1 byte
CES	1 byte

OUTPUTS: None

RECEIPT CODES: FF by Layer_3
00 by LAPD to indicate successful reception
11 by LAPD to indicate error in the command

DESCRIPTION: This primitive is used to request establishing multiple frame operation.

NOTES:

PRIMITIVE: DL_RELEASE_REQUEST

CODE: DL_REL_RQ

MAILBOX: L3_L2_COMMAND

INPUTS:

SAPI	1 byte
CES	1 byte

OUTPUTS: None

RECEIPT CODES: FF by Layer_3
00 by LAPD to indicate successful reception
11 by LAPD to indicate error in the command

DESCRIPTION: This primitive is used to terminate a previously established multiple frame operation.

NOTES:

PRIMITIVE: DL_DATA_REQUEST

CODE: DL_DA_RQ

MAILBOX: L3_L2_COMMAND

INPUTS:

SAPI	1 byte
CES	1 byte
I_Message_Address	4 byte
I_Message_Length	2 byte
Reference_Number	2 byte

OUTPUTS:

Available_Slots	1 byte

RECEIPT CODES: FF by Layer_3
00 by LAPD to indicate successful receipt
11 by LAPD to indicate error in the command

DESCRIPTION: This primitive is used to pass Layer_3 messages to the LAPD using acknowledged operation.

NOTES:

PRIMITIVE: DL_UNIT_DATA_REQUEST

CODE: DL_U_DA_RQ

MAILBOX: L3_L2_COMMAND

INPUTS:

SAPI	1 byte
CES	1 byte
UI_Message_Address	4 bytes
UI_Message_Length	2 bytes
Reference_Number	2 bytes

OUTPUTS: None

RECEIPT CODES: FF by Layer_3
00 by LAPD to indicate successful reception
11 by LAPD to indicate error in the command

DESCRIPTION: This primitive is used by the Layer_3 to transmit a message to its peer using unacknowledged operation.

NOTES:

5.2 Layer_2 to Layer_3 Interface

The detailed functional specifications for the primitives directed from Layer_2 to Layer_3 are described in this section.

PRIMITIVE: DL_ESTABLISH_INDICATION

CODE: DL_EST_IN

MAILBOX: L2_L3_EVENT

INPUTS:

SAPI	1 byte
CES	1 byte

OUTPUTS: None

RECEIPT CODES: FF by LAPD
00 by Layer_3 to indicate successful receipt
11 by Layer_3 to indicate error

DESCRIPTION: This primitive is used to indicate the outcome of the procedure for establishing multiple frame operation.

NOTES:

PRIMITIVE: DL_RELEASE_INDICATION

CODE: DL_REL_IN

MAILBOX: L2_L3_EVENT

INPUTS:

SAPI	1 byte
CES	1 byte

OUTPUTS: None

RECEIPT CODES: FF by LAPD
00 by Layer_3 to indicate successful receipt
11 by Layer_3 to indicate error in the event

DESCRIPTION: This primitive is used to indicate the termination of a previously established multiple frame operation session.

NOTES:

PRIMITIVE: DL_UNIT_DATA_INDICATION

CODE: DL_U_DA_IN

MAILBOX: L2_L3_EVENT

INPUTS:

SAPI	1 byte
CES	1 byte
UI_Message_Address	4 bytes
UI_Message_Length	2 bytes
Reference_Number	2 bytes

OUTPUTS: None

RECEIPT CODES: FF by LAPD
00 by Layer_3 to indicate successful receipt
11 by Layer_3 to indicate error in the event

DESCRIPTION: This primitive is used to indicate the receipt of a Layer_3 message from a peer using the unacknowledged operation.

NOTES:

PRIMITIVE: DL_DATA_INDICATION

CODE: DL_DA_IN

MAILBOX: L2_L3_EVENT

INPUTS:

SAPI	1 byte
CES	1 byte
I_Message_Address	4 bytes
I_Message_Length	2 bytes
Refno	2 bytes

OUTPUTS: None

RECEIPT CODES: FF by LAPD
00 by Layer_3 to indicate successful receipt
11 by Layer_3 to indicate error in the event

DESCRIPTION: This primitive is used to indicate the receipt of a message from a peer Layer_3 using the acknowledged transfer operation.

NOTES:

PRIMITIVE: DL_ESTABLISH_CONFIRM

CODE: DL_EST_CON

MAILBOX: L2_L3_EVENT

INPUTS:

SAPI	1 byte
CES	1 byte

OUTPUTS: None

RECEIPT CODES: FF by LAPD
00 by Layer_3 to indicate successful receipt
11 by Layer_3 to indicate error in the event

DESCRIPTION: This primitive is used to confirm the establishment of multiple frame operation.

NOTES:

PRIMITIVE: DL_RELEASE_CONFIRM

CODE: DL_REL_CON

MAILBOX: L2_L3_EVENT

INPUTS:

SAPI	1 byte
CES	1 byte

OUTPUTS: None

RECEIPT CODES: FF by LAPD
00 by Layer_3 to indicate successful receipt
11 by Layer_3 to indicate error in the event

DESCRIPTION: This primitive is used to confirm the termination of a previously established multiple frame operation session.

NOTES:

5.3 Management Entity to Layer_2 Interface

The detailed functional specifications for the primitives directed from the Layer_2 Management Entity to Layer_2 is described in this section.

PRIMITIVE: MDL_ASSIGN_REQUEST

CODE: MDL_AS_RQ

MAILBOX: ME_L2_COMMAND

INPUTS:

TEI	1 byte
CES	1 byte

OUTPUTS: None

RECEIPT CODES: FF by ME
00 by LAPD to indicate successful receipt
11 by LAPD to indicate error in command

DESCRIPTION: This primitive is used to assign the TEI value. The LAPD may then enter the TEI_ASSIGN state.

NOTES:

PRIMITIVE: MDL_REMOVE_REQUEST

CODE: MDL_R_RQ

MAILBOX: ME_L2_COMMAND

INPUTS:

TEI	1 byte

OUTPUTS: None

RECEIPT CODES: FF by ME
00 by LAPD to indicate successful receipt
11 by LAPD to indicate error in command

DESCRIPTION: This primitive is used to remove the association of the assigned TEI value with the specified connection endpoint.

NOTES:

PRIMITIVE: MDL_UNIT_DATA_REQUEST

CODE: MDL_U_DA_RQ

MAILBOX: ME_L2_COMMAND

INPUTS:

UI_Message_Address	4 bytes
UI_Message_Length	2 bytes
Reference_Number	2 bytes

OUTPUTS: None

RECEIPT CODES: FF by ME
00 by LAPD to indicate successful receipt
11 by LAPD to indicate error in command

DESCRIPTION: This primitive is used to pass information messages from the Management Entity to the LAPD, which should then transmit these messages using unacknowledged operation with the broadcast address.

NOTES:

PRIMITIVE: MDL_XID_REQUEST

CODE: MDL_X_RQ

MAILBOX: ME_L2_COMMAND

INPUTS:

SAPI	1 byte
CES	1 byte
XID_Message_Address	4 bytes
XID_Message_Length	2 bytes
Reference_Number	2 bytes

OUTPUTS: None

RECEIPT CODES: FF by ME
00 by LAPD to indicate successful receipt
11 by LAPD to indicate error in command

DESCRIPTION: This primitive is used to transmit an XID command frame.

NOTES:

5.4 Layer_2 to Management Entity Interface

The detailed functional specifications for the primitives directed from Layer_2 to the Layer_2 Management Entity are described in this section.

PRIMITIVE: MDL_XID_RESPONSE

CODE: MDL_X_RS

MAILBOX: L2_ME_COMMAND

INPUTS:

SAPI	1 byte
CES	1 byte
XID_Message_Address	4 bytes
XID_Message_Length	2 bytes
Reference_Number	2 bytes

OUTPUTS: None

RECEIPT CODES: FF by ME
00 by LAPD to indicate successful receipt
11 by LAPD to indicate error in command

DESCRIPTION: This primitive is used to transmit an XID response frame.

NOTES:

PRIMITIVE: MDL_ERROR_RESPONSE

CODE: MDL_ER_RS

MAILBOX: L2_ME_COMMAND

INPUTS:

SAPI	1 byte
CES	1 byte
Error_Code	1 byte

OUTPUTS: None

RECEIPT CODES: FF by ME
00 by LAPD to indicate successful receipt
11 by LAPD to indicate error in command

DESCRIPTION: This primitive is used to indicate an error response.

NOTES:

PRIMITIVE: MDL_ASSIGN_INDICATION

CODE: MDL_AS_IN

MAILBOX: L2_ME_EVENT

INPUTS: CES 1 byte

OUTPUTS: None

RECEIPT CODES: FF by LAPD
00 by ME to indicate successful receipt
11 by ME to indicate error in event

DESCRIPTION: This primitive is used to indicate that a TEI needs to be assigned.

NOTES:

PRIMITIVE: MDL_UNIT_DATA_INDICATION

CODE: MDL_U_DA_IN

MAILBOX: L2_ME_EVENT

INPUTS:

UI_Message_Address	4 bytes
UI_Message_Length	2 bytes
Reference_Number	2 bytes

OUTPUTS: None

RECEIPT CODES: FF by LAPD
00 by ME to indicate successful receipt
11 by ME to indicate error in event

DESCRIPTION: This primitive is used to indicate to the Management Entity that an information frame has been received via unacknowledged operation.

NOTES:

PRIMITIVE: MDL_XID_INDICATION

CODE: MDL_X_IN

MAILBOX: L2_ME_EVENT

INPUTS:

SAPI	1 byte
CES	1 byte
XID_Message_Address	4 bytes
XID_Message_Length	2 bytes
Reference_Number	2 bytes

OUTPUTS: None

RECEIPT CODES: FF by LAPD
00 by ME to indicate successful receipt
11 by ME to indicate error in event

DESCRIPTION: This primitive is used to indicate that an XID command frame has been received.

NOTES:

PRIMITIVE: MDL_XID_CONFIRM

CODE: MDL_X_CON

MAILBOX: L2_ME_EVENT

INPUTS:

SAPI	1 byte
CES	1 byte
XID_Message_Address	4 bytes
XID_Message_Length	2 bytes
Reference_Number	2 bytes

OUTPUTS: None

RECEIPT CODES: FF by LAPD
00 by ME to indicate successful receipt
11 by ME to indicate error in event

DESCRIPTION: This primitive is used to confirm that a successful XID response has been received.

NOTES:

PRIMITIVE: MDL_ERROR_INDICATION

CODE: MDL_ER_IN

MAILBOX: L2_ME_EVENT

INPUTS:

SAPI	1 byte
CES	1 byte
Error_Code	1 byte

OUTPUTS: None

RECEIPT CODES: FF by LAPD
00 by ME to indicate successful receipt
11 by ME to indicate error in event

DESCRIPTION: This primitive is used to notify the Management Entity that an error has occurred.

NOTES: Error Codes are defined in Q.921 Specifications Appendix II.

PRIMITIVE: MDL_TIMER_SERVICE_INDICATION

CODE: MDL_TM_IN

MAILBOX: L2_ME_EVENT

INPUTS:

ppppimm 1 byte
where ppppp indicates a priority number
(0 = low, 31 = high)
i = 0 indicates that an interrupt should be generated upon timer time-out
i = 1 indicates that a global variable should be set upon timer time-out
mm = 00 to start the timer
= 01 to stop the timer
= 10 to restart the timer
= 11 reserved

Reference_Number	2 bytes
Time_Out value	2 bytes
Pointer to global variable	4 bytes

OUTPUTS: None

RECEIPT CODES: FF by LAPD
00 by ME to indicate successful receipt
11 by ME to indicate error in event

DESCRIPTION: This primitive is used to start, restart, or stop a timer.

NOTES: When starting or restarting the timer, the time_out value must be specified. Also, the pointer to global variable parameter must be set only if "i" is set to "1."

When stopping the timer, only the first two parameters are required.

PRIMITIVE: MDL_TIMER_SERVICE_RESPONSE

CODE: MDL_TM_RS

MAILBOX: L2_ME_COMMAND

INPUTS: Reference_Number 2 bytes

OUTPUTS: None

RECEIPT CODES: FF by ME
00 by LAPD to indicate successful receipt
11 by LAPD to indicate error in event

DESCRIPTION: This primitive is used to notify the LAPD of a timer time-out.

NOTES:

PRIMITIVE: MDL_BUFFER_SERVICE_INDICATION

CODE: MDL_BF_IN

MAILBOX: L2_ME_EVENT

INPUTS: ———m 1 byte
where m = 0 indicates allocation
m = 1 indicates de-allocation
Size in Byte 2 bytes

OUTPUTS: Starting Address 4 bytes
Reference_Number 2 bytes

RECEIPT CODES: FF by LAPD
00 by ME to indicate request is granted
11 by ME to indicate request is denied

DESCRIPTION: This primitive is used to request a buffer from the Management Entity.

NOTES: The output parameters are returned only if the request is granted.

Advanced Micro Devices

AmLink3
Interface Reference Guide

Publication #	Rev.	Amendment
10812	A	/0

Issue Date: 1988

TABLE OF CONTENTS

1 Introduction

The software modules described in this document provide complete ISDN capabilities up to layer 4 according to OSI's 7 layer architecture. The details of the functions provided by these layers are described in the following sections.

1.1 OSI Software Structure

Figure 1 illustrates the relationship between these OSI layers. The physical layer capabilities are provided by AMD's S-interface devices such as (Am79C30A) which not only provide the proper 'S' interface connections but also perform the HDLC frame structuring functions of the D-channel as well. For the B-channel the Am79C401 device can be used to provide the HDLC framing and protocol support.

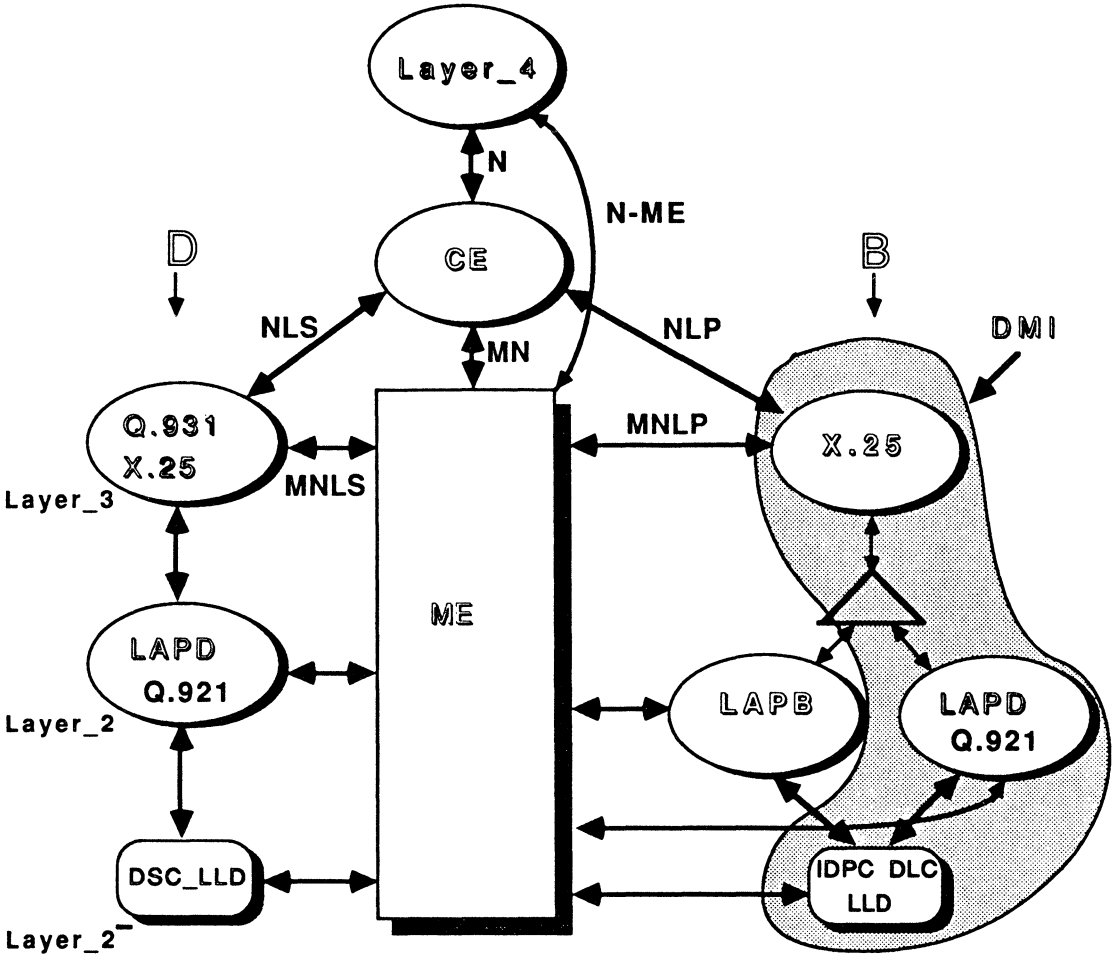
The Lower Layer Device drivers (LLD's) hide the device specific characteristics from the higher layers and provide a universal data transfer mechanism for layers 2 and above. The device specific functions such as the management of the Main Audio Processor (MAP) unit in Am79C30A are handled between the Management Entity (ME) and the specific LLD.

Layer_2 or the Data Link Layer (DLL) is in charge of providing an error free transport mechanism for the Layer_3. Layer_2 can be thought of as a messenger which carries sequenced envelopes for the Layer_3 entities across the interface. The Layer_3 simply places the message targeted for its peer Layer_3 inside this envelope and hands it down to the Layer_2. It is then the responsibility of the Layer_2 to carry this information to the other side. If by any chance the envelope is lost in the system, the Layer_2 has the responsibility of locating and re-delivering it transparent to all higher layers. Layer_2 protocols used to carry D-channel signalling procedures are described in the CCITT Q.921 Recommendation. The D-channel can also be used to carry user data other than signalling information for which the Q.921 procedures (LAPD) are utilized as well.

The Layer_3 on the D_channel is in charge of establishing, maintaining, and terminating network connections (i.e. connecting a path across the network) between 2 terminating ends. These signalling and routing procedures are described in CCITT Q.931 Recommendation. Layer_3 also performs a set of functions on data units (or packets), such as segmenting and blocking, sequencing, flow control, expedited transfer, and reset and offers the corresponding services to layer 4 which are described in the X.25 protocols. Layer_3 signalling procedures are switch dependent and each existing ISDN switch [AT&T-5ESS, Northern Telecom DMS100, etc.] has slightly different signalling requirements.

The Digital Multiplexing Interface (DMI) as described by AT&T uses the Layer_3 packet layer protocols together with Layer_2's LAPD procedures to transfer Layer_4 messages over the B-channel. In packet switch (PS) 5ESS applications the B-channel data is transferred using X.25 on Layer_3 and LAPB on Layer_2. In circuit switch (CS) 5ESS applications the B-channel data can be transferred via any user defined protocol such as X.25, DMI, etc.

The Coordinating Entity is in charge of receiving the commands/information from Layer_4 and transferring it through proper B or D channels to the appropriate peer.



CE- Coordinating Entity
ME- Management Entity
NLS- Network Layer Signalling Entity

NLP- Network Layer Packet Handling Entity

△- Implementation Option

Figure 1. AmLink3™ Software Block Diagram

The Management Entity is in charge of providing and maintaining harmonious inter-layer communications. The Management Entity is shared between all OSI layers and provides services such as: buffer/memory management, timer services, message exchange environment between various tasks, etc.

1.2 AMD's ISDN Terminal Coprocessor Board™ (ITCB™)

The ITC Board is designed to be used as a development environment for ISDN application software. Therefore, the software modules described in this document are ported onto this board. The hardware objectives of the ITC Board are:

To provide an evaluation vehicle for AMD's ISDN chip set
To provide a development and certification environment for
software modules designed for OSI architecture.
To become a demonstration tool for AMD's total ISDN solution

The following sections describe the portion of this software which could reside on the ITCB and the interface between the ITCB resident software and the surrounding system.

1.3 Software Boundary between PC and ITCB

Figure 2 shows the software blocks and the boundary between the software residing inside the PC and the software running on the ITCB. Starting from the higher layer downward the structure resembles the general architecture described in Figure 1 with the exception of the PC to ITCB communication interface blocks.

1.3.1 Application Layer

This layer resides in the PC and provides such services as voice calls and data calls.

1.3.1.1 Voice Calls

This module runs on the PC and allows the user to place a voice call. The program prompts the user to enter the number to be dialed then it proceeds with the call initialization procedures and informs the user of the status of the call. On the receiving side this layer allows the user to receive calls and alerts the user of the incoming calls.

1.3.1.2 Data Calls

This module runs on the PC and allows the user to send and receive PC files and interactive keyboard entered data using the X.25 packet layer protocol.

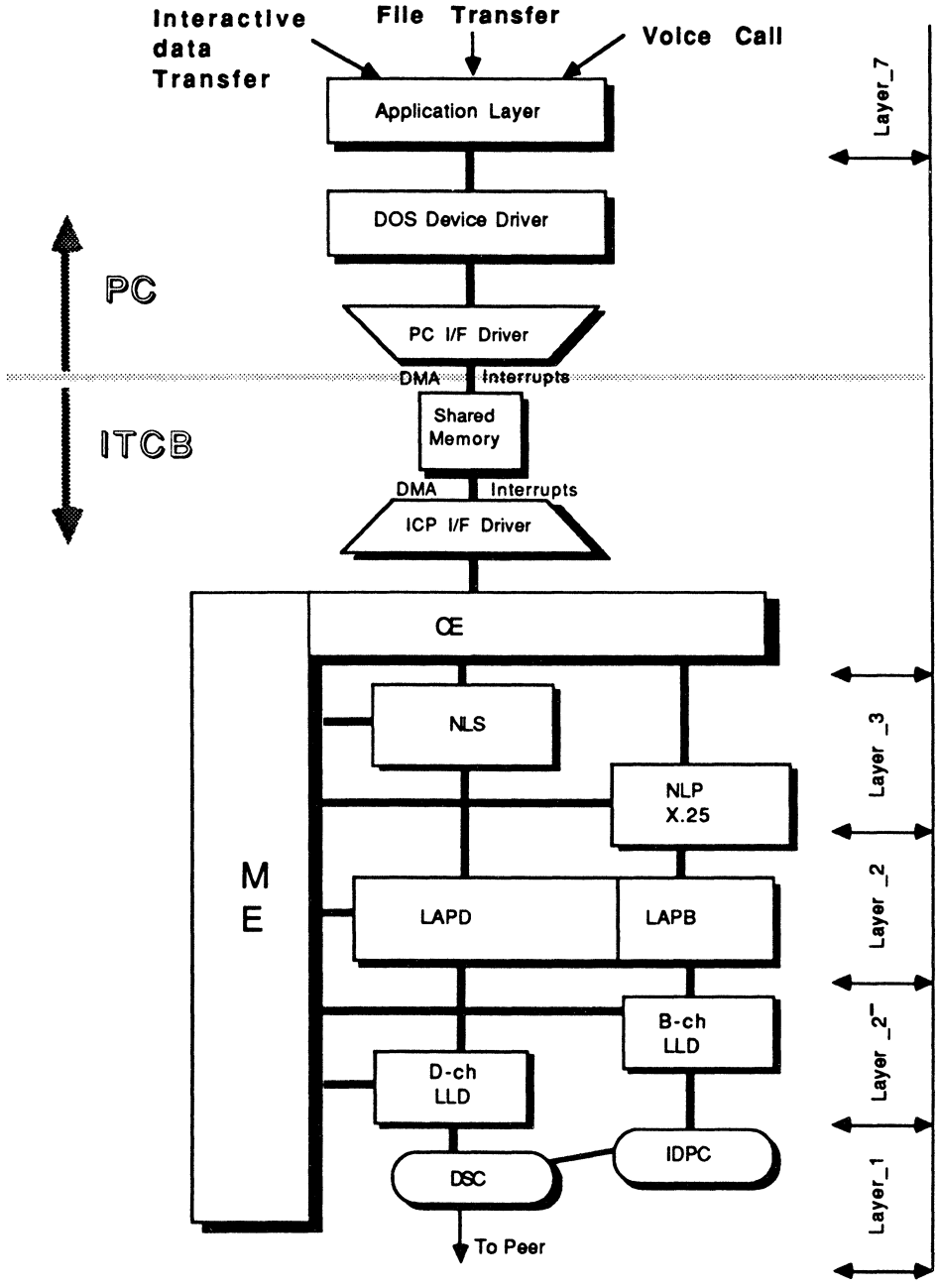


Figure 2. PC/ITCB Software Block Diagram

1.3.2 Dos Device Driver[1]

This optional module provides an interface between user defined tasks (programs) and the functions provided by the lower layers. It allows the user supplied application software packages to utilize DOS function calls to establish voice and data calls and X.25 data transfers.

1.3.3 PC to ITCB Interface Driver

This drives resides in the PC and has the responsibility of transferring data between the PC and the ITCB. This communication is achieved via shared memory space between the PC and The ITC Board, which is enhanced by Dual Port Memory Control (DPMC) capabilities of the Am79C401 Integrated Data Protocol Controller™ (IDPC™).

This interface handles all I/O port addressing, interface interrupts, and data transfers.

1.3.4 ITCB to PC Interface Driver

This driver resides in the ITCB and has the responsibility of transferring data between the ITCB and PC. This communication is achieved via shared memory space between ITCB and PC. This interface handles all interrupt and data transfer functions.

1.3.5 Coordinating Entity

The Coordinating Entity (CE) as described previously receives the Layer_4 commands and information and transfers them through proper B or D channels to the appropriate peer entity.

1.3.6 Management Entity

The Management Entity (ME) provides the memory management, timer support, inter-processor communication, and control functions for all layers residing on the ITCB. In this implementation a real time executive called "pSOS", which is provided by Software Components Group, Inc. is used to aid the design and implementation of this module. Interactions with pSOS and the rest of this system are handled through a minimum set of procedures which reduce the operating system dependency of the overall code.

1.3.7 Network Layer Signalling Entity (NLS)

As described previously this entity, which resides on the ITC Board provides the Layer_3 D-channel signalling protocol described in the CCITT Q.931 specifications.

---

1 Dos Device Driver is optional and will be available at a later date.

1.3.8 Network Layer Packet Handling Entity (NLP)

This entity provides the X.25 packet layer protocol capabilities for data transfer operations. The entity can then use the LAP protocol to transfer this information to the LLD's.

1.3.9 LAPD/LAPB (AmLink™)

This module provides the Layer_2 data link functions as specified in CCITT's Q.921 specifications.

1.3.10 D-Channel LLD

These device drivers provide the D-channel data transfer capability for the Layer_2 modules. As described previously the device specific functions are resolved between the LLD and the Management Entity.

1.3.11 B-Channel LLD

These device drivers provide the B-Channel data transfer capability. They too use the Management Entity to resolve device specific services.

2 Demonstration Capability

AmLink3™ along with the ITC Board serves as the demonstration vehicle for complete ISDN terminal equipment up to layer 4. The application layer at the OSI top level will provide voice and data call setups and tear-downs. For data calls both bidirectional file transfer and interactive data modes are supported.

3 Future Expansions and Options

3.1 Switch Specific Layers

The NLS module is initially designed for AT&T's 5ESS switch; DMS-100 and other switches will be added shortly.

3.2 Rate Adaptation Schemes[2]

Rate adaptation software will be added to support various packet oriented rate adaptation schemes such as V.120.

---

2 V.120 Rate Adaptation is optional and will be provided in future.

4 Software Development Environment

All software modules are coded in 'C' and 8088/86 assembly languages.

4.1 PC-Based Software

The Microsoft 'C' compiler version 4.0 and the Microsoft MASM assembler version 5.0 were used for code development. The Codeview Microsoft debugging tool was also used to aid the debugging activities.

All windowing activities are generated by direct calls to Bios. Since no windowing packages are used for this purpose, users can alter and compile the code without the need to purchase additional windowing tools.

4.2 ITCB Based Software

Most of the software modules running on the ITCB were initially developed and tested on the PC using the Am79B300 board. These modules where then downloaded onto the ITCB.

Unfortunately, the Microsoft 'C' compiler assumes PC based Bios/DOS environment for the target code; therefore several changes were required to transfer the compiled object codes to absolute code. To do this we used the Link & Locate tool developed by Systems and Software, Inc. which was recommended by Microsoft's staff.

For initial debugging and testing purposes the 80188 emulator of the HP64000 systems was used. The pROBE debugger (developed by Software Components Group, Inc.) along with the ITCB control program were used to assist debugging throughout porting activities onto the ITC Board.

5 Program Down Loading and Execution

The ITC Board contains a simple down loader which together with the PC resident control program provide the mechanism to download and execute the final code. The downloading and execution of the code is automated by the Layer_4 user interface module of AmLink3 software package.

6 Certification and Verification

It is our intention to verify proper operation of the AmLink3 and ITCB with most ISDN switches.

Currently the operation of the Q.931 portion of the AmLink3 package is verified on AT&T's 5ESS switch.

1 Introduction

This document provides an overview of AMD IDSN Layer_3 software architecture.

1.1 ISDN Services Supported

This first generation AMD Basic Rate Interface software architecture is designed to take advantage of the basic ISDN services provided by the AT&T Network Systems 5ESS Switch. Most of these services are generic to ISDN and will be supported by other switch manufacturers. Where differences among switch manufacturers are anticipated, the architecture attempts to localize those differences in a few software modules, maximizing the amount of software that is switch-independent.

The major ISDN services supported are:

1. Basic Voice Call Service.

2. Packet-switched, nailed-up X.25 (ISDN D-Channel) Data Call Service.

3. Packet-switched, nailed-up X.25 (ISDN B-Channel) Data Call Service.

("Nailed-up" means "no circuit-switched (out-of-band) call setup required.")

This architecture is designed to simultaneously support a voice call on one B-Channel and multiple data calls on the D-Channel and the other B-Channel. Simultaneous data calls on both B-Channels are not supported.

In general, services and features of the 5ESS Basic Rate Interface that are designated as optional or supplementary are not implemented in this first generation architecture. However, the "hooks" are present in the architecture to make implementing these options an "add-on" rather than a rewrite. It is anticipated that many of these services and features will be implemented in future software generations.

2 Hardware Base

The hardware base upon which this software architecture is overlaid is an IBM-compatible PC, running the MS-DOS operating system Version 2.0 or later, with a standard configuration of microprocessor ("PC Central Processing Unit" or "PC CPU"), system RAM, and floppy and/or hard disk storage. Added to this hardware base is a plug-in ISDN Terminal Coprocessor Board (ITCB) with the following major components:

80188 or 80186 processor ("Communications Co-processor")

D-Channel interface chip (e.g., Am79C30A Digital Subscriber Controller™ (DSC))

B-Channel interface chip (e.g., Am79C401 Integrated Data Protocol Controller (IDPC))

"Shared RAM" which both the PC CPU and the Communications Co-processor can access via the IDPC Dual-Port Timing Controller. This RAM is used for communications between the two processors.

"Local RAM" which only the Communications Co-processor can access. This RAM is used for Co-processor data (e.g., stack and global variables) and executable code storage.

Minimal ROM at absolute memory location hex "FFFF0" to allow for downloading of ITC board executable code during board initialization. Later implementations may place some or all of the Co-processor executable code in ROM.

3 Overview of the Software Layers

Figure 1 shows the layering (partitioning) of the software including references to the International Standards Organization (ISO) seven-layered Open Systems Interconnection (OSI) Model.

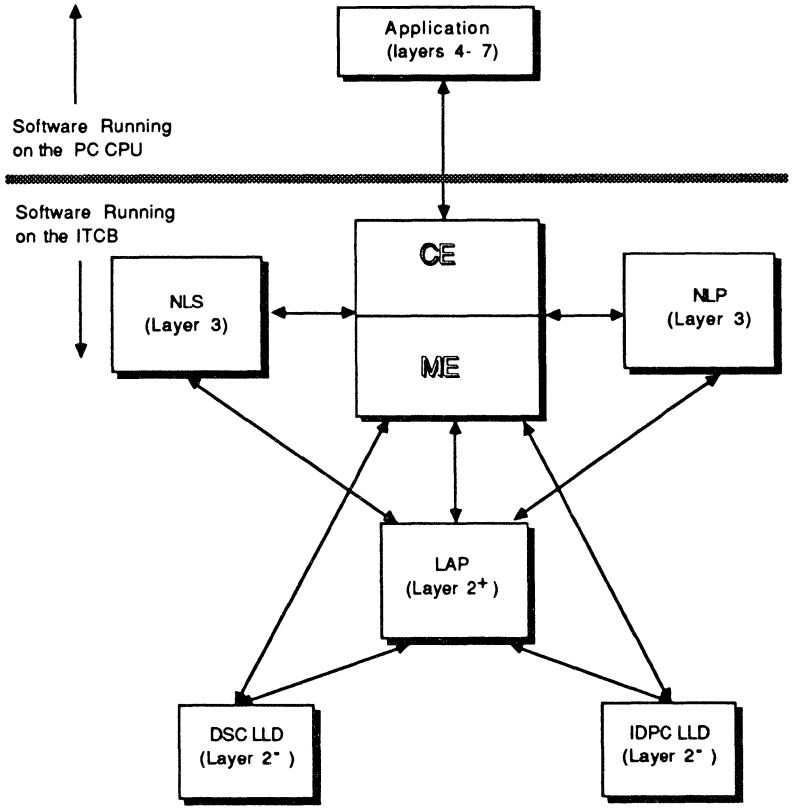


Figure 1. Software Layer Entities and Inter-layer Interfaces

The following basic mechanisms characterize each interface between adjacent software layers:

1. SET OF WELL-DEFINED PRIMITIVES (also referred to as commands or messages) and associated parameters that each layer uses to communicate with an adjacent layer.

2. PRIMITIVE TRANSFER POINT by means of which a layer sends a primitive command code (and associated parameters) to be executed by an adjacent layer. All Primitive Transfer Points are uni-directional. Primitive Transfer Points are of two basic types: "acknowledged" and "unacknowledged." In the case of an "acknowledged" transfer point, the layer that is sending a primitive does not send another primitive through the transfer point until the layer receiving the primitive has explicitly acknowledged receipt of the first primitive. In contrast, a layer that is sending a primitive via an "unacknowledged" transfer point may send another primitive at any time without knowing or caring when the adjacent layer receives primitives sent earlier.

In this architecture, an "acknowledged" transfer point is referred to as a "mailbox" and an "unacknowledged" transfer point is referred to as a "message exchange." A mailbox is further characterized by the fact that both the entity sending the primitive and the entity receiving the primitive can read and write the contents of the mailbox. In the case of a message exchange, the sending entity can only write to the transfer point; the receiving entity can only read from the transfer point.

In general, a mailbox is used in a Primitive Transfer Point if the sending entity wishes for the receiving entity to process the primitive before the sending entity sends another primitive. A message exchange is used where this restriction is not necessary. In addition, the message exchange mechanism provided by many real time operating systems allow multiple primitives to be queued between layers. Later sections of this document describe the type of Primitive Transfer Point used at each inter-layer interface.

The following sections outline the functions of each of the layers depicted in Figure 1, from the top down.

3.1 Software Running on the PC CPU

THE APPLICATION LAYER (Layer 7) interacts with the PC user to get network connection information (e.g., a telephone number to dial) and exchanges user data with the other end of an ISDN network connection. This data may take the form of a disk file or data typed interactively on the PC keyboard.

THE DOS DEVICE DRIVER (Layers 4-6)[1] converts system call requests from the Application Layer (e.g., OPEN, WRITE, IOCTL) into primitives for the Coordinating Entity to execute.

---

1 In the initial implementation the functions of the OSI Layers 4 to 7 are combined in an application module referred to as "Layer_4".

Similarly, the DOS Device Driver receives primitives from the Coordinating Entity (e.g., containing user data from the far end of the telephone connection) for transfer to the Application via a READ or IOCTL system call.

3.2 Software Running on the Communications Co-Processor

THE COORDINATING ENTITY (CE) coordinates the activities of the Network Layer Signalling (NLS) entity and the Network Layer Packet (NLP) entity. The NLS entity is responsible for Layer 3 out-of-band voice and data call setup while the NLP entity is responsible for Layer 3 in-band call setup and transfer of user data. For example, for a circuit-switched data call, the CE ensures that the call has been setup (other end has answered the phone) via messaging on the ISDN D-Channel before allowing user data to be transferred on the circuit-switched ISDN B-Channel.

THE MANAGEMENT ENTITY (ME) provides the glue necessary to make all the other layers running on the Communications Co-processor play together smoothly. Among the functions performed by the ME are to:

* Handle protocol management tasks such as Q.921 TEI negotiation.

* Control Low-Level-Driver functions which are not handled by Layer 2 such as Am79C30A DSC tone generation to the voice handset (e.g., dial tone).

* Provide services such as timer services and buffer allocation.

In this architecture, the interface between Coordinating Entity and Management Entity consists of direct subroutine calls instead of the more formal message exchange or mailbox Primitive Transfer Point mechanisms. Formal primitive transfer between the CE and the ME is eliminated in order to enhance performance as allowed by the ISO and CCITT standards. To all layers outside the CE and the ME, the two entities appear as separate and distinct.

REAL TIME OPERATING SYSTEM In this software architecture, real-time operating system dependence is minimized by localizing all interactions with the real-time operating system in a few "operating system interface functions." (Subroutines are referred to as "functions" in the C programming language.) In the first AMD implementation of this architecture, real-time operating system services are provided by the Software Components Group, Inc., pSOS real-time operating system.

LAYER 3 NETWORK LAYER SIGNALLING (NLS) entity exchanges out-of-band signalling messages on the D-Channel with the ISDN network to setup and teardown voice and data calls. These messages contain information such as the number to dial and information about call progress such as Dial Tone, Ringing or Busy, and Answered. The CCITT Q.931 specification describes the "core" signalling protocol used by the NLS. Each switch manufacturer adds its own optional signalling features to the Q.931 "core."

LAYER 3 NETWORK LAYER PACKET (NLP) entity provides reliable, error-controlled transfer of user data to and from the peer, independent of the Layer 2 protocol in use. (Depending on the type of logical connection used, the NLP entity may have to perform in-band call setup before user data transfer can begin.)

This architecture subdivides the NLP entity into Call Control and Data Phase components. Logical call connections that use the full CCITT X.25 Packet Layer Protocol use both the Call Control and Data Phase NLP components.

LAYER 2 DATA LINK LAYER (DLL) entity provides a reliable, error-controlled data transport service for carrying Layer 3 messages to and from the next link of a physical channel. This architecture supports CCITT Q.921 (LAPD) and X.25 Link Layer (LAPB) protocols at layer 2. The DLL entity interacts with the D-Channel Low-Level Driver and the B-Channel Low-Level Driver to transmit and receive frames on the ISDN D-Channel and B-Channel respectively.

D-CHANNEL LOW-LEVEL DRIVER provides hardware independence to the Data Link Layer on the D-Channel by handling all details of programming the D-Channel interface chip (e.g., Am79C30A DSC) hardware registers.

B-CHANNEL LOW-LEVEL DRIVER provides hardware independence to the Data Link Layer on the B-Channel by handling all details of programming the B-Channel interface chip (e.g., Am79C401 IDPC Data Link Controller) hardware registers.

4 Inter-Layer Interfaces

4.1 Applications to MS-DOS Driver Interface[2]

The DOS Device Driver supplied by AMD provides a consistent application interface for high-level application programs. The interface allows use of either DOS function calls or those provided by high-level languages, such as 'C'. A library module is also supplied which facilitates software coding and interface at the application layer. However, the DOS device driver, itself, is independent of the actual coding method used for the application. The host system must be running DOS Release 2.0 or later. The AMD ISDN DOS device driver is installed by placing the line:

device=amd_bri.sys

in the "config.sys" file on the host system.

Application commands to the device driver can be placed in three categories: initialization commands, call control commands, and data transfer commands. These commands may be accessed by simple C function calls: "open()" (3DH), "close()" (3EH), "read()" (3FH), "write()" (40H), and "ioctl()" (44H, AL = 02H or 03H). Each command may also be accesed via DOS software interrupt 21H using the hexadecimal AH register function call code listed in parentheses following the C language function call name.

An application opens up a channel via the "open()" (3DH) call and closes it via the "close()" (3EH) call. "Ioctl()" (44H, AL = 02H or 03H) is used to provide call control facilities to the application. Call control commands include application access and control of PAD attributes of the local (X.3) or remote (X.29) system. The only call that the application must make with "ioctl()" is to CONNECT for establishing a call. "Close()" will automatically do a DISCONNECT on the call, if the application has not already done so via "ioctl()." "Read()" (3FH) and "write()" (40H) are used for data reception and transmission.

Initialization commands fall into two types: initialization of the ISDN code on the ITC board, and initialization of DOS device driver asynchronous event handling code within the application.

The first type of initialization is done by a call to the DOS device driver library function "init_icp()." This function is called by an ITC board initialization program (".exe" module) which may be invoked in the "autoexec.bat" file at DOS boot time or executed interactively without rebooting DOS if the user has changed the "amd_bri.cup" ("Customer User Profile") file.

---

2 Dos Device Driver is optional and will be available at a later date

The second type of initialization is performed within any application program which wishes to receive asynchronous events (e.g., receipt of expedited data) when the application calls the DOS device driver library function "bri_init()." "Bri_init()" (or the appropriate "ioctl()" (44H) call as described in the "ISDN DOS Device Driver Interface Reference Guide") must be used once in the application program to initialize asynchronous event handling within the application.

Two files are used during ITC board initialization: "amd_bri.cod" which contains the ISDN code module, and "amd_bri.cup" ("Customer User Profile") which contains information pertaining to services subscribed to by the user from their ISDN network supplier.

Call control commands are done via the "ioctl()" (44H, AL = 02H or 03H) function call. A user-named 'C' structure of type "bri_struct" is used to pass necessary information. Commands which the application can access via "ioctl()" (but which it does not have to use, except for CONNECT) are:

N_CONN	to CONNECT to a far end-point
N_DISC	to DISCONNECT a far end-point
N_FLOW	to enable/disable FLOW control of data from far end-point
N_EXPED	to send EXPEDITED user data (dependent on CONNECT parameters)
N_RESET	to RESET the data link
N_Q_DATA	to send/receive non-PAD (X.29) data
N_INIT	to INITIALIZE the ISDN software, if not already done
N_RESTART	to INITIALIZE the ISDN software, unconditionally

In addition, commands are available via "ioctl()" to set local or remote PAD parameters (if PAD is in use for this call):

P_SET	to SET PAD parameters (local or remote)
P_RD	to READ PAD parameters (local or remote)
P_S_RD	to SET and READ PAD parameters (local or remote)
P_PARM	to obtain PAD parameters (in response to P_RD or P_S_RD)

Any response from a PAD command (from P_RD or P_S_RD) is transmitted by calling the application-supplied handler routine (as given in the "bri_init()" function call). If the application has **not** supplied a handler routine, PAD responses can not be communicated to the application.

"Bri_init()" has one parameter. The single argument of the "bri_init()" function is the address of the application-supplied function (the "BRI_INT handling function") that the DOS device driver is to call when a BRI_INT event occurs. This is how an application receives all call control and data progress messages and any response from a PAD command (i.e., from P_RD or P_S_RD).

If the application does not want to supply an asynchronous event handler, then the application may pass NULL (0H) as the "bri_init()" argument. An application **must** supply an asynchronous event handling function if:

the application is expecting to receive expedited data
or
the application is explicitly using X.29 commands
or
the application wants direct control over call control responses.

A 'C' global structure, "BRI_strct" of type "bri_struct," contains valid information when the DOS device driver calls the asynchronous event handling function. Another 'C' global buffer, "BRI_data," contains with any incoming expedited data. Upon exit from the asynchronous event handling function, the "BRI_strct" struct will be marked invalid by the DOS device driver by setting the "command" struct element to zero. In general, it is wise for any application programmer to avoid usage of function calls beginning with bri_ or global data which begins with BRI_.

4.2 Inter-processor Interface (Layer 4/Coordinating Entity Interface)

This interface is the only interface in this architecture that crosses the boundary between the PC CPU and the ITC board Communications Co-processor. In the PC CPU to Comm Co-processor

direction (Layer 4 to CE), the interface consists of an "acknowledged" mailbox called "L4CE" and a hardware interrupt. See Figure 2. In the opposite direction, the interface consists of an "acknowledged" mailbox called "CEL4" and a hardware interrupt. See Figure 3.

There is a basic asymmetry in this interface. The Layer 4 entity generally waits for the CE to finish processing a primitive sent to the CE before Layer 4 continues processing. However, in the other direction, the CE does not wait for the Layer 4 to finish processing a primitive before continuing with other CE tasks. This asymmetry is a result of the fact that the Layer 4 is running as part of the single-tasking DOS operating system while the CE is running under the multi-tasking pSOS operating system.

The primitives used between these layers are:

N_CONN_RQ	Request from LAYER 4 TO CE to establish connection.
N_CONN_RS	Response from LAYER 4 TO CE to indicate connection acknowledged by Layer 4.
N_DISC_RQ	Request from LAYER 4 TO CE to disconnect connection.
N_RES_RQ	Request from LAYER 4 TO CE to RESET connection.
N_RES_RS	Response from LAYER 4 TO CE to indicate reset acknowledged.

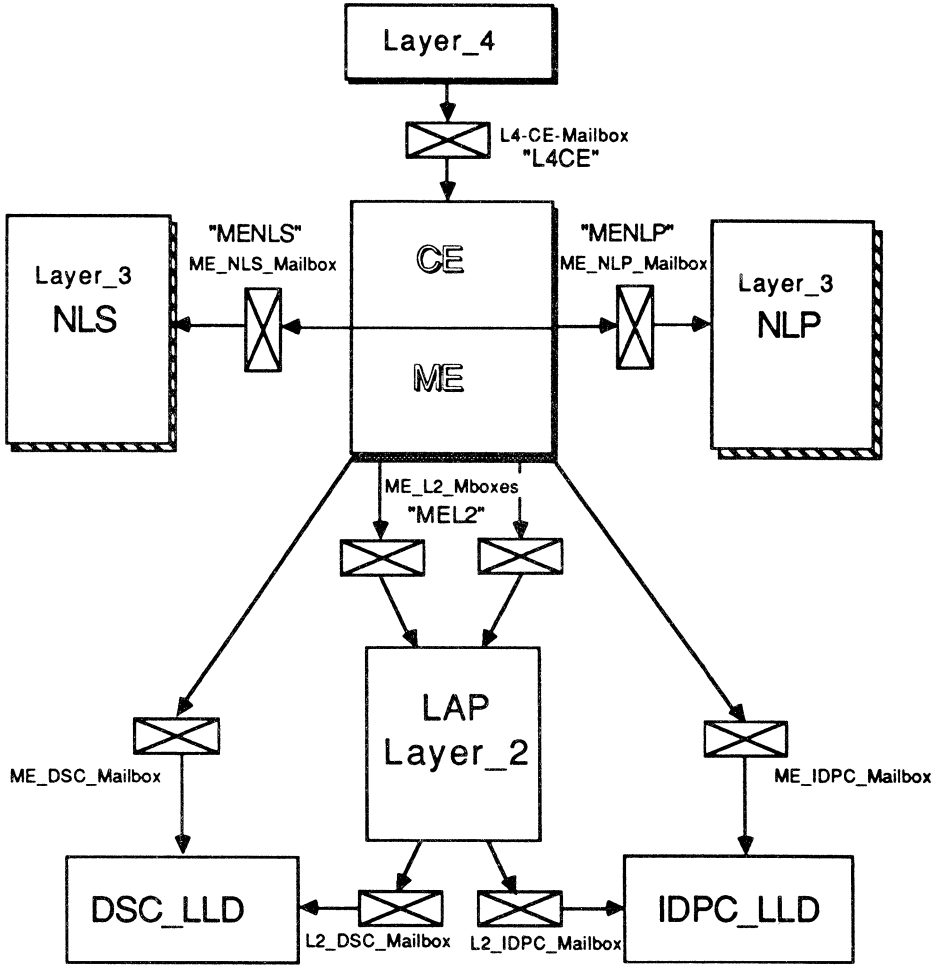


Figure 2. Higher-layer-to-Lower-layer communication mailboxes

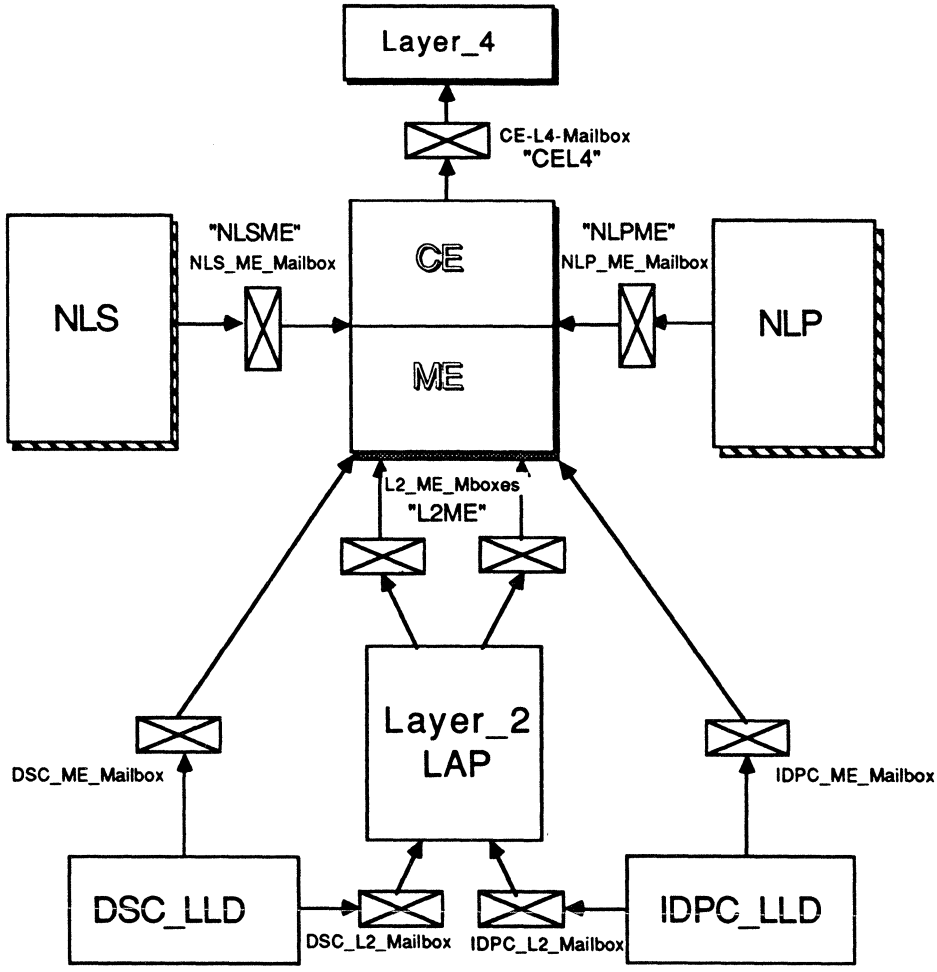


Figure 3. Lower-layer-to-Higher-layer communication mailboxes

N_DATA_RQ	Request from LAYER 4 TO CE to transmit data.
N_DACK_RQ	Request from LAYER 4 TO CE to acknowledge receipt of data received from the CE.
N_EXP_RQ	Request from LAYER 4 TO CE to send expedited data.
N_STAT_RQ	Request from LAYER 4 TO CE to inform the CE of special conditions.
N_START_RQ	Request from LAYER 4 TO CE to start normal operation after ITC board initialization.
N_CONN_IN	Indication from CE TO LAYER 4 of an incoming connection.
N_CONN_CF	Confirmation from CE TO LAYER 4 of established connection.
N_DISC_IN	Indication from CE TO LAYER 4 of disconnection.
N_RES_IN	Indication from CE TO LAYER 4 of a data link reset.
N_RES_CF	Confirmation from CE TO LAYER 4 of requested data link reset.
N_DATA_IN	Indication from CE TO LAYER 4 of incoming data.
N_DACK_IN	Indication from CE TO LAYER 4 of far-end acknowledgement of receipt of data sent earlier by the local end.
N_EXP_IN	Indication from CE TO LAYER 4 of received expedited data.
N_STAT_IN	Indication from CE TO LAYER 4 of special status conditions.

The "Coordinating Entity Interface Reference Guide" contains a detailed description of this interface.

4.3 Intra-board Interfaces

This section describes the nature of the interfaces between software layer entities running on the ITC board.

4.3.1 ITC Real-Time Operating System

The ITC board communications protocol layer entities - the NLS, NLP, and DLL entities - are specifically designed to be operating system independent. However, these entities must share the

Communication Co-processor in an efficient fashion. Furthermore, these protocol entities must exchange primitives in an orderly manner. The AmLink3 software architecture is designed to work with any real-time operating system which provides these basic services.

The first AmLink3 software implementation uses the "pSOS-86 Real-Time, Multi-processing Operating System Kernal" ("pSOS") from Software Components Group, Inc. to provide the basic processor sharing and primitive message transfer functions. Operating system independence is maintained for the protocol entities by means of funneling all interactions with the real time operating system (OS) through carefully isolated operating system interface functions (subroutines). OS function calls are performed only in these operating system interface functions. Thus, if another operating system is to be substituted for pSOS, only the operating system interface functions need to be altered -- without affecting the protocol entities' code.

In the sections that follow, pSOS terminology is used to provide concrete illustrations. The reader should keep in mind that the functions provided by pSOS are provided in very similar form by other real time operating systems.

Only a subset of pSOS's capabilities are utilized in order to minimize operating system dependence in this implementation. The primary pSOS services used are its "messages exchanges" and related process scheduling functions.

pSOS message exchanges are used primarily in this implementation to provide one-way primitive transfer FIFO's between layer entities. A secondary use of message exchanges is for an occasional semaphore to allocate a physical resource (e.g., Coordinating Entity-to-Layer 4 Mailbox) which is being contended for by several process entities.

In general, a pSOS process is created for each inter-layer pSOS message exchange FIFO used. Other inter-layer interactions (e.g., Data Link Layer-to-Management Entity) take place via direct function (subroutine) calls rather than through pSOS.

Since the number of processes is fixed (i.e., based on the static number of inter-layer message exchanges rather than the dynamic number of active network call connections), performance of the system is not significantly degraded due to process scheduling overhead as new logical connections are added.

Both pSOS message exchanges and processes can have an associated four character mnemonic. The convention that is used in this document is to give the same name to both a given FIFO message exchange and the process that is driven by that message exchange. Example: The NLS-to-Coordinating Entity FIFO message exchange and the CE sub-entity pSOS process that unloads that FIFO are both named "S3CE." The corresponding NLP-to-CE FIFO and process are named "P3CE."

4.3.2 Coordinating Entity/Network Layer -- Packet (Layer 3 NLP) Interface

This interface consists of an "unacknowledged" pSOS message exchange in each direction.

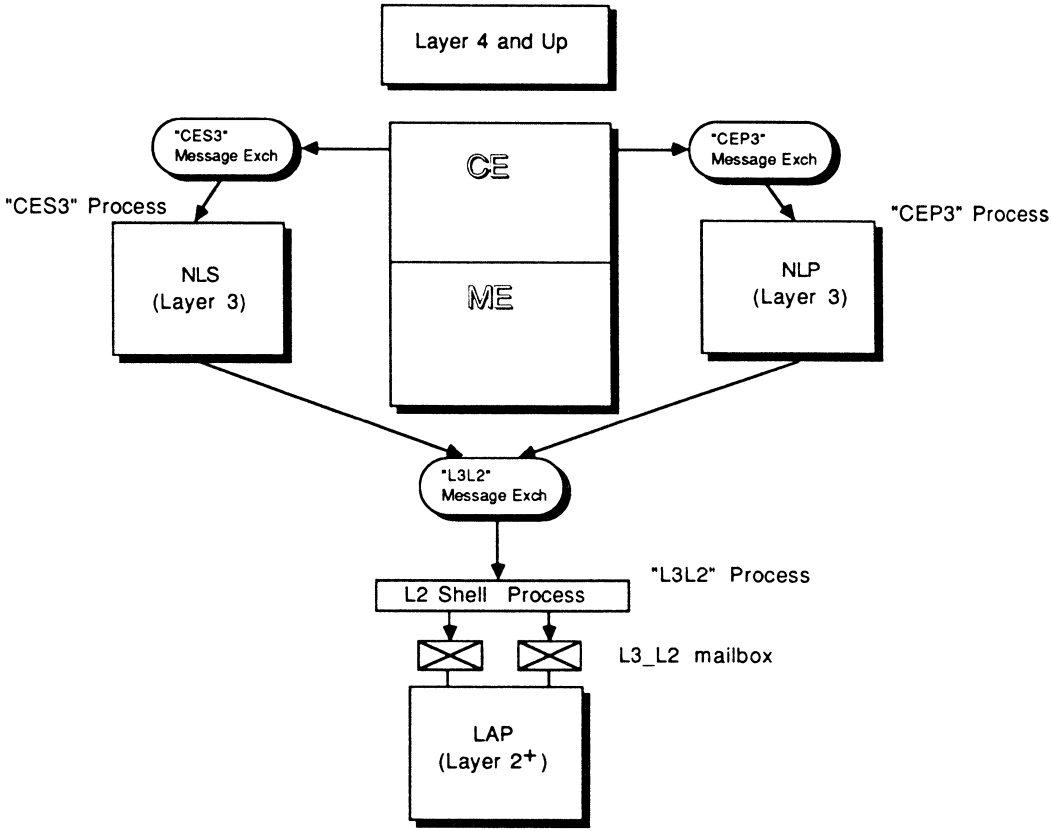


Figure 4. Higher Layer to Lower Layer Message Exchange

In the CE-to-NLP direction, the CE formats a primitive message including parameters and calls the "ce_nlp_snd()" function with a pointer to the primitive message as a function argument. Function "ce_nlp_snd()" calls the "SEND_X" pSOS function to queue the primitive on the "CEP3" pSOS message exchange. See Figure 4.

The operating system independent NLP function "cep3()" processes primitives sent by the CE. Startup software that runs during ITC Board initialization creates process "CEP3" using function "cep3()" as the process's starting address without "cep3()" having any "awareness" that it is running under pSOS.

The "cep3()" NLP function calls the pSOS "REQ_X" function to get the next primitive from the CE to process. At the "REQ_X" call process "CEP3" is blocked from running by pSOS until there is a primitive in the "CEP3" message exchange. At that time, the "REQ_X" function returns to the "cep3()" function with a pointer to the primitive message as a function argument.

In the NLP-to-CE direction, the NLP formats a primitive message including parameters and calls the "nlp_ce_snd()" function with a pointer to the primitive message as a function argument. Function "nlp_ce_snd()" calls the "SEND_X" pSOS function to queue the primitive on the "P3CE" pSOS message exchange. See Figure 5.

The operating system independent CE function "p3ce()" processes primitives sent by the NLP. Startup software that runs during ITC Board initialization creates process "P3CE" using function "p3ce()" as the process's starting address without "p3ce()" having any "awareness" that it is running under pSOS.

The "p3ce()" CE function calls the pSOS "REQ_X" function to get the next primitive from the NLP to process. At the "REQ_X" call inside process "P3CE" is blocked from running by pSOS until there is a primitive in the "P3CE" message exchange. At that time, the "REQ_X" pSOS function returns to the "p3ce()" function with a pointer to the primitive message as a function argument.

A variety of commands and indications may be passed between the Coordinating Entity and the NLP. A variety of call processing primitives are necessary for in-band call setup in a X.25 network. Other calls are used for data transfer and may be sent using LAPB (B-channel) or LAPD (D-channel) depending on the physical channel being used for the connection. The primitives used between these layers are:

NLP_CONN_RQ	Request from LAYER 3 NLP TO CE to establish in-band connection.
NLP_CONN_RS	Response from LAYER 3 NLP TO CE to indicate connection confirmed.
NLP_DISC_RQ	Request from LAYER 3 NLP TO CE to disconnect in-band connection.

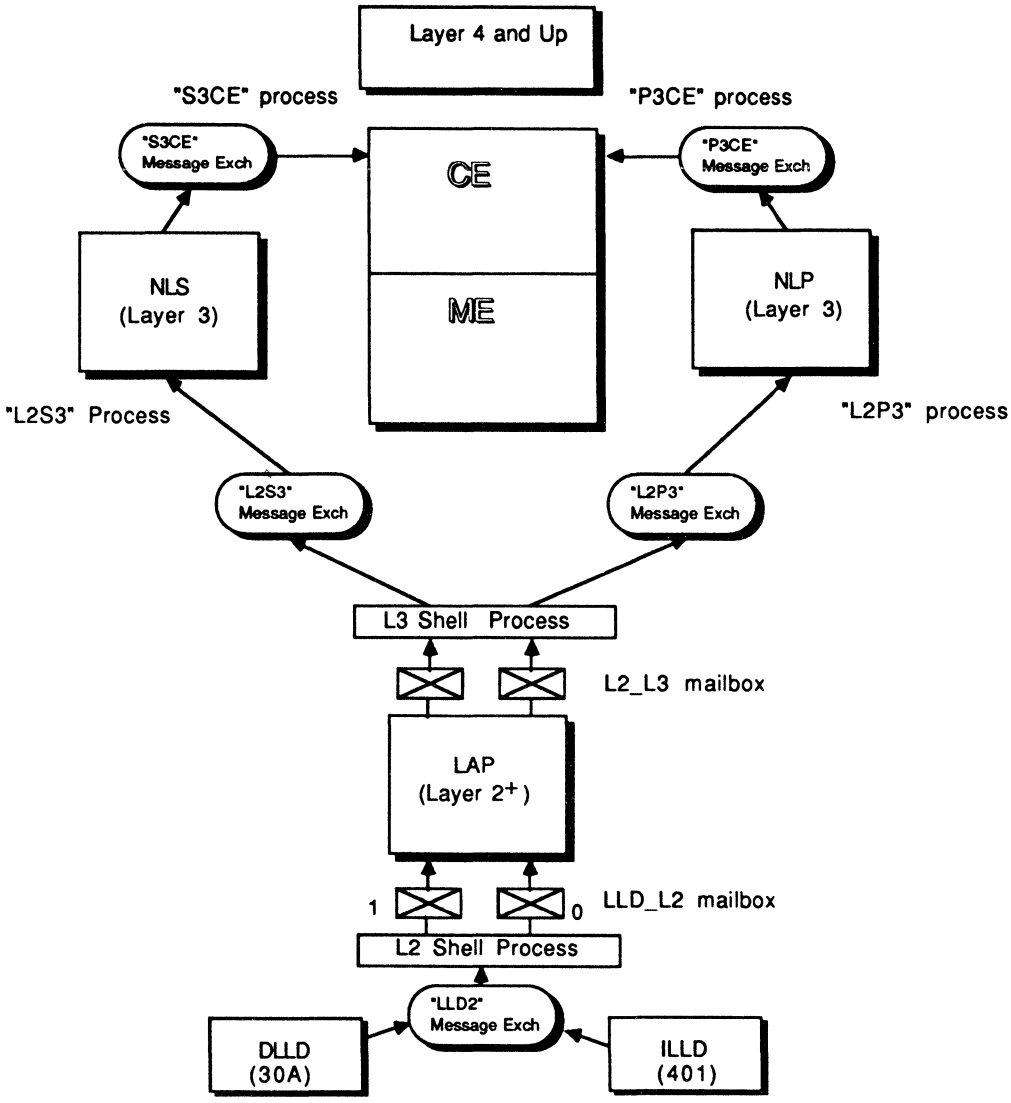


Figure 5. Lower Layer to Higher Layer Message Exchange

NLP_RES_RQ	Request from LAYER 3 NLP TO CE to RESET data link.
NLP_RES_RS	Response from LAYER 3 NLP TO CE to indicate RESET acknowledged.
NLP_DATA_RQ	Request from LAYER 3 NLP TO CE to transmit data.
NLP_DACK_RQ	Request from LAYER 3 NLP TO CE to acknowledge D-bit data.
NLP_EXP_RQ	Request from LAYER 3 NLP TO CE to send expedited data.
NLP_CONN_IN	Indication from CE TO LAYER 3 NLP of an in-band connection.
NLP_CONN_CF	Confirmation from CE TO LAYER 3 NLP of established connection.
NLP_DISC_IN	Indication from CE TO LAYER 3 NLP of in-band disconnection.
NLP_DISC_CF	Confirmation from CE TO LAYER 3 NLP of in-band disconnection
NLP_RES_IN	Indication from CE TO LAYER 3 NLP of a RESET on data link.
NLP_RES_CF	Confirmation from CE TO LAYER 3 NLP of requested RESET data link.
NLP_DATA_IN	Indication from CE TO LAYER 3 NLP of incoming data.
NLP_DACK_IN	Indication from CE TO LAYER 3 NLP of far-end window rotation.
NLP_EXP_IN	Indication from CE TO LAYER 3 NLP of received expedited data.

4.3.3 Coordinating Entity/Network Layer -- Signalling (Layer 3 NLS) Interface

The CE/NLS interface consists of an "unacknowledged" pSOS message exchange in each direction.

In the CE-to-NLS direction, the CE formats a primitive message including parameters and calls the "ce_nls_snd()" function with a pointer to the primitive message as a function argument. Function "ce_nls_snd()" calls the "SEND_X" pSOS function to queue the primitive on the "CES3" pSOS message exchange. See Figure 4.

The operating system independent NLS function "ces3()" processes primitives sent by the CE. Startup software that runs during ITC Board initialization creates process "CES3" using function "ces3()" as the process's starting address without "ces3()" having any "awareness" that it is running under pSOS.

The "ces3()" NLS function calls the pSOS "REQ_X" function to get the next primitive from the CE to process. At the "REQ_X" call process "CES3" is blocked from running by pSOS until there is a primitive in the "CES3" message exchange. At that time, the "REQ_X" function returns to the "ces3()" function with a pointer to the primitive message as a function argument.

In the NLS-to-CE direction, the NLS formats a primitive message including parameters and calls the "nls_ce_snd()" function with a pointer to the primitive message as a function argument. Function "nls_ce_snd()" calls the "SEND_X" pSOS function to queue the primitive on the "S3CE" pSOS message exchange. See Figure 5.

The operating system independent CE function "s3ce()" processes primitives sent by the NLS. Startup software that runs during ITC Board initialization creates process "S3CE" using function "s3ce()" as the process's starting address without "s3ce()" having any "awareness" that it is running under pSOS.

The "s3ce()" CE function calls function "nls_ce_rcv()" to get the next primitive from the NLS to process. "nls_ce_rcv()" calls the pSOS "REQ_X" function to get the next primitive from the "S3CE" message exchange. At the "REQ_X" call inside "nls_ce_rcv()," process "S3CE" is blocked from running by pSOS until there is a primitive in the "S3CE" message exchange. At that time, the "REQ_X" pSOS function returns to the "nls_ce_rcv()" function. The "ce_nls_rcv()" function then returns to the "s3ce()" function with a pointer to the primitive message as a function argument.

The primitives used between these layers are:

NLS_CONN_RQ	Request from CE to LAYER 3 NLS to establish connection using out-of-band signalling.
NLS_CONN_RS	Response from CE to LAYER 3 NLS to indicate connection confirmed.
NLS_DISC_RQ	Request from CE to LAYER 3 NLS to disconnect out-of-band connection.
NLS_CONN_IN	Indication from LAYER 3 NLS to CE of incoming connection using out-of-band signalling.
NLS_CONN_CF	Confirmation from LAYER 3 NLS to CE of out-of-band connection completion.
NLS_DISC_IN	Indication from LAYER 3 NLS to CE of incoming out-of-band disconnect.

NLS_DISC_CF	Confirmation from LAYER 3 NLS to CE that out-of-band disconnect is complete.

The "Network Layer Signalling Interface Reference Guide" contains a detailed description of this interface.

4.3.4 Management Entity/Layer 3 NLP Interface

The ME/NLP interface consists of an "acknowledged" mailbox in each direction. No pSOS message exchanges are involved.

In the ME-to-NLP direction, the ME loads a primitive into the mailbox pointed to by the "ME_NLP" mailbox pointer and calls the "me_nlpservice()" function entry point in the NLP entity. See Figure 2. Once the NLP has finished processing the primitive, the NLP writes the receipt code in the mailbox and returns.

In the NLP-to-ME direction, the NLP loads a primitive into the mailbox pointed to by the "NLP_ME" mailbox pointer and calls the "nlp_meservice()" function entry point in the ME. See Figure 3. Once the ME has finished processing the primitive, the ME writes the receipt code in the mailbox and returns.

A variety of commands and indications may be passed between the Management Entity and the NLP. These primitives are used for timer management and buffer allocation and deallocation. The primitives used between these layers are:

MNLP_AS_RQ	Request from ME to LAYER 3 NLP to assign LCI/SAPI.
MNLP_RM_RQ	Request from ME to LAYER 3 NLP to disassociate LCI/SAPI from conn.
MNLP_TM_RS	Response from ME to LAYER 3 NLP indicating timer expiration.
MNLP_TM_IN	Indication from LAYER 3 NLP to ME indicating timer service needed.
MNLP_BF_IN	Indication from LAYER 3 NLP to ME indicating buffer service need.
MNLP_CID_IN	Indication from LAYER 3 NLP to ME indicating that a Connection ID (ConnID) is needed.
MNLP_ER_IN	Indication from LAYER 3 NLP to ME indicating an error condition.

4.3.5 Management Entity/Layer 3 NLS Interface

The ME/NLS interface consists of an "acknowledged" mailbox in the NLS_to_ME direction. No pSOS message exchanges are involved.

In the NLS-to-ME direction, the NLS loads a primitive into the mailbox pointed to by the "NLS_ME" mailbox pointer and calls the "nls_meservice()" function entry point in the ME. See Figure 3. Once the ME has finished processing the primitive, the ME writes the receipt code in the mailbox and returns.

The primitives used between these layers are:

MNLS_CP_IN	Indication from LAYER 3 NLS to ME indicating call setup progress.
MNLS_BF_IN	Indication from LAYER 3 NLS to ME indicating buffer service need.
MNLS_CID_IN	Indication from LAYER 3 NLS to ME indicating that a Connection ID (ConnID) is needed.
MNLS_ER_IN	Indication from LAYER 3 NLS to ME indicating an error condition.

4.3.6 Layer 3/Layer 2 (NLP/LAPD, NLP/LAPB, and NLS/LAPD) Interface

The interface between the Layer 2 Data Link Layer entity (LAPB/LAPD) and Layer 3 is not symmetric. This is due to the fact that Layer 3 (NLS or NLP) does not care, with received data, what physical channel was used. In the other direction, Layer 2 (LAPB/LAPD) does not care which Layer 3 entity (NLS or NLP) is sending data to be transmitted -- only which physical channel and logical link (CES:SAPI) are to be used. (Note: The terms "CES," "SAPI,", and "CEI" are defined in CCITT Q.921.)

In both directions, the CES component of the CEI parameter found in each Layer 2/Layer 3 primitive is used to translate between physical channel and appropriate Layer 3 entity -- NLS or NLP. In this architecture, there is always a fixed relationship between a given CES and a particular physical channel. In the Layer 3 to Layer 2 direction, CES = CES_DNUPX25 (D-Channel Nailed-up X.25) and CES = CES_NLS imply D-Channel; all other CES's imply B-Channel by default. In the other direction, CES = CES_NLS implies NLS; all other CES's imply NLP.

In the Layer 3 to Layer 2 direction, the Layer 2 "front-end" is a pSOS process named "L3L2." The "L3L2" process waits (via the pSOS "REQ_X" function) for a message on the Layer 3-to-Layer 2 pSOS message exchange named "L3L2."

When process "L3L2" runs as a result of the presence of a primitive in the "L3L2" message exchange, process "L3L2" translates the CEI in the primitive to the appropriate physical channel, copies the primitive to the appropriate L3L2 mailbox based on the physical channel and invokes LAPB/LAPD via the "l3_l2service()" function to process the primitive. When the AmLINK(TM) package finishes processing the primitive and returns from "l3_l2service()", the "L3L2" process checks the mailbox receipt code and relinquishes the processor by executing the "REQ_X" pSOS function on the "L3L2" message exchange.

In the Layer 2 to Layer 3 direction, the Layer 3 "front-end" is the function "l2_l3service()." Once a primitive has been loaded into the appropriate "L2L3mbox" based on physical channel, LAPD/LAPB calls function "l2_l3service()." This function uses the CEI (CES:SAPI) parameter found in the "L2L3mbox" to determine whether the primitive should be delivered to the NLS or to the NLP entity. Function "l2_l3service()" then calls the pSOS "SEND_X" function to send the primitive to the appropriate pSOS message exchange -- "L2S3" for NLS or "L2P3" for NLP. Function "l2_l3service()" then writes the "L2L3mbox" receipt code before returning.

A number of primitives are used for information transfer between Layers 2 and 3. Note that the primitives do not differentiate between NLS and NLP, which is appropriate as, in a proper OSI implementation, layer 2 should not know what the content of data is.

DL_EST_RQ	Request from Layer 3 to establish link layer connection.
DL_REL_RQ	Request from Layer 3 to tear down link layer connection.
DL_DA_RQ	Request from Layer 3 to transmit sequenced data.
DL_U_DA_RQ	Request from Layer 3 to transmit unacknowledged data.
DL_EST_IN	Indication to Layer 3 that link layer connection is established.
DL_REL_IN	Indication to Layer 3 that link layer connection is torn down.
DL_U_DA_IN	Indication to Layer 3 of received unacknowledged data.
DL_DA_IN	Indication to Layer 3 of received data.
DL_EST_CF	Confirmation to Layer 3 that link layer connection is established.
DL_REL_CF	Confirmation to Layer 3 that link layer connection is torndown.

The "ISDN AmLINK(TM) Interface Reference Guide" contains a detailed description of this interface.

5 Inter-Layer Control Entry Points

Whenever a primitive is transferred between software layer entities or when a hardware interrupt occurs, a software entity is invoked to process the primitive or the hardware interrupt. These software entities are of three basic types:

* pSOS Processes.

* Function call entry points.

* Interrupt handlers.

This section summarizes the various software entities of these three types.

5.1 pSOS Processes

All pSOS processes run on the ITC Board and are created and started by ITC Board initialization code following ITC Board hardware reset. Each process is created with starting address equal to the operating system independent function (subroutine) listed in the "Description" column. Each process receives primitives to process from a pSOS message exchange with the same name as the "pSOS Process Name."

pSOS Process Name	Description
CEP3	Layer 3 Network Layer Packet (NLP) function "cep3()" which receives and processes primitives sent by the Coordinating Entity (CE).
CES3	Layer 3 Network Layer Signalling (NLS) function "ces3()" which receives and processes primitives sent by the Coordinating Entity (CE).
P3CE	Coordinating Entity (CE) function "p3ce()" which receives and processes primitives sent by the Layer 3 Network Layer Packet (NLP) entity.
S3CE	Coordinating Entity (CE) function "s3ce()" which receives and processes primitives sent by the Layer 3 Network Layer Signalling (NLS) entity.
L3L2	Layer 2 function "l3l2()" which receives and processes primitives sent by both the Layer 3 Network Layer Signalling (NLS) and Layer 3 Network Layer Packet (NLP) entities.
L2P3	Layer 3 Network Layer Packet (NLP) function "l2p3()" which receives and processes primitives sent by Layer 2.

L2S3	Layer 3 Network Layer Signalling (NLS) function "l2s3()" which receives and processes primitives sent by Layer 2.
DLL2	Layer 2 function "dll2.c" which receives and processes primitives sent by the Low Level Drivers.

5.2 Function Call Entry Points

Function call entry points involve direct function (subroutine) calls between software entities. All "service" function entry points run on the ITC Board.

Function Name	Description
open(), close(), read(), write(), ioctl()	PC CPU DOS Device Driver functions called by the user application running on the PC CPU to establish and teardown calls and transfer user data.
user-supplied, user-named AEH	PC CPU application "Asynchronous Event Handler" function which processes asynchronous events (e.g., receipt of expedited data or far end disconnect) when called by the DOS device driver. The DOS device driver is notified of the starting address of the AEH function via an application call of the DOS device driver "bri_init()" library function.
nlp_meservice()	Management Entity (ME) function which the Layer 3 Network Layer Packet (NLP) entity calls to process an NLP-to-ME primitive.
nls_meservice()	Management Entity (ME) function which the Layer 3 Network Layer Signalling (NLS) entity calls to process an NLS-to-ME primitive.
me_nlpservice()	Layer 3 Network Layer Packet (NLP) function which the Management Entity (ME) calls to process an ME-to-NLP primitive.

Function call entry points between the Management Entity and Layer 2 are described in the "ISDN AmLINK(TM) Interface Reference Guide." Function call entry points between the Management Entity and the D-Channel and B-Channel Low-Level Drivers are described in the Low-Level Driver Reference Guides for the D-Channel and B-Channel interface chips that are used in a given implementation of this architecture.

5.3 Interrupt Handlers

Interrupt handlers are invoked as a result of a hardware interrupt or 80X8X software interrupt.

Interrupt Handler Name	Description
CEL4	PC CPU DOS Device Driver hardware interrupt handler invoked when the ITC Board Coordinating Entity (CE) executes the ITC-to-PC CPU hardware interrupt.
L4CE	ITC Board Coordinating Entity (CE) hardware interrupt handler invoked when the PC CPU DOS Device Driver executes the PC CPU-to-ITC hardware interrupt.
TISR	ITC Board Management Entity (ME) hardware interrupt handler invoked when the 80188 hardware timer 2 interrupt occures.

It is beyond the scope of this document to specify the nature of the D-Channel and B-Channel chip-specific hardware interrupt handlers and other hardware dependent (e.g., hardware timer) interrupt handlers.

6 Global Architectural Features

In order to optimize performance in certain critical areas, the following conventions are adopted that span more than one inter-layer boundary:

BUFFER "HOLES"

When Layer 4 is preparing to execute an N_DATA_REQUEST or N_EXPEDITED_DATA_REQUEST, it copies user data to a shared memory buffer that has been allocated by the ITC board processor. Layer 4 copies the user data to an address that is offset by seven bytes from the start of the buffer, leaving a seven-byte "hole" at the front of the buffer. This is done to leave room for the protocol headers that Layer 3 and Layer 2 will put "in front of" the user data before it is transmitted. (Similarly, when Layer 3 allocates a buffer to contain a packet that it wishes for Layer 2 to transmit, Layer 3 leaves a four-byte "hole" for the Layer 2 header.)

This "awareness" of the "hole" by Layer 3 and Layer 2 avoids the time-consuming need for each of these layers to recopy the data to a new buffer each time a new protocol header is to be attached.

BUFFER "REFNUMS"

The ITC board Management Entity is given the responsibility for managing all dynamically allocated shared memory buffer allocation and deallocation. After a buffer is allocated, it is frequently the case that less than the full buffer is actually filled with data. Also, the starting address of data within a buffer changes as each layer processes and discards its layer protocol header before passing the buffer to the next higher layer. In order to avoid the need for each layer to keep track of the original size and starting address of a buffer, the ME provides an identifier for each buffer at allocation time that is known as a buffer Reference Number or "refnum." When it is time for a buffer to be deallocated, the entity that is finished with the buffer only needs to pass the refnum associated with the buffer to the ME instead of the original buffer size and starting address.

FLOW CONTROL

When an entity that is receiving or transmitting data is running out of buffer space, it is important that flow control be exercised as soon as possible. If a flow control message must travel through several layers of software, then flow control will be inefficient because of the overhead of each layer's processing of the message and related real-time operating system message exchange and scheduler overhead. The more inefficient the flow control, the more buffer space must be kept in reserve to store data that was sent after the throttling entity sent the flow control message but before the entity being throttled receives the flow control message. The more quickly responsive the flow control, the more efficiently the existing buffer space can be used and less often will be the need for retransmission of data discarded because of lack of buffer space.

In order to minimize the delay in exerting flow control, the following mechanisms are used in this architecture:

THROTTLING THE APPLICATION

The Coordinatity Entity/Management Entity (CEME) keeps track of how many buffers are in use. During a buffer alocate operation, when the number of buffers in use exceeds a defined constant number of buffers ("High water mark"), the CEME sends a N_STATUS_INDICATION (FLOW_ON) primitive to Layer 4 to exert flow control. This causes Layer 4 to cease sending N_DATA_REQUEST primitives until flow control is turned off.

During a buffer deallocation operation, when the number of buffers in use is less than a defined constant number of buffers ("Low water mark"), the CEME sends an N_STATUS_INDICATION (FLOW_OFF) primitive to Layer 4 to turn off flow control. This allows Layer 4 to again transfer N_DATA_REQUEST primitives.

7 Initialization

7.1 DOS Device Driver Initialization

Like any installable DOS device driver, the AMD ISDN BRI DOS Device Driver is loaded at boot time via an entry in the "config.sys" disk file. As part of the device driver load process, DOS invokes the device driver "device initialization code" to allow the device driver to prepare to receive function calls (e.g., "open()," "ioctl(),", etc.).

Once the DOS Device Driver is ready to receive the first function calls, ITC board initialization is done via a call to the DOS device driver library function "init_icp()." This function is called by an ITC board initialization program (".exe" module) which may be invoked in the "autoexec.bat" file at DOS boot time or executed interactively without rebooting DOS if the user has changed the "amd_bri.cup" ("Customer User Profile") file. The initialization does several things: downloads executable code to the ITC board, initializes configuration tables, initiates code processes on the ITC board, passes along subscriber configuration information for the ISDN link, and obtains a buffer from the ITC board for subsequent commands.

The DOS device driver will use a map table set up in a header block of downloadable code to specify locations and lengths of downloaded code. The device driver will then pass the data, and block address, to a bootstrap routine on the ITC board, with that process worrying about proper final locations. The exact structure and memory mapping needed for initialization of the code on the board is yet to be defined. The ITC bootstrap routine will then call the "main()" function in the downloaded code to initialize pSOS and the ISDN software. The ITC board initialization scenario is described in some detail in the section that follows.

7.2 ITC Board Initialization

Hardware reset causes the 80186 processor on the board to jump to location hex FFFF0. A small routine (the "ITC bootstrap routine") in ROM at that address loops checking the for a DOWNLOAD byte command from the PC CPU.

Software running on the PC CPU starts the download of the ITC software by sending the DOWNLOAD command to the ITCB via shared RAM. When the ROM routine detects the DOWNLOAD command, it interacts with the PC CPU to download the board software using shared RAM and a prearranged download protocol.

Once this step is complete, the PC CPU writes the starting address of the board software to shared RAM. The PC CPU then writes GO command to shared RAM.

When the ROM routine detects the GO command, it performs an indirect jump to the starting address of the code.

At this point, the downloaded board software initialization code starts to run. The board init code interacts with the PC CPU software via commonly agreed shared memory locations and bi-directional hardware interrupts to download pSOS and to download the "amd_bri.cup" run-time parameter file which contains such parameters as the following:

Call Setup/Maintenance TEI (x81 for Auto)
D-Channel X.25 (non-Auto) TEI
Minimum D-Channel X.25 LCI - 2-way
Maximum D-Channel X.25 LCI - 2-way
Minimum B-Channel X.25 LCI - 2-way
Maximum B-Channel X.25 LCI - 2-way
Minimum D-Channel X.25 LCI - 1-way outgoing
Maximum D-Channel X.25 LCI - 1-way outgoing
Minimum B-Channel X.25 LCI - 1-way outgoing
Maximum B-Channel X.25 LCI - 1-way outgoing
Default Service Type
Default Voice B-Channel
Default Data B-Channel
DSC LLD Init Parms (SET_CLK parm, SET_PWR parm,
MAP IPB and DLC IPB images?)
IDPC DLC LLD Init Parameters (DLC IPB image ?)
Count of Permanent Virtual Circuits (N PVC's)
PVC #0 LCI
PVC #0 Physical Channel
.
.
.
PVC #N-1 LCI
PVC #N-1 Physical Channel

At this point, the board init code:

* Calls layer entity init functions that are not part of any pSOS process startup initialization (e.g., ME internal initialization)

* Starts pSOS

* Creates all pSOS messages exchanges

* Spawns and activates all pSOS processes. Each pSOS process runs briefly (performing startup initialization duties) and then is blocked by pSOS, waiting for a message on the pSOS message exchange with the same name (e.g., "S3CE") as the process.

* Perform all D-Channel and B-Channel Low-Level Driver commands necessary to initialize the D-Channel and B-Channel interface chips' hardware.

At this point, Layer 4 sends an "N_START_REQUEST" primitive which causes the CE to:

* Allocate an ITC Board shared RAM buffer for the Layer 4 to use in future Layer 4-to-CE primitive transfers that require an indirect parameter block.

* Perform protocol-related duties such as TEI negotiation and assignment.

1 General Description

1.1 Purpose

This document describes the interface between the Coordinating Entity (CE) and the Layer 3 Network Layer Signalling (NLS) entity running on the ICP board. The purpose of this interface is to allow the orderly transfer of out-of-band call setup primitives between the CE and the NLS.

1.2 Development Environment

The Layer 3 Network Layer Signalling (NLS) module is implemented using Microsoft 'C' compiler Version 4.0 and the Microsoft Macro Assembler Version 5.0.

2 Functional Description

The major components of this interface are as follows:

* A message exchange ("unacknowledged Primitive Transfer Point") in each of two directions: CE-to-NLS and NLS-to-CE.
* A mailbox ("acknowledged Primitive Transfer Point") in each of two directions: ME-to-NLS and NLS-to-ME.
* A set of primitives (command codes) associated with each uni-directional message exchange and mailbox.
* A set of parameters associated with each primitive.

2.1 Inter-Layer Functional Interfaces

The functional interfaces between the NLS entity and adjacent layers (levels) are of two basic types: "acknowledged" and "unacknowledged" Primitive Transfer Points. At an "acknowledged" type of inter-layer interface, the entity sending a primitive waits for the receiving entity to

acknowledge receipt of the primitive before the sending entity proceeds with any further processing. In the "unacknowledged" case, when a layer entity sends a primitive, it does not wait for any acknowledgement of receipt of the primitive before proceeding. (These basic types are described in greater detail in the "AMD ISDN BRI Software Architecture Reference Guide.")

In different implementations, the physical mechanisms used to provide "acknowledged" and "unacknowledged" Primitive Transfer Points may vary. For example, the "unacknowledged" type Primitive Transfer Point may be implemented using the inter-process message queue services provided by one of the many real-time operating systems available.

The NLS entity is designed to be independent from the physical mechanisms used to transfer primitives to and from adjacent layers. All primitive transfers with other layers are accomplished via operating system-independent inter-entity function calls. If a user wishes to change the physical mechansim for sending and receiving inter-layer primitives at a particular inter-layer interface, it is only necessary to re-write the appropriate inter-entity interface functions. For the NLS, these functions are:

nls_meservice()	Called by the NLS to notify the ME that the NLS has placed a primitive to be executed in the appropriate "acknowledged" Primitive Transfer Point.
ce_nls_snd()	Called by the Coordinating Entity to send a primitive to the NLS entity via the appropriate "unacknowledged" Primitive Transfer Point.
ces3()	NLS function used to receive a primitive from the Coordinating Entity via the appropriate "unacknowledged" Primitive Transfer Point.
nls_ce_snd()	Called by the NLS to send a primitive to the Coordinating Entity via the appropriate "unacknowledged" Primitive Transfer Point.
s3ce()	Coordinating Entity function used to receive a primitive to the NLS entity via the appropriate "unacknowledged" Primitive Transfer Point.
l2_nls_snd()	Called by Layer 2 to send a primitive to the NLS entity via the appropriate "unacknowledged" Primitive Transfer Point.
l2s3()	NLS function used to receive a primitive from Layer 2 (LAPD) via the appropriate "unacknowledged" Primitive Transfer Point.
nls_l2_snd()	Called by the NLS to send a primitive to Layer 2 (LAPD) via the appropriate "unacknowledged" Primitive Transfer Point.

1312()	Layer 2 function used to receive a primitive from the NLS entity via the appropriate "unacknowledged" Primitive Transfer Point.

Note: Function names ending in "service" denote inter-entity service functions used with "acknowledged" Primitive Transfer Points. Function names ending in "_snd" and "_rcv" indicate inter-entity functions used with "unacknowledged" Primitive Transfer Points.

For the sake of brevity, an "acknowledged" Primitive Transfer Point is referred to as a "mailbox" in the rest of this document. An "unacknowledged" Primitive Transfer Point is referred to as a "message exchange."

2.2 Inter-Layer Communication

The following sections describe the primitives, and their parameters, of communications between the Layer 3 Network Layer Signalling (NLS) package and the Coordinating Entity and Management Entity. Primitives between NLS and Layer 2 (LAPB/LAPD) are described in "ISDN AmLINK Interface Reference Guide." The primitives listed here describe in detail the format of the messages to be passed, but do not specifically tie down the code to the method of passage. This is done deliberately, once again, to preserve operating system independence.

2.3 CE and NLS Communication Interface

The interface between the Coordinating Entity and the NLS entity uses message queues, as described in the inter-layer functional interfaces section. Messages between the NLS and the Coordinating Entity are concerned with the out-of-band setup and teardown of circuit-switched voice and data calls. Most messages are able to be self-contained within the size of a message block. Incoming CONNECT messages and outgoing call CONNECT requests have more information associated with them than can be contained within a message block. Therefore, they make use of an additional buffer which is used to contain parameters. This is referred to as an "indirect parameter block."

2.3.1 CE to NLS Interface

The CE-to-NLS interface primitives are communicated via the "CES3" message exchange.

The structure of a message sent by the CE to the "CES3" message exchange is as follows:

Parameter	Length (Bytes)	Message Offset (Bytes)
Reserved for Operating System	8	0
Primitive Code	1	8
Receipt Code	1	9
Data Attributes	1	10
Data Length	2	11
Data Refnum	2	13
Cause	2	15
Logical Channel ID (LCI)	2	17
Connection ID (ConnID)	1	18
Data Address	4	22

This message format is used for all primitives passing among the Layer 4, Coordinating Entity (CE), Network Layer Packet (NLP), and Network Layer Signalling (NLS) entities. Each primitive passing from the CE to the NLS uses only a few of the parameters listed above. The detailed description of each primitive indicates which parameters are used for that primitive.

The following is a C language representation of the Layer 4/CE/NLP/NLS common message structure:

```
struct l4_ce_l3_msg {
  unsigned long reserved;  /* reserved for oper sys use */
  unsigned long home_exch; /* reserved for oper sys use */
  unsigned char primcode;  /* primitive command code */
  unsigned char receipt;   /* primitive receipt code */
  unsigned char d_attrib;  /* data attributes */
  unsigned short datalen;  /* length of data buf */
  unsigned short refnum;   /* refnum of data buf */
  unsigned char cause[2];  /* cause of DISC,RESET,STATUS */
  unsigned short lci;      /* Logical Channel Number */
  unsigned char connid;    /* CONNection ID */
  char (far *dataptr);     /* ptr to prim-specific data buf */
 } ;
```

In the CE-to-NLS direction the CE formats a primitive message including parameters and calls the "ce_nls_snd()" function with a pointer to the primitive struct as a function argument. The NLS is awakened within the "ces3()" function when the message is received from the CE.

The following primitives are used for communication from the Coordinating Entity to NLS:

NLS_CONNECT_REQUEST Request for NLS to establish a logical connection.

NLS_CONNECT_RESPONSE Response indicating that CE accepts an incoming logical connection.

NLS_DISCONNECT_REQUEST Request for NLS to disconnect a logical connection.

2.3.2 NLS to CE Interface

The NLS-TO-CE interface primitives are communicated via the "S3CE" pSOS message exchange.

The structure of a message sent by the NLS to the "S3CE" message exchange is as follows:

Parameters	Length (Bytes)	Message Offset (Bytes)
Reserved for Operating System	8	0
Primitive Code	1	8
Receipt Code	1	9
Data Attributes	1	10
Data Length	2	11
Data Refnum	2	13
Cause	2	15
Logical Channel ID (LCI)	2	17
Connection ID (ConnID)	1	18
Data Address	4	22

This message format is used for all primitives passing among the Layer 4, Coordinating Entity (CE), Network Layer Packet (NLP), and Network Layer Signalling (NLS) entities. Each primitive passing from the NLS to the CE uses only a few of the parameters listed above. The detailed description of each primitive indicates which parameters are used for that primitive.

In the NLS-to-CE direction the NLS formats a primitive message including parameters and calls the "nls_ce_snd()" function with a pointer to the primitive struct as a function argument. The CE is awakened within the "s3ce()" function when the message is received from the NLS.

The following primitives are used for communication from NLS to the Coordinating Entity:

NLS_CONNECT_INDICATION Indication to CE of an incoming attempt to establish a logical connection.

NLS_CONNECT_CONFIRM Confirmation to CE of successful establishment of an outgoing logical connection.

NLS_DISCONNECT_INDICATION Indication to CE of an incoming disconnection of a logical connection.

NLS_DISCONNECT_CONFIRM Confirmation to CE of successful disconnection of a logical connection.

2.4 ME and NLS Communication Interface

The interface between the Management Entity and the NLS entity uses "acknowledged" mailboxes, as described in the inter-layer functional interfaces section. Primitives between the NLS and the Management Entity are concerned with timer and buffer allocation/deallocation and call progress (e.g., dial tone or busy) status updating.

2.4.1 NLS to ME Interface

The NLS-to-ME interface primitives are communicated via the NLS_ME_Mailbox. The NLS_ME_Mailbox is at a fixed location in RAM.

The structure of the NLS_ME_Mailbox is as follows:

	Length (Bytes)
Primitive Code	1
Receipt Code	1
Primitive Parameters	14

To initiate a primitive transfer, the NLS writes a primitive code, any associated parameters, and a hex FF receipt code in the NLS_ME_Mailbox (pointed to by the "NLS_ME" mailbox pointer) and calls the "nls_meservice()" function entry point in the the ME entity. Once the ME has finished processing the primitive, the ME writes the receipt code in the mailbox and returns.

The following primitives are used for communication from the NLS entity to the Management Entity:

MNLS_SIGNAL_INDICATION Indication from the NLS to the ME that a call progress NLS message (e.g., indicating dial tone) has arrived.

MNLS_TIMER_INDICATION Request from the NLS to the ME to start a timer for the NLS.

MNLS_BUFFER_INDICATION Request from the NLS to the ME to allocate or deallocate a buffer.

MNLS_CONNID_INDICATION Request from the NLS to the ME to allocate a Connection ID.

MNLS_ERROR_INDICATION Indication from the NLS to the ME that an NLS Error has occurred.

The valid Receipt Code values are:

Code (Hex)	Description
FF	set by NLS before calling ME
00	set by ME to indicate successful reception of the primitive
11	set by ME to indicate error -- primitive ignored

3 Detailed Functional Specifications

In this section the NLS and MNLS primitives and their associated command/event codes and parameters are described in more depth.

The NLS primitives are used for communication interface between the Coordinating Entity and the NLS. The MNLS primitives are used between the Management Entity and the NLS.

In case of 2 byte long parameters, the least significant byte is loaded first. For example if LENGTH is a 2 byte long parameter the parameter field will have the following format:

Octet 1 Low order 8 bits of LENGTH (LSB)
Octet 2 High order 8 bits of LENGTH (MSB)

In case of address parameters, the parameter will occupy four bytes and will have the format as described below:

Octet 1 Low order 8 bits of the OFFSET
Octet 2 High order 8 bits of the OFFSET
Octet 3 Low order 8 bits of the SEGMENT
Octet 4 High order 8 bits of the SEGMENT

Each primitive description contains the following information:

PRIMITIVE:	The name of the primitive
CODE:	Mnemonic code representing the primitive
INPUT:	Input parameters required for this primitive
DESCRIPTION:	The function provided by this primitive
NOTES:	Special considerations or characteristics of this primitive

In addition, MNLS primitives require two additional information sections, due to their use of "acknowledged" mailboxes:

OUTPUTS:	Any output parameters placed into the mailbox by the entity receiving the primitive, for use by the requesting entity upon acknowledgement of the primitive
RECEIPT CODES:	Codes entered into the receipt code field of the mailbox

3.1 CE to NLS Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the Coordinating Entity to the NLS entity are described in this section.

3.1.1 NLS_CONNECT_REQUEST

PRIMITIVE: NLS_CONNECT_REQUEST

CODE:	NLS_CONN_RQ	Bytes
INPUTS:	ConnID	1
	Data Addr =	
	Indirect Parameter Blk Address	4
	Data Refnum =	
	Indirect Parameter Blk Refnum	2
	Indirect Parameter Block	

	Voice/Data Indicator	1
	Service Type	1
	Reserved	1
	Channel Type	4
	Reserved	1
	Out-of-Band Called Address Length	1
	Out-of-Band Called Address Offset	1
	Out-of-Band Calling Address Length	1
	Out-of-Band Calling Address Offset	1
	Reserved	2
	User Data Block 1 Addresss Length	1
	User Data Block 1 Addresss Offset	1
	User Data Block 2 Address Length	1
	User Data Block 2 Address Offset	1
	Out-of-Band Called Addresss	variable
	Out-of-Band Calling Address	variable
	User Data Blocks 1 and 2	variable

DESCRIPTION: Request for NLS to establish a logical connection.

NOTES: VOICE/DATA INDICATOR

Voice/Data Indicator	

0	Data call
1	Voice call

SERVICE TYPE

Service Type	
0	Use init default
1	DMI Mode 3
2	Nailed-up X.25
3	X.31

3.1.2 NLS_CONNECT_RESPONSE

PRIMITIVE:	NLS_CONNECT_RESPONSE	
CODE:	NLS_CONN_RS	Bytes
INPUTS:	ConnID	1
	Data Length =	
	Optional User Data Length	2
	Data Address =	
	Optional User Data Address	4
	Refnum =	
	Optional User Data Refnum	2
DESCRIPTION:	Response to NLS indicating that CE accepts an incoming logical connection.	
NOTES:	Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Refnum and Data Address inputs are not used.	

3.1.3 NLS_DISCONNECT_REQUEST

PRIMITIVE:	NLS_DISCONNECT_REQUEST	
CODE:	NLS_DISC_RQ	Bytes
INPUTS:	ConnID	1
	Cause	2
	(only 1 byte used for Q.931 - 5ESS BRI Spec p. IV-28)	
	Data Length =	
	Optional User Data Length	2
	Data Address =	
	Optional User Data Address	4
	Data Refnum =	
	Optional User Data Refnum	2
DESCRIPTION:	Request for NLS to disconnect a logical connection.	
NOTES:	Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Refnum and Data Address inputs are not used.	

3.2 NLS to CE Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the NLS entity to the Coordinating Entity are described in this section.

3.2.1 NLS_CONNECT_INDICATION

PRIMITIVE: NLS_CONNECT_INDICATION

CODE: NLS_CONN_IN

INPUTS:		Bytes
	Data Address =	
	Indirect Parameter Blk Address	4
	Data Refnum =	
	Indirect Parameter Blk Refnum	2
	Indirect Parameter Block	

	Voice/Data Indicator	1
	Service Type	1
	Reserved	6
	Out-of-Band Called Address Length	1
	Out-of-Band Called Address Offset	1
	Out-of-Band Calling Address Length	1
	Out-of-Band Calling Address Offset	1
	Reserved	2
	User Data Block 1 Addresss Length	1
	User Data Block 1 Addresss Offset	1
	User Data Block 2 Address Length	1
	User Data Block 2 Address Offset	1
	Out-of-Band Called Address	variable
	Out-of-Band Calling Address	variable
	User Data Blocks 1 and 2	variable

DESCRIPTION: Indication to the CE of an incoming logical connection.

NOTES:

VOICE/DATA INDICATOR

Voice/Data Indicator	
0	Data call
1	Voice call

SERVICE TYPE

Service Type	
0	Use default
1	DMI Mode 3
2	Nailed-up X.25
3	X.31

3.2.2 NLS_CONNECT_CONFIRM

PRIMITIVE:	NLS_CONNECT_CONFIRM	
CODE:	NLS_CONN_CF	Bytes
INPUTS:	ConnID	1
	Data Length =	
	Optional User Data Length	2
	Data Address =	
	Optional User Data Address	4
	Data Refnum =	
	Optional User Data Buffer Refnum	2
DESCRIPTION:	Confirmation to CE of successful logical connection.	
NOTES:	Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Refnum and Data Address inputs are not used.	

3.2.3 NLS_DISCONNECT_INDICATION

PRIMITIVE:	NLS_DISCONNECT_INDICATION	
CODE:	NLS_DISC_IN	Bytes
INPUTS:	ConnID	1
	Cause	2
	(only 1 byte used for Q.931 - 5ESS BRI Spec p. IV-28)	
	Data Length =	
	Optional User Data Length	2
	Data Address =	
	Optional User Data Address	4
	Data Refnum =	
	Optional User Data Buffer Refnum	2
DESCRIPTION:	Indication to CE that a logical connection has been disconnected.	
NOTES:	Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Refnum and Data Address inputs are not used.	

3.2.4 NLS_DISCONNECT_CONFIRM

PRIMITIVE:	NLS_DISCONNECT_CONFIRM	
CODE:	NLS_DISC_CF	Bytes
INPUTS:	ConnID	1
DESCRIPTION:	Confirmation to the CE that a logical connection has been disconnected.	

3.3 NLS to ME Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the NLS entity to the Management Entity are described in this section.

3.3.1 MNLS_SIGNAL_INDICATION

PRIMITIVE:	MNLS_SIGNAL_INDICATION	
CODE:	MNLS_SIG_IN	Bytes
INPUTS:	Signal Type	1
RECEIPT CODES:	FF by NLS 00 by ME if valid primitive 11 by ME if invalid primitive	
DESCRIPTION:	Indication from the NLS to the ME that a call progress NLS message (e.g., indicating dial tone) has arrived.	
NOTES:	SIGNAL TYPE	

Signal Type	
0	No tone
1	Dial tone
2	Ringing
3	Answered
4	Busy tone
5	Reorder tone
7F	End Alert

This primitive is only executed for voice. See page IV-33 of the 5ESS BRI Spec for related information.

3.3.2 MNLS_BUFFER_SERVICE_INDICATION

PRIMITIVE:	MNLS_BUFFER_SERVICE_INDICATION	
CODE:	MNLS_BF_IN	Bytes
INPUTS:	Buffer Service Type	1
	Buf Refnum (DEALLOCATE)	2
OUTPUTS:	Buf Size (ALLOCATE)	2
	Buf Address (ALLOCATE)	4
	Buf Refnum (ALLOCATE)	2
RECEIPT CODES:	FF by NLS 00 by ME if valid primitive 11 by ME if invalid primitive	
DESCRIPTION:	Request from the NLS to the ME to allocate or deallocate a buffer.	
NOTES:	BUFFER SERVICE TYPE	

Buffer Service Type	
0	Allocate
1	Deallocate

3.3.3 MNLS_CONNID_INDICATION

PRIMITIVE:	MNLS_CONNID_INDICATION	
CODE:	MNLS_CID_IN	Bytes
INPUTS:	none	
OUTPUTS:	ConnID	1
RECEIPT CODES:	FF by NLS 00 by ME if valid primitive 11 by ME if invalid primitive	
DESCRIPTION:	Request from the NLS to the ME to allocate a Connection ID.	
NOTES:		

3.3.4 MNLS_ERROR_INDICATION

PRIMITIVE:	MNLS_ERROR_INDICATION	
CODE:	MNLS_ER_IN	Bytes
INPUTS:	NLS Error Code	1
OUTPUTS:	none	
RECEIPT CODES:	FF by NLS 00 by ME if valid primitive 11 by ME if invalid primitive	
DESCRIPTION:	Indication from the NLS to the ME that an NLS error has occurred.	
NOTES:	The NLS Error Codes are defined in AmLink3 User's Guide.	

1 General Description

1.1 Purpose

This document describes the interface between the Coordinating Entity (CE) running on the AMD ISDN Terminal Coprocessor Board (ITCB) and the Layer 4 software running on the PC CPU. The purpose of this interface is to allow the orderly transfer of information between the CE and Layer 4.

1.2 Development Environment

The Coordinating Entity module is implemented using Microsoft 'C' compiler Version 4.0 and the Microsoft Macro Assembler Version 5.0.

2 Functional Description

The major components of this interface are as follows:

* A hardware interrupt in two directions: PC CPU-to-ITC Board processor (Layer 4-to-CE) and ITC Board processor-to-PC CPU (CE-to-Layer 4)
* A mailbox in shared RAM associated with each uni-directional hardware interrupt
* A set of primitives (command codes) associated with each uni-directional mailbox
* A set of parameters associated with each primitive

Only one primitive can be outstanding between layers in each direction at any time.

2.1 Inter-Layer Functional Interfaces

In each direction, the interface between the Layer 4 entity and the Coordinating Entity (CE) is an "acknowledged" Primitive Transfer Point. At an "acknowledged" type of inter-layer interface, the entity sending a primitive waits for the receiving entity to acknowledge receipt of the primitive before the sending entity can send another primitive.

For the sake of brevity, an "acknowledged" Primitive Transfer Point is referred to as a "mailbox" in the rest of this document.

2.2 Inter-Layer Communication

The following sections describe the primitives, and their parameters, of communications between the Layer 4 entity and the Coordinating Entity.

2.3 Coordinating Entity and Layer 4 Communication Interface

The interface between the Coordinating Entity and Layer 4 uses mailboxes, as described in the inter-layer functional interfaces section. Messages between the Coordinating Entity and Layer 4 are concerned with the setup and teardown of voice and data calls and the transfer of user data. Most messages are able to be self-contained within the size of a mailbox. Incoming CONNECT messages and outgoing call CONNECT requests have more information associated with them than can be contained within a mailbox. Therefore, they make use of an additional buffer which is used to contain parameters. This is referred to as an "indirect parameter block."

2.3.1 Layer 4 to CE Interface

The Layer 4-to-CE interface primitives are communicated via the L4_CE_Mailbox. The L4_CE_Mailbox is located in RAM ("shared RAM") that can be accessed by both the main CPU and the ITC Board processor.

The structure of a message sent by Layer 4 via the L4_CE_Mailbox is as follows:

Parameters	Length (Bytes)	Message Offset (Bytes)
Reserved for Operating System	8	0
Primitive Code	1	8
Receipt Code	1	9
Data Attributes	1	10
Data Length	2	11
Data Refnum	2	13
Cause	2	15
Logical Channel ID (LCI)	2	17
Connection ID (ConnID)	1	18
Data Address	4	22

This message format is used for all primitives passing among the Layer 4, Coordinating Entity (CE), Network Layer Packet (NLP), and Network Layer Signalling (NLS) entities. Each primitive passing from Layer 4 to the CE uses only a few of the parameters listed above. The detailed description of each primitive indicates which parameters are used for that primitive.

The following is a C language representation of the Layer 4/CE/NLP/NLS common message structure:

```
struct l4_ce_l3_msg {
  unsigned long reserved; /* reserved for oper sys use */
  unsigned long home_exch; /* reserved for oper sys use */
  unsigned char primcode; /* primitive command code */
  unsigned char receipt;  /* primitive receipt code */
  unsigned char d_attrib; /* data attributes */
  unsigned short datalen; /* length of data buf */
  unsigned short refnum;  /* refnum of data buf */
  unsigned char cause[2]; /* cause of DISC,RESET,STATUS */
  unsigned short lci;     /* Logical Channel Number */
  unsigned char connid;   /* CONNection ID */
  char (far *dataptr);    /* ptr to prim-specific data buf */
 } ;
```

To initiate a primitive transfer, Layer 4 writes a primitive code, any associated parameters, and a hex FF receipt code in the L4_CE_Mailbox. The Layer 4 entity then issues the L4-to-CE inter-processor hardware interrupt and continues checking the Receipt Code for a value other than hex FF.

Upon occurence of the hardware interrupt, the CE's Layer 4-to-CE hardware interrupt handler is invoked. First, the CE checks to see if there is any cleanup work that needs to be performed as a result of completion of the last CE-to-Layer 4 primitive transaction. If a CE_L4_Mailbox primitive transaction is outstanding (i.e., internal CE flag "CEL4inuse" is set), then the CE checks the CE_L4_Mailbox Receipt Code for a value other than hex FF. If a value other than FF is in the Receipt Code, then the CE deallocates the ITC Board buffer used in the last CE-to-Layer 4 primitive transaction (if any), clears "CEL4inuse", and executes a pSOS "SEND_X" call to the "CEL4" semaphore message exchange to make the CE_L4_Mailbox available again.

The CE interrupt handler then proceeds with the Layer 4-to-CE primitive transaction. The CE checks the validity of the L4_CE_Mailbox primitive code and parameters. If mailbox contents are invalid, the CE writes hex 11 to the L4_CE_Mailbox Receipt Code and ignores the primitive. If the primitive and parameters are valid, the CE processes the primitive. When the primitive has been completely processed, the CE writes 0 to the Receipt Code. (Note: "Complete processing" will usually mean the queuing of the primitive on either the CE_NLS_Message_Exchange or the CE_NLP_Message_Exchange after a relatively small amount of processing by the CE.)

When Layer 4 detects that the Receipt Code changes from hex FF, the primitive transaction is complete and the L4_CE_Mailbox is free for another primitive.

The valid L4_CE_Mailbox Primitive Codes are:

N_CONNECT_REQUEST	Request to CE to establish a logical connection.

N_CONNECT_RESPONSE Response indicating that Layer 4 accepts an incoming logical connection.

N_DISCONNECT_REQUEST Request to CE to disconnect a logical connection.

N_DATA_REQUEST Request to CE to transmit a block of user data on a logical connection.

N_DATA_ACK_REQUEST Request to CE to acknowledge receipt of user data on a logical connection.

N_EXPED_DATA_REQUEST Request to CE to transmit a block of expedited user data on a logical connection.

N_RESET_REQUEST Request to CE to reset a logical connection.

N_RESET_RESPONSE Response indicating that Layer 4 acknowledges an incoming reset on a logical connection.

N_START_REQUEST Request to CE to start normal operation. (Sent by Layer 4 after ITC Board initialization.)

The valid Receipt Code values are:

Code (Hex)	Description
FF	set by Layer 4 before interrupting CE
00	set by CE to indicate successful reception of the primitive
11	set by CE to indicate error -- primitive ignored

2.3.2 CE to Layer 4 Interface

The CE-to-Layer 4 interface primitives are communicated via the CE_L4_Mailbox. The CE_L4_Mailbox is located in RAM ("shared RAM") that can be accessed by both the main CPU and the ITC Board processor.

The structure of a message sent by the CE via the CE_L4_Mailbox is as follows:

Parameters	Length (Bytes)	Message Offset (Bytes)
Reserved for Operating System	8	0
Primitive Code	1	8
Receipt Code	1	9
Data Attributes	1	10
Data Length	2	11
Data Refnum	2	13
Cause	2	15
Logical Channel ID (LCI)	2	17
Connection ID (ConnID)	1	18
Data Address	4	22

This message format is used for all primitives passing among the Layer 4, Coordinating Entity (CE), Network Layer Packet (NLP), and Network Layer Signalling (NLS) entities. Each primitive passing from Layer 4 to the CE uses only a few of the parameters listed above. The detailed description of each primitive indicates which parameters are used for that primitive.

For performance efficiency, the CE-to-Layer 4 primitive transfer involves both the CE-to-Layer 4 hardware interrupt AND the Layer 4-to-CE hardware interrupt. This involvement of BOTH hardware interrupts in a CE-to-Layer 4 primitive transfer is done primarily to avoid the need for the CE to continue looping on the CE_L4_Mailbox Receipt Code while Layer 4 processes the primitive. In this way, the ITC processor can be released for other tasks while the PC CPU processes the primitive.

There are three CE sub-entities that need to access the CE_L4_Mailbox: NLS-to-CE pSOS process, NLP-to-CE pSOS process, and ME-to-CE function. Before one of these entities uses the mailbox, it must gain access to the mailbox by calling function "CEL4_wt()." "CEL4_wt()" performs a pSOS "REQ_X" call on the CE_L4_Mailbox-available semaphore (pSOS message exchange) called "CEL4." (During board initialization, a pSOS "SEND_X" call was made on message exchange "CEL4" to make the mailbox available at the start.) "CEL4_wt()" returns when the mailbox is available. This can only happen when Layer 4 issues the L4-to-CE hardware interrupt. Layer 4 issues the L4-to-CE hardware interrupt in three situations:

1. Layer 4 needs to send a primitive to the CE via the L4_CE_Mailbox.

2. Layer 4 needs to notify the CE that the CE_L4_Mailbox is again available for use (e.g., parameters and data have been moved out of the CE_L4_Mailbox and any associated indirect parameter block pointed to by an address in the mailbox.)

3. Both 1. and 2.

Once the CE_L4_Mailbox is available, the CE sub-entity writes a primitive code, any associated parameters, and a hex FF Receipt Code in the CE_L4_Mailbox. The sub-entity sets a flag internal to the CE called "CEL4inuse." (This flag will be checked by the L4-to-CE hardware interrupt handler to determine whether to check the CE_L4_Mailbox receipt code to issue a pSOS "SEND_X" call on the "CEL4" semaphore to make the CE_L4_Mailbox available to the CE sub-entities again.)

The CE sub-entity then issues the CE-to-Layer 4 inter-processor hardware interrupt (by calling function "do_l4_int()") and assumes that Layer 4 got the primitive. The CE does NOT check the Receipt Code at this point.

Upon occurence of the hardware interrupt, the Layer 4 interrupt handler is invoked. Layer 4 checks the validity of the CE_L4_Mailbox primitive code and parameters. If mailbox contents are invalid, Layer 4 writes hex 11 to the L4_CE_Mailbox Receipt Code and ignores the primitive. Layer 4 issues the Layer 4-to-CE hardware interrupt to notify the CE that the primitive has been ignored.

If the primitive and parameters are valid, Layer 4 processes the primitive code and parameters. (This may involve a block move or DMA move of data from a buffer on the ITC Board to PC system RAM.) Layer 4 then writes 0 to the Receipt Code. (If Layer 4 has generated a Layer 4-to-CE primitive as a result of processing the CE-to-Layer 4 primitive, then Layer 4 also writes the new primitive to the L4_CE_Mailbox.) Layer 4 then generates the Layer 4-to-CE hardware interrupt.

When the CE's Layer 4-to-CE interrupt handler is invoked, it first checks the private CE "CEL4inuse" flag to see if a CE_L4_Mailbox primitive transaction is outstanding. If so, the CE interrupt handler checks whether the CE_L4_Mailbox Receipt Code is not equal to hex FF. If the Receipt Code is not equal to hex FF, a CE-to-Layer 4 primitive transaction is in progress and the CE's interrupt handler must "clean up" any loose ends from that transaction. To accomplish this, the CE interrupt handler deallocates the ITC Board buffer used in the last CE-to-Layer 4 primitive (if any), clears the "CEL4inuse" flag and performs a pSOS "SEND_X" call to the "CEL4" semaphore to make the CE_L4_Mailbox available again.

Having finished any clean up of an outstanding CE-to-Layer 4 transaction, the CE interrupt handler is ready to process any Layer 4-to-CE primitive transaction that is associated with the Layer 4-to-CE hardware interrupt. The presence of a pending Layer 4-to-CE primitive is indicated by a L4_CE_Mailbox Receipt Code equal to hex FF. If the L4_CE_Mailbox Receipt Code is hex FF, the Layer 4-to-CE interrupt handler then processes a Layer 4-to-CE primitive which is accompanying the signal to make the CE_L4_Mailbox available again.

The valid CE_L4_Mailbox Primitive Codes are:

N_CONNECT_INDICATION Indication to Layer 4 of an incoming attempt to establish a logical connection.

N_CONNECT_CONFIRM Confirmation to Layer 4 of successful establishment of an outgoing logical connection.

N_DISCONNECT_INDICATION Indication to Layer 4 of an incoming disconnection of a logical connection.

N_DATA_INDICATION Indication to Layer 4 of the reception of a block of user data on a logical connection.

N_DATA_ACK_INDICATION Indication to Layer 4 of receipt of acknowledgement of user data reception by the far end on a logical connection.

N_EXPED_DATA_INDICATION Indication to Layer 4 of reception of a block of expedited user data on a logical connection.

N_STATUS_INDICATION Indication to Layer 4 to act on a change in the status of a logical connection (e.g., Flow Control toggle on a logical connection).

N_RESET_INDICATION Indication to Layer 4 of reception of a reset on a logical connection.

N_RESET_CONFIRM Confirmation to Layer 4 that an outgoing reset request on a logical connection has been completed.

The valid Receipt Code values are:

Code (Hex)	Description
-----	----------

FF	set by CE before interrupting Layer 4
00	set by Layer 4 to indicate successful reception of the primitive
11	set by Layer 4 to indicate error -- primitive ignored

3 Detailed Functional Specifications

In this section the CE and Layer 4 interface primitives and their associated command/event codes and parameters are described in more depth.

The "N_" primitives are used for communication interface between the Coordinating Entity and Layer 4.

In case of 2 byte long parameters, the least significant byte is loaded first. For example if is a 2 byte long parameter the parameter field will have the following format:

Octet 1 Low order 8 bits of LENGTH (LSB)
Octet 2 High order 8 bits of LENGTH (MSB)

In case of address parameters, the parameter will occupy four bytes and will have the format as described below:

Octet 1 Low order 8 bits of the OFFSET
Octet 2 High order 8 bits of the OFFSET
Octet 3 Low order 8 bits of the SEGMENT
Octet 4 High order 8 bits of the SEGMENT

Each primitive description contains the following information:

PRIMITIVE:	The name of the primitive
CODE:	Mnemonic code representing the primitive
INPUT:	Input parameters required for this primitive
OUTPUTS:	Any output parameters placed into the mailbox by the entity receiving the primitive, for use by the requesting entity upon acknowledgement of the primitive
RECEIPT CODES:	Codes entered into the receipt code field of the mailbox
DESCRIPTION:	The function provided by this primitive
NOTES:	Special considerations or characteristics of this primitive

3.1 Layer 4 to CE Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the Layer 4 to the Coordinating Entity are described in this section.

3.1.1 N_CONNECT_REQUEST

PRIMITIVE:	N_CONNECT_REQUEST	
CODE:	N_CONN_RQ	Bytes
INPUTS:	Data Address =	
	Indirect Parameter Blk Address	4
	Indirect Parameter Block	

	Voice/Data Indicator	1
	Service Type	1
	PVC Select	1
	Channel Type	1
	In-Band Called Address Length	1
	In-Band Called Address Offset	1
	In-Band Calling Address Length	1
	In-Band Calling Address Offset	1
	Out-of-Band Called Address Length	1
	Out-of-Band Called Address Offset	1
	Out-of-Band Calling Address Length	1
	Out-of-Band Calling Address Offset	1
	Receipt Confirmation Indicator	1
	Expedited Data Indicator	1
	Reserved	2
	User Data Address Length	1
	User Data Address Offset	1
	In-Band Called Address	variable
	In-Band Calling Address	variable
	Out-of-Band Called Address	variable
	Out-of-Band Calling Address	variable
	User Data	variable
OUTPUTS:	ConnID	
1		
	Data Address	
4		
RECEIPT CODES:	FF by Layer 4	
	00 by CE if valid prim and input parms	
	11 by CE if invalid primitive or parms	
DESCRIPTION:	Request for CE to establish a logical connection.	

NOTES:

The ConnID output is the Connection Identifier that the CE allocates for all layers to use in subsequent inter-layer primitives involving this logical connection.

The Data Address output is the address of the free buffer that the CE allocates for Layer 4 to use with the next Layer 4-to-CE primitive that requires a data buffer.

VOICE/DATA INDICATOR

Voice/Data Indicator	
0	Data call
1	Voice call

SERVICE TYPE

Service Type	
0	Use init default
1	DMI Mode 3
2	Nailed-up X.25
3	X.31 (Out-of-band call setup then In-band X.25 call setup)

PVC SELECT

Hi-order bit = 1 ==> Low-order 7 bits = PVC #
Hi-order bit = 0 ==> No PVC used

Low-order 7-bits PVC to use

CHANNEL TYPE

Hi-order bit = 1 ==> Low-order
7 bits contain
Channel to use
Hi-order bit = 0 ==> Use Default
Voice or Data
Channel Type

Low-order 7-bits

0	D-Channel
1	B1 (B-Channel)
2	B2 (B-Channel)

3.1.2 N_CONNECT_RESPONSE

PRIMITIVE:	N_CONNECT_RESPONSE	
CODE:	N_CONN_RS	Bytes
INPUTS:	ConnID	1
	Data Length = Optional User Data Length	2
	Data Address = Optional User Data Address	4
OUTPUTS:	Data Address	4
RECEIPT CODES:	FF by Layer 4 00 by CE if valid primitive 11 by CE if invalid primitive	
DESCRIPTION:	Response indicating that Layer 4 accepts an incoming logical connection.	
NOTES:	Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Address input is not used and there is no Data Address output parameter. If the Data Length is non-zero, then the Data Address output is the address of the free buffer that the CE allocates for Layer 4 to use with the next Layer 4-to-CE primitive that requires a data buffer.	

3.1.3 N_DISCONNECT_REQUEST

PRIMITIVE:	N_DISCONNECT_REQUEST	
CODE:	N_DISC_RQ	
	Bytes	
INPUTS:	ConnID	1
	Cause	2
	(only 1 byte used for Q.931 - 5ESS BRI Spec p. IV-28)	
	Data Length =	
	Optional User Data Length	2
	Data Address=	
	Optional User Data Address	4
OUTPUTS:	Data Address	4
RECEIPT CODES:	FF by Layer 4 00 by CE if valid primitive 11 by CE if invalid primitive	
DESCRIPTION:	Request for CE to disconnect a logical connection.	
NOTES:	Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Address input is not used and there is no Data Address output parameter.	
	If the Data Length is non-zero, then the Data Address output is the address of next free buffer that the CE allocates for Layer 4 to use with the next Layer 4-to-CE primitive that requires a data buffer.	

3.1.4 N_DATA_REQUEST

PRIMITIVE:	N_DATA_REQUEST	
CODE:	N_DATA_RQ	Bytes
INPUTS:	ConnID	1
	Data Length	2
	Data Address	4
	Data Attributes	1
OUTPUTS:	Data Address	4
RECEIPT CODES:	FF by Layer 4 00 by CE if valid primitive 11 by CE if invalid primitive	
DESCRIPTION:	Request for CE to transmit data on a logical connection.	
NOTES:	DATA ADDRESS and DATA LENGTH User data is located at Data Address plus an offset of 7 bytes. This 7-byte "hole" is left at the "front" of the user data to allow room for the Layer 2 and Layer 3 protocol packet headers. Similarly, the Data Length parameter is equal to the actual user data length plus 7 bytes. The Data Address output is the address of the free buffer that the CE allocates for Layer 4 to use with the next Layer 4-to-CE primitive that requires a data buffer.	

DATA ATTRIBUTES

Bit	
0	=0 => not D-bit data, =1 => D-bit data
1	=0 => not M-bit data, =1 => M-bit data
2	=0 => not Q-bit data, =1 => Q-bit data

3.1.5 N_DATA_ACKNOWLEDGE_REQUEST

PRIMITIVE:	N_DATA_ACKNOWLEDGE_REQUEST	
CODE:	N_DACK_RQ	Bytes
INPUTS:	ConnID	1
OUTPUTS:	none	
RECEIPT CODES:	FF by Layer 4 00 by CE if valid primitive 11 by CE if invalid primitive	
DESCRIPTION:	Request for CE to acknowledge receipt of user data on a logical connection.	

3.1.6 N_EXPEDITED_DATA_REQUEST

PRIMITIVE:	N_EXPEDITED_DATA_REQUEST	
CODE:	N_EXP_RQ	Bytes
INPUTS:	ConnID	1
	Data Length	2
	Data Address	4
OUTPUTS:	Data Address	4
RECEIPT CODES:	FF by Layer 4 00 by CE if valid primitive 11 by CE if invalid primitive	
DESCRIPTION:	Request for CE to transmit expedited data on a logical connection.	
NOTES:	Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Address input is not used and there is no Data Address output parameter. If the Data Length is non-zero, user data is located at Data Address plus an offset of 7 bytes. This 7-byte "hole" is left at the "front" of the data to allow room for the Layer 2 and Layer 3 protocol packet headers. Similarly, the Data Length parameter is equal to the actual user data length plus 7 bytes. If the Data Length is non-zero, then the Data Address output is the address of the free buffer that the CE allocates for Layer 4 to use with the next Layer 4-to-CE primitive that requires a data buffer.	

3.1.7 N_RESET_REQUEST

PRIMITIVE:	N_RESET_REQUEST	
CODE:	N_RES_RQ	Bytes
INPUTS:	ConnID Cause/Diagnostic Code	1 2
OUTPUTS:	none	
RECEIPT CODES:	FF by Layer 4 00 by CE if valid primitive 11 by CE if invalid primitive	
DESCRIPTION:	Request for CE to reset a logical connection.	

3.1.8 N_RESET_RESPONSE

PRIMITIVE:	N_RESET_RESPONSE	
CODE:	N_RES_RS	Bytes
INPUTS:	ConnID	1
OUTPUTS:	none	
RECEIPT CODES:	FF by Layer 4 00 by CE if valid primitive 11 by CE if invalid primitive	
DESCRIPTION:	Response indicating that Layer 4 acknowledges an incoming reset on a logical connection.	

3.1.9 N_START_REQUEST

PRIMITIVE:	N_START_REQUEST	
CODE:	N_START_RQ	Bytes
INPUTS:	none	
OUTPUTS:	Data Address	4
RECEIPT CODES:	FF by Layer 4 00 by CE if valid primitive 11 by CE if invalid primitive	
DESCRIPTION:	Request to CE to start normal operation. (Sent by Layer 4 after ITC Board initialization.)	
NOTES:	The Data Address output is the address of the free buffer that the CE allocates for Layer 4 to use with the first Layer 4-to-CE primitive following power-on reset.	

3.2 CE to Layer 4 Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the Coordinating Entity to Layer 4 are described in this section.

3.2.1 N_CONNECT_INDICATION

PRIMITIVE:	N_CONNECT_INDICATION	
CODE:	N_CONN_IN	Bytes
INPUTS:	ConnID	1
	Data Address = Indirect Parameter Blk Address	4
	Indirect Parameter Block	

	Voice/Data Indicator	1
	Service Type	1
	PVC Select	1
	Channel Type	1
	In-Band Called Address Length	1
	In-Band Called Address Offset	1
	In-Band Calling Address Length	1
	In-Band Calling Address Offset	1
	Out-of-Band Called Address Length	1
	Out-of-Band Called Address Offset	1
	Out-of-Band Calling Address Length	1
	Out-of-Band Calling Address Offset	1
	Receipt Confirmation Indicator	1
	Expedited Data Indicator	1
	Reserved	2
	User Data Address Length	1
	User Data Address Offset	1
	In-Band Called Address	variable
	In-Band Calling Address	variable
	Out-of-Band Called Address	variable
	Out-of-Band Calling Address	variable
	User Data	variable
OUTPUTS:	none	
RECEIPT CODES:	FF by CE 00 by Layer 4 if valid primitive 11 by Layer 4 if invalid primitive	
DESCRIPTION:	Indication to Layer 4 of an incoming logical connection.	

NOTES:

VOICE/DATA INDICATOR

Voice/Data Indicator	
0	Data call
1	Voice call

SERVICE TYPE

Service Type	
0	Use init default
1	DMI Mode 3
2	Nailed-up X.25
3	X.31 (Out-of-band call setup then In-band X.25 call setup)

PVC SELECT

Hi-order bit = 1 ==> Low-order 7 bits = PVC #
Hi-order bit = 0 ==> No PVC used

Low-order 7-bits PVC to use

CHANNEL TYPE

Hi-order bit = 1 ==> Low-order 7 bits contain Channel to use
Hi-order bit = 0 ==> Use Default Voice or Data Channel Type

Low-order 7-bits	
0	D-Channel
1	B1 (B-Channel)
2	B2 (B-Channel)

3.2.2 N_CONNECT_CONFIRM

PRIMITIVE:	N_CONNECT_CONFIRM	
CODE:	N_CONN_CF	Bytes
INPUTS:	ConnID	1
	Data Length = Optional User Data Length	2
	Data Address = Optional User Data Address	4
OUTPUTS:	none	
RECEIPT CODES:	FF by CE 00 by Layer 4 if valid primitive 11 by Layer 4 if invalid primitive	
DESCRIPTION:	Confirmation to Layer 4 of successful logical connection.	
NOTES:	Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Address input is not used.	

3.2.3 N_DISCONNECT_INDICATION

PRIMITIVE:	N_DISCONNECT_INDICATION	
CODE:	N_DISC_IN	Bytes
INPUTS:	ConnID	1
	Cause	2
	(only 1 byte used for Q.931 - 5ESS BRI Spec p. IV-28)	
	Data Length = Optional User Data Length	2
	Data Address = Optional User Data Address	4
OUTPUTS:	none	
RECEIPT CODES:	FF by CE 00 by Layer 4 if valid primitive 11 by Layer 4 if invalid primitive	
DESCRIPTION:	Indication to Layer 4 that a logical connection has been disconnected.	
NOTES:	Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Address input is not used.	

3.2.4 N_DATA_INDICATION

PRIMITIVE:	N_DATA_INDICATION	
CODE:	N_DATA_IN	Bytes
INPUTS:	ConnID	1
	Data Length	2
	Data Address	4
	Data Attributes	1
OUTPUTS:	none	
RECEIPT CODES:	FF by CE 00 by Layer 4 if valid primitive 11 by Layer 4 if invalid primitive	
DESCRIPTION:	Indication to Layer 4 of incoming user data on a logical connection.	
NOTES:	DATA ADDRESS and DATA LENGTH User data is located at Data Address. The Layer 2 and Layer 3 protocol packet headers have already been "stripped." Similarly, the Data Length parameter is equal to the actual user data length. DATA ATTRIBUTES Bit --- 0 =0 => not D-bit data, =1 => D-bit data 1 =0 => not M-bit data, =1 => M-bit data 2 =0 => not Q-bit data, =1 => Q-bit data	

3.2.5 N_DATA_ACKNOWLEDGE_INDICATION

PRIMITIVE:	N_DATA_ACKNOWLEDGE_INDICATION	
CODE:	N_DACK_IN	Bytes
INPUTS:	ConnID	1
OUTPUTS:	none	
RECEIPT CODES:	FF by CE 00 by Layer 4 if valid primitive 11 by Layer 4 if invalid primitive	
DESCRIPTION:	Indication to Layer 4 of received acknowledgement for user data sent on a logical connection.	

3.2.6 N_EXPEDITED_DATA_INDICATION

PRIMITIVE:	N_EXPEDITED_DATA_INDICATION	
CODE:	N_EXP_IN	Bytes
INPUTS:	ConnID	1
	Data Length	2
	Data Address	4
OUTPUTS:	none	
RECEIPT CODES:	FF by CE 00 by Layer 4 if valid primitive 11 by Layer 4 if invalid primitive	
DESCRIPTION:	Indication to Layer 4 of received expedited data on a logical connection.	
NOTES:	Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Address input is not used. If the Data Length is non-zero, user data is located at Data Address. The Layer 2 and Layer 3 protocol packet headers have already been "stripped." Similarly, the Data Length parameter is equal to the actual user data length.	

3.2.7 N_STATUS_INDICATION

PRIMITIVE:	N_STATUS_INDICATION	
CODE:	N_STAT_IN	Bytes
INPUTS:	ConnID Status	1 1
OUTPUTS:	none	
RECEIPT CODES:	FF by CE 00 by Layer 4 if valid primitive 11 by Layer 4 if invalid primitive	
DESCRIPTION:	Indication to Layer 4 to act on a change in the status of a logical connection (e.g., Flow Control toggle).	
NOTES:	STATUS	

0 => Transmitter Busy flow control on
(Layer 4 suspends N_DATA_REQUESTs)
1 => Trensmitter Busy Flow control off
(Layer 4 resumes N_DATA_REQUESTs)
2 => Voice Handset Off Hook
3 => Voice Handset On Hook
4 => ITC Board Software unrecoverable error
5 => Begin voice call alerting
6 => End voice call alerting

3.2.8 N_RESET_INDICATION

PRIMITIVE:	N_RESET_INDICATION	
CODE:	N_RES_IN	Bytes
INPUTS:	ConnID Cause/Diagnostic Code	1 2
OUTPUTS:	none	
RECEIPT CODES:	FF by CE 00 by Layer 4 if valid primitive 11 by Layer 4 if invalid primitive	
DESCRIPTION:	Indication to Layer 4 to reset a logical connection.	

3.2.9 N_RESET_CONFIRM

PRIMITIVE:	N_RESET_CONFIRM	
CODE:	N_RES_CF	Bytes
INPUTS:	ConnID	1
OUTPUTS:	none	
RECEIPT CODES:	FF by CE 00 by Layer 4 if valid primitive 11 by Layer 4 if invalid primitive	
DESCRIPTION:	Confirmation to Layer 4 that a logical connection has been reset.	

1 General Description

1.1 Purpose

The X.25 Call Control module performs the layer 3 X.25 functions dealing with call setup and teardown. This document should be used in conjunction with the "ISDN X.25 Data Phase Interface Reference Guide," for a complete look at the functions of X.25 layer 3.

1.2 Development Environment

The X.25 Call Control module is implemented using Microsoft 'C' compiler Version 4.0 and the Microsoft Macro Assembler Version 5.0.

2 Functional Description

2.1 Inter-layer Functional Interfaces

The functional interfaces between the Network Layer Packet (NLP) entity and adjacent layers (levels) are of two basic types: "acknowledged" and "unacknowledged" Primitive Transfer Points. At an "acknowledged" type of inter-layer interface, the entity sending a primitive waits for the receiving entity to acknowledge receipt of the primtive before the sending entity proceeds with any further processing. In the "unacknowledged" case, when a layer entity sends a primitive, it does not wait for any acknowledgement of receipt of the primitive before proceeding. (These basic types are described in greater detail in the "AMD ISDN BRI Software Architecture Reference Guide.")

In different implementations, the physical mechanisms used to provide "acknowledged" and "unacknowledged" Primitive Transfer Points may vary. For example, the "unacknowledged" type Primitive Transfer Point may be implemented using the inter-process message queue services provided by one of the many real-time operating systems available. An "acknowledged" type of command may use a "mailbox" type with a receipt code field.

The NLP entity is designed to be independent from the physical mechanisms used to transfer primitives to and from adjacent layers. All primitive transfers with other layers are accomplished via operating system-independent inter-entity function calls. If a user wishes to change the physical mechansim for sending and receiving inter-layer primitives at a particular inter-layer interface, it is only necessary to re-write the appropriate inter-entity interface functions. For the NLP, these functions are:

me_nlpservice()	Called by the ME to notify the NLP that the ME has placed a primitive to be executed in the appropriate "acknowledged" Primitive Transfer Point.
nlp_meservice()	Called by the NLP to notify the ME that the NLP has placed a primitive to be executed in the appropriate "acknowledged" Primitive Transfer Point.

ce_nlp_snd()	Called by the Coordinating Entity to send a primitive to the NLP entity via the appropriate "unacknowledged" Primitive Transfer Point.
cep3()	NLP function to receive a primitive from the Coordinating Entity via the appropriate "unacknowledged" Primitive Transfer Point.
nlp_ce_snd()	Called by the NLP to send a primitive to the Coordinating Entity via the appropriate "unacknowledged" Primitive Transfer Point.
p3ce()	Coordinating Entity function to receive a primitive to the NLP entity via the appropriate "unacknowledged" Primitive Transfer Point.
l2_nlp_snd()	Called by the Layer 2 (LAPD) to send a primitive to the NLP entity via the appropriate "unacknowledged" Primitive Transfer Point.
l2p3()	NLP function to receive a primitive from Layer 2 (LAPD) via the appropriate "unacknowledged" Primitive Transfer Point.
nlp_l2_snd()	Called by the NLP to send a primitive to Layer 2 (LAPD) via the appropriate "unacknowledged" Primitive Transfer Point.
l3l2()	Layer 2 function to receive a primitive from the NLP entity via the appropriate "unacknowledged" Primitive Transfer Point.

Note: Function names ending in "service" denote inter-entity service functions used with "acknowledged" Primitive Transfer Points. Function names ending in "_snd" and "_rcv" indicate inter-entity functions used with "unacknowledged" Primitive Transfer Points.

For the sake of brevity, an "acknowledged" Primitive Transfer Point is referred to as a "mailbox" in the rest of this document. An "unacknowledged" Primitive Transfer Point is referred to as a "message exchange."

2.2 Inter-layer Communication

The following sections describe the primitives, and their parameters, of communications between the Layer 3 Network Layer Packet (NLP) package and the Coordinating Entity and Management Entity. Primitives between NLP and Layer 2 (LAPB/LAPD) are described in the "ISDN AmLINK Interface Reference Guide." The primitives listed here describe in detail the format of the messages to be passed, but do not specifically tie down the code to the method of passage. This is done deliberately to preserve operating system independence.

2.3 CE and X25_CC Communication Interface

The interface between the Coordinating Entity and the X.25 Call Control entity uses message queues, as described in the inter-layer functional interfaces section. Messages between the NLP (X.25 Call Control) and the Coordinating Entity are concerned with setup and teardown of in-band X.25 links. Most messages are able to be self-contained within the size of a message block. Incoming CONNECT messages and outgoing call CONNECT requests have more information associated with them than can be contained within a message block. Therefore, they make use of an additional buffer which is used to contain parameters. This is referred to as an "indirect parameter block."

2.3.1 CE to X25_CC Interface

The CE-to-NLP interface primitives are communicated via the "CEP3" message exchange.

The structure of a message sent by the CE to the "CEP3" message exchange is as follows:

Parameter	Length (Bytes)	Message Offset (Bytes)
Reserved for Operating System	8	0
Primitive Code	1	8
Receipt Code	1	9
Data Attributes	1	10
Data Length	2	11
Data Refnum	2	13
Cause	2	15
Logical Channel ID (LCI)	2	17
Connection ID (ConnID)	1	18
Data Address	4	22

This message format is used for all primitives passing among the Layer 4, Coordinating Entity (CE), Network Layer Packet (NLP), and Network Layer Signalling (NLS) entities. Each primitive passing from the CE to the NLP uses only a few of the parameters listed above. The detailed description of each primitive indicates which parameters are used for that primitive.

The following is a C language representation of the Layer 4/CE/NLP/NLS common message structure:

```
struct l4_ce_l3_msg {
  unsigned long reserved;  /* reserved for oper sys use */
  unsigned long home_exch; /* reserved for oper sys use */
  unsigned char primcode;  /* primitive command code */
  unsigned char receipt;   /* primitive receipt code */
  unsigned char d_attrib;  /* data attributes */
  unsigned short datalen;  /* length of data buf */
  unsigned short refnum;   /* refnum of data buf */
  unsigned char cause[2];  /* cause of DISC,RESET,STATUS */
  unsigned short lci;      /* Logical Channel Number */
  unsigned char connid;    /* CONNection ID */
  char (far *dataptr);     /* ptr to prim-specific data buf */
 } ;
```

In the CE-to-NLP direction the CE formats a primitive message including parameters and calls the "ce_nlp_snd()" function with a pointer to the primitive struct as a function argument. The NLP is awakened within the "cep3()" function when the message is received from the CE.

The following primitives are used for communication from the Coordinating Entity to X25_CC:

NLP_CONN_RQ	Request from CE to NLP to establish in-band connection.
NLP_CONN_RS	Response from CE to NLP to indicate connection confirmed.
NLP_DISC_RQ	Request from CE to NLP to disconnect in-band connection.

2.3.2 X25_CC to CE Interface

The NLP-to-CE interface primitives are communicated via the "P3CE" message exchange.

The structure of a message sent by the NLP to the "P3CE" message exchange is as follows:

Parameter	Length (Bytes)	Message Offset (Bytes)
Reserved for Operating System	8	0
Primitive Code	1	8
Receipt Code	1	9
Data Attributes	1	10
Data Length	2	11
Data Refnum	2	13
Cause	2	15
Logical Channel ID (LCI)	2	17
Connection ID (ConnID)	1	18
Data Address	4	22

This message format is used for all primitives passing among the Layer 4, Coordinating Entity (CE), Network Layer Packet (NLP), and Network Layer Signalling (NLS) entities. Each primitive passing from the NLP to the CE uses only a few of the parameters listed above. The detailed description of each primitive indicates which parameters are used for that primitive.

In the NLP-to-CE direction the NLP formats a primitive message including parameters and calls the "nlp_ce_snd()" function with a pointer to the primitive struct as a function argument. The CE is awakened within the "p3ce()" function when the message is received from the NLP.

The following primitives are used for communication from X25_CC to the Coordinating Entity:

NLP_CONN_IN	Indication from NLP to CE of an in-band connection.
NLP_CONN_CF	Confirmation from NLP to CE of established connection.
NLP_DISC_IN	Indication from NLP to CE of in-band disconnection.
NLP_DISC_CF	Confirmation from NLP to CE of in-band disconnection.

2.4 ME and X25_CC Communication Interface

The interface between the Management Entity and the X.25 Call Control entity uses "acknowledged" mailboxes, as described in the inter-layer functional interfaces section. Primitives between the NLP (X.25 Call Control) and the Management Entity are concerned with timer and buffer allocation/deallocation and LCI/CES/SAPI assignments.

2.4.1 ME to X25_CC Interface

The ME-to-NLP interface primitives are communicated via the ME_NLP_Mailbox. The ME_NLP_Mailbox is at a fixed location in RAM.

The structure of the ME_NLP_Mailbox is as follows:

	Length (Bytes)
Primitive Code	1
Receipt Code	1
Primitive Parameters	14

To initiate a primitive transfer, the ME writes a primitive code, any associated parameters, and a hex FF receipt code in the ME_NLP_Mailbox (pointed to by the "ME_NLP" mailbox pointer) and calls the "me_nlpservice()" function entry point in the the NLP entity. Once the NLP has finished processing the primitive, the NLP writes the receipt code in the mailbox and returns.

The following primitives are used for communication from the Management Entity to X25_CC:

MNLP_AS_RQ	Request from ME to NLP to assign LCI/SAPI.
MNLP_RM_RQ	Request from ME to NLP to disassociate LCI/SAPI from ConnID.
MNLP_TM_RS	Response from ME to NLP indicating timer expiration.

2.4.2 X25_CC to ME Interface

The NLP-to-ME interface primitives are communicated via the NLP_ME_Mailbox. The NLP_ME_Mailbox is at a fixed location in RAM.

The structure of the NLP_ME_Mailbox is as follows:

	Length (Bytes)
Primitive Code	1
Receipt Code	1
Primitive Parameters	14

To initiate a primitive transfer, the NLP writes a primitive code, any associated parameters, and a hex FF receipt code in the NLP_ME_Mailbox (pointed to by the "NLP_ME" mailbox pointer) and calls the "nlp_meservice()" function entry point in the the ME entity. Once the ME has finished processing the primitive, the ME writes the receipt code in the mailbox and returns.

The following primitives are used for communication from the X25_CC to Management Entity:

MNLP_TM_IN	Indication from NLP to ME indicating timer service needed.
MNLP_BF_IN	Indication from NLP to ME indicating buffer service needed.
MNLP_CID_IN	Request from the NLP to the ME to allocate a Connection ID.
MNLP_ER_IN	Indication from the NLP to the ME that an NLP Error has occurred.

3 Detailed Functional Specifications

In this section the NLP and MNLP primitives and their associated command/event codes and parameters are described in more individual depth.

The NLP primitives are used for communication interface between the Coordinating Entity and the NLP. The MNLP primitives are used between the Management Entity and the NLP.

In case of 2 byte long parameters, the least significant byte is loaded first. For example if LENGTH is a 2 byte long parameter the parameter field will have the following format:

Octet 1 Low order 8 bits of LENGTH (LSB)
Octet 2 High order 8 bits of LENGTH (MSB)

In case of address parameters, the parameter will occupy four bytes and will have the format as described below:

Octet 1 Low order 8 bits of the OFFSET
Octet 2 High order 8 bits of the OFFSET
Octet 3 Low order 8 bits of the SEGMENT
Octet 4 High order 8 bits of the SEGMENT

Each primitive description contains the following information:

PRIMITIVE:	The name of the primitive
CODE:	Mnemonic code representing the primitive
INPUT:	Input parameters required for this primitive
DESCRIPTION:	The function provided by this primitive
NOTES:	Special considerations or characteristics related for this primitive

In addition, MNLP primitives require two additional information sections, due to their use of "acknowledged" mailboxes:

RECEIPT CODES:	Codes entered into the receipt code field of the mailbox
OUTPUTS:	Output parameters as needed, placed into the mailbox by the requested entity, for use by the requestor upon acknowledgement of the primitive

3.1 CE to X25_CC Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the Coordinating Entity to X25_CC are described in this section.

3.1.1 NLP_CONNECT_REQUEST

PRIMITIVE: NLP_CONNECT_REQUEST

CODE: NLP_CONN_RQ

INPUTS:

ConnID	1 byte
Data Address = Indirect Parm Blk Addr	4 bytes
Data Refnum = Indirect Parm Blk Refnum	2 bytes

Indirect Parameter Block

Reserved	1
Service Type	1
PVC Select	1
Channel Type	1
In-band Called Address Length	1
In-band Called Address Offset in Indirect Parameter Block	1
In-band Calling Address Length	1
In-band Calling Address Offset in Indirect Parameter Block	1
Reserved	4
Receipt Confirmation Indicator	1
Expedited Data Indicator	1
Reserved	2
User Data Length	1
User Data Offset	1
In-band Called Address	var.
In-band Calling Address	var.
User Data	var.

DESCRIPTION: This primitive is used to request the establishment of an outgoing call connection.

NOTES: Service type has the following values:

0 Use init default value
1 DMI Mode 3
2 Nailed-up X.25
3 X.31 (Out-of-band call setup followed by in-band X.25 call setup)

PVC Select has the following values:
Hi-order bit = 1: Low-order 7 bits contain PVC to use.
Hi-order bit = 0: PVC # doesn't matter (call doesn't use PVC)

Channel type has the following values:
Hi-order bit = 1: Low-order 7 bits contain Channel to use
Hi-order bit = 0: Use Default Voice or Data Channel Type
Low-order 7 bits:
0 D-channel
1 B1 (B-channel)
2 B2 (B-channel)

3.1.2 NLP_CONNECT_RESPONSE

PRIMITIVE:	NLP_CONNECT_RESPONSE	
CODE:	NLP_CONN_RS	
INPUTS:	ConnID	1 byte
	Data Length = Optional User Data Length	2 bytes
	Data Address = Optional User Data Address	4 bytes
	Data Refnum = Optional User Data Refnum	2 bytes
DESCRIPTION:	This primitive is used to indicate that a connection for an incoming call connection can be completed.	
NOTES:	Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Refnum and Data Address inputs are not used.	

3.1.3 NLP_DISCONNECT_REQUEST

PRIMITIVE: NLP_DISCONNECT_REQUEST

CODE: NLP_DISC_RQ

INPUTS:

ConnID	1 byte
Cause/Diagnostic Code	2 bytes
Data Length = Optional User Data Length	2 bytes
Data Address = Optional User Data Address	4 bytes
Data Refnum = Optional User Data Refnum	2 bytes

DESCRIPTION: This primitive is used to request that a call connection be taken down.

NOTES: Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Refnum and Data Address inputs are not used.

3.2 X25_CC to CE Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the X25_CC to the Coordinating Entity are described in this section.

3.2.1 NLP_CONNECT_INDICATION

PRIMITIVE: NLP_CONNECT_INDICATION

CODE: NLP_CONN_IN

INPUTS:

ConnID	1 byte
Data Address = Indirect Parm Blk Addr	4 bytes
Data Refnum = Indirect Parm Blk Refnum	2 bytes
LCI	2 bytes

Indirect Parameter Block

Reserved	1
Service Type	1
PVC Select	1
Channel Type	1
In-band Called Address Length	1
In-band Called Address Offset in Indirect Parameter Block	1
In-band Calling Address Length	1
In-band Calling Address Offset in Indirect Parameter Block	1
Reserved	4
Receipt Confirmation Indicator	1
Expedited Data Indicator	1
Reserved	2
User Data Length	1
User Data Offset	1
In-band Called Address	var.
In-band Calling Address	var.
User Data	var.

DESCRIPTION: This primitive is used to indicate an incoming call.

NOTES: Service type has the following values:

0 Use init default value
1 DMI Mode 3
2 Nailed-up X.25
3 X.31 (Out-of-band call setup followed by in-band X.25 call setup)

PVC Select has the following values:
Hi-order bit = 1: Low-order 7 bits contain

PVC to use.
Hi-order bit = 0: PVC # doesn't matter
(call doesn't use PVC)

Channel type has the following values:
Hi-order bit = 1: Low-order 7 bits contain
Channel to use
Hi-order bit = 0: Use Default Voice or
Data Channel Type
Low-order 7 bits:
0 D-channel
1 B1 (B-channel)
2 B2 (B-channel)

3.2.2 NLP_CONNECT_CONFIRMATION

PRIMITIVE:	NLP_CONNECT_CONFIRMATION	
CODE:	NLP_CONN_CF	
INPUTS:	ConnID	1 byte
	Data Length = Optional User Data Length	2 bytes
	Data Address = Optional User Data Address	4 bytes
	Data Refnum = Optional User Data Refnum	2 bytes
DESCRIPTION:	This primitive is used to confirm that an outgoing call connection has been completed.	
NOTES:	Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Refnum and Data Address inputs are not used.	

3.2.3 NLP_DISCONNECT_INDICATION

PRIMITIVE: NLP_DISCONNECT_INDICATION

CODE: NLP_DISC_IN

INPUTS:

ConnID	1 byte
Cause/Diagnostic Code	2 bytes
Data Length = Optional User Data Length	2 bytes
Data Address = Optional User Data Address	4 bytes
Data Refnum = Optional User Data Refnum	2 bytes

DESCRIPTION: This primitive is used to indicate disconnection from the switch of a current connected line.

NOTES: Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Refnum and Data Address inputs are not used.

3.2.4 NLP_DISCONNECT_CONFIRMATION

PRIMITIVE:	NLP_DISCONNECT_CONFIRMATION	
CODE:	NLP_DISC_CF	
INPUTS:	ConnID	1 byte
DESCRIPTION:	This primitive is used to confirm that a requested disconnect has been completed.	
NOTES:		

3.3 ME to X25_CC Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the Management Entity to X25_CC are described in this section.

3.3.1 MNLP_ASSIGN_REQUEST

PRIMITIVE: MNLP_ASSIGN_REQUEST

CODE: MNLP_AS_RQ

INPUTS:

ConnID	1 byte
LCI (== 1 for DMI, dummy for CES_NLS)	2 bytes
CES	1 byte
SAPI (dummy parameter for nailed-up X.25)	1 byte
L3 Window size (W)	1 byte
L3 Max User Data Field Size	1 byte

OUTPUTS: none

RECEIPT CODES:
FF by ME
00 by NLP to indicate successful receipt
11 by NLP to indicate error in command

DESCRIPTION: This primitive is used by the Management Entity to assign a particular channel as indicated by LCI/CES/SAPI to a ConnID

NOTES: The CES implies the Physical channel; CES == CES_NLS indicates a D-channel. Likewise, CES == CES_DNUPX25 implies use of the D-channel for data. Any other value indicates use of the B-channel.

3.3.2 MNLP_REMOVE_REQUEST

PRIMITIVE: MNLP_REMOVE_REQUEST

CODE: MNLP_RM_RQ

INPUTS: ConnID 1 byte

OUTPUTS: none

RECEIPT CODES: FF by ME
00 by NLP to indicate successful receipt
11 by NLP to indicate error in command

DESCRIPTION: This primitive is used to break the connection between the ConnID and a particular channel.

NOTES:

3.3.3 MNLP_TIMER_SERVICE_RESPONSE

PRIMITIVE: MNLP_TIMER_SERVICE_RESPONSE

CODE: MNLP_TM_RS

INPUTS: Timer Refnum 2 bytes

OUTPUTS: none

RECEIPT CODES: FF by ME
00 by NLP to indicate successful receipt
11 by NLP to indicate error in command

DESCRIPTION: This primitive is used to indicate that a previously requested timer has expired.

NOTES:

3.4 X25_CC to ME Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the X25_CC to the Management Entity are described in this section.

3.4.1 MNLP_TIMER_SERVICE_INDICATION

PRIMITIVE: MNLP_TIMER_SERVICE_INDICATION

CODE: MNLP_TM_IN

INPUTS:

Service Type (0 = start timer, 1 = stop timer, 2 = restart timer)	1 byte
Timer Refnum	2 bytes
Timeout Value	2 bytes

OUTPUTS: none

RECEIPT CODES: FF by NLP
00 by ME to indicate successful receipt
11 by ME to indicate error in command

DESCRIPTION: This primitive is used to request the Management Entity to start, stop, or restart timers needed for X.25.

NOTES:

3.4.2 MNLP_BUFFER_SERVICE_INDICATION

PRIMITIVE: MNLP_BUFFER_SERVICE_INDICATION

CODE: MNLP_BF_IN

INPUTS:

Service Type (0 = allocate, 1 = deallocate)	1 byte
Min Buf Size (alloc)	2 bytes
Buffer Refnum (dealloc)	2 bytes

OUTPUTS:

Buffer Address (alloc)	4 bytes
Buffer Refnum (alloc)	2 bytes

RECEIPT CODES: FF by NLP
00 by ME to indicate request is granted
11 by ME to indicate request is denied

DESCRIPTION: This primitive is used to request buffer service -- either to allocate or deallocate a particular buffer.

NOTES: The output parameters are returned only if the request is granted.

The Buffer Refnum returned as an output during buffer allocation is used as an input during deallocation to specify which buffer to deallocate.

3.4.3 MNLP_CONNID_INDICATION

PRIMITIVE:	MNLP_CONNID_INDICATION	
CODE:	MNLP_CID_IN	
INPUTS:	none	
OUTPUTS:	ConnID	1 byte
RECEIPT CODES:	FF by NLP 00 by ME to indicate request is granted 11 by ME to indicate request is denied	
DESCRIPTION:	Request from the NLP to the ME to allocate a Connection ID.	
NOTES:	The output parameter is returned only if the request is granted.	

3.4.4 MNLP_ERROR_INDICATION

PRIMITIVE:	MNLP_ERROR_INDICATION	
CODE:	MNLP_ER_IN	
INPUTS:	NLP Error Code	1 byte
OUTPUTS:	none	
RECEIPT CODES:	FF by NLP 00 by ME to indicate request is granted 11 by ME to indicate request is denied	
DESCRIPTION:	Indication from the NLP to the ME that an NLP error has occurred.	
NOTES:	The NLP Error Codes are defined in the AmLink3 User's guide.	

1 General Description

1.1 Purpose

The X.25 Data Phase module performs the layer 3 X.25 functions dealing with data transfer and data link management. This document should be used in conjunction with the "ISDN X.25 Call Control Interface Reference Guide," for a complete look at the functions of X.25 layer 3. However, for cases of pure data transfer, with no call control features needed, this document may be used by itself.

1.2 Development Environment

The X.25 Data Phase module is implemented using Microsoft 'C' compiler Version 5.0 and the Microsoft Macro Assembler Version 5.0.

2 Functional Description

2.1 Inter-layer Functional Interfaces

The functional interfaces between the Network Layer Packet (NLP) entity and adjacent layers (levels) are of two basic types: "acknowledged" and "unacknowledged" Primitive Transfer Points. At an "acknowledged" type of inter-layer interface, the entity sending a primitive waits for the receiving entity to acknowledge receipt of the primtive before the sending entity proceeds with any further processing. In the "unacknowledged" case, when a layer entity sends a primitive, it does not wait for any acknowledgement of receipt of the primitive before proceeding. (These basic types are described in greater detail in the "AMD ISDN BRI Software Architecture Reference Guide.")

In different implementations, the physical mechanisms used to provide "acknowledged" and "unacknowledged" Primitive Transfer Points may vary. For example, the "unacknowledged" type Primitive Transfer Point may be implemented using the inter-process message queue services provided by one of the many real-time operating systems available. An "acknowledged" type of command may use a "mailbox" type with an receipt code field.

The NLP entity is designed to be independent from the physical mechanisms used to transfer primitives to and from adjacent layers. All primitive transfers with other layers are accomplished via operating system-independent inter-entity function calls. If a user wishes to change the physical mechansim for sending and receiving inter-layer primitives at a particular inter-layer interface, it is only necessary to re-write the appropriate inter-entity interface functions. For the NLP, these functions are:

me_nlpservice()	Called by the ME to notify the NLP that the ME has placed a primitive to be executed in the appropriate "acknowledged" Primitive Transfer Point.
nlp_meservice()	Called by the NLP to notify the ME that the NLP has placed a primitive to be executed in the appropriate "acknowledged" Primitive Transfer Point.
ce_nlp_snd()	Called by the Coordinating Entity to send a primitive to the NLP entity via the appropriate "unacknowledged" Primitive Transfer Point.
cep3()	NLP function to receive a primitive from the Coordinating Entity via the appropriate "unacknowledged" Primitive Transfer Point.
nlp_ce_snd()	Called by the NLP to send a primitive to the Coordinating Entity via the appropriate "unacknowledged" Primitive Transfer Point.
p3ce()	Coordinating Entity function to receive a primitive to the NLP entity via the appropriate "unacknowledged" Primitive Transfer Point.
l2_nlp_snd()	Called by the Layer 2 (LAPD) to send a primitive to the NLP entity via the appropriate "unacknowledged" Primitive Transfer Point.
l2p3()	NLP function to receive a primitive from Layer 2 (LAPD) via the appropriate "unacknowledged" Primitive Transfer Point.
nlp_l2_snd()	Called by the NLP to send a primitive to Layer 2 (LAPD) via the appropriate "unacknowledged" Primitive Transfer Point.
l3l2()	Layer 2 function to receive a primitive from the NLP entity via the appropriate "unacknowledged" Primitive Transfer Point.

Note: Function names ending in "service" denote inter-entity service functions used with "acknowledged" Primitive Transfer Points. Function names ending in "_snd" and "_rcv" indicate inter-entity functions used with "unacknowledged" Primitive Transfer Points.

For the sake of brevity, an "acknowledged" Primitive Transfer Point is referred to as a "mailbox" in the rest of this document. An "unacknowledged" Primitive Transfer Point is referred to as a "message exchange."

2.2 Inter-layer Communication

The following sections describe the primitives, and their parameters, of communications between the Layer 3 Network Layer Packet (NLP) package and the Coordinating Entity and Management Entity. Primitives between NLP and Layer 2 (LAPB/LAPD) are described in the "ISDN AmLINK Interface Reference Guide." The primitives listed here describe in detail the format of the messages to be passed, but do not specifically tie down the code to the method of passage. This is done deliberately to preserve operating system independence.

2.3 CE and X25_DP Communication Interface

The interface between the Coordinating Entity and the X25_DP entity uses message queues, as described in the inter-layer functional interfaces section. Messages between the NLP (X.25 Data Phase) and the Coordinating Entity are concerned with data transfer. Most messages are able to be self-contained within the size of a message block. Incoming and outgoing DATA and EXPED_DATA requests have data associated with them which can not be contained within a message block. Therefore, they make use of an additional buffer which is used to contain parameters. This is referred to as an "indirect parameter block."

2.3.1 CE to X25_DP Interface

The CE-to-NLP interface primitives are communicated via the "CEP3" message exchange.

The structure of a message sent by the CE to the "CEP3" message exchange is as follows:

Parameter	Length (Bytes)	Message Offset (Bytes)
Reserved for Operating System	8	0
Primitive Code	1	8
Receipt Code	1	9
Data Attributes	1	10
Data Length	2	11
Data Refnum	2	13
Cause	2	15
Logical Channel ID (LCI)	2	17
Connection ID (ConnID)	1	18
Data Address	4	22

This message format is used for all primitives passing among the Layer 4, Coordinating Entity (CE), Network Layer Packet (NLP), and Network Layer Signalling (NLS) entities. Each primitive passing from the CE to the NLP uses only a few of the parameters listed above. The detailed description of each primitive indicates which parameters are used for that primitive.

The following is a C language representation of the Layer 4/CE/NLP/NLS common message structure:

```
struct l4_ce_l3_msg {
  unsigned long reserved;  /* reserved for oper sys use */
  unsigned long home_exch; /* reserved for oper sys use */
  unsigned char primcode;  /* primitive command code */
  unsigned char receipt;   /* primitive receipt code */
  unsigned char d_attrib;  /* data attributes */
  unsigned short datalen;  /* length of data buf */
  unsigned short refnum;   /* refnum of data buf */
  unsigned char cause[2];  /* cause of DISC,RESET,STATUS */
  unsigned short lci;      /* Logical Channel Number */
  unsigned char connid;    /* CONNection ID */
  char (far *dataptr);     /* ptr to prim-specific data buf */
 } ;
```

In the CE-to-NLP direction the CE formats a primitive message including parameters and calls the "ce_nlp_snd()" function with a pointer to the primitive struct as a function argument. The NLP is awakened within the "cep3()" function when the message is received from the CE.

The following primitives are used for communication from the Coordinating Entity to X25_DP:

NLP_RES_RQ	Request from CE to NLP to RESET data link.
NLP_RES_RS	Response from CE to NLP to indicate reset acknowledged.
NLP_DATA_RQ	Request from CE to NLP to transmit data.
NLP_DACK_RQ	Request from CE to NLP to acknowledge D-bit data.
NLP_EXP_RQ	Request from CE to NLP to send expedited data.

2.3.2 X25_DP to CE Interface

The NLP-to-CE interface primitives are communicated via the "P3CE" message exchange.

The structure of a message sent by the NLP to the "P3CE" message exchange is as follows:

Parameter	Length (Bytes)	Message Offset (Bytes)
Reserved for Operating System	8	0
Primitive Code	1	8
Receipt Code	1	9
Data Attributes	1	10
Data Length	2	11
Data Refnum	2	13
Cause	2	15
Logical Channel ID (LCI)	2	17
Connection ID (ConnID)	1	18
Data Address	4	22

This message format is used for all primitives passing among the Layer 4, Coordinating Entity (CE), Network Layer Packet (NLP), and Network Layer Signalling (NLS) entities. Each primitive passing from the NLP to the CE uses only a few of the parameters listed above. The detailed description of each primitive indicates which parameters are used for that primitive.

In the NLP-to-CE direction the NLP formats a primitive message including parameters and calls the "nlp_ce_snd()" function with a pointer to the primitive struct as a function argument. The CE is awakened within the "p3ce()" function when the message is received from the NLP.

The following primitives are used for communication from X25_DP to the Coordinating Entity:

NLP_RES_IN	Indication from NLP to CE of a RESET on data link.
NLP_RES_CF	Confirmation from NLP to CE of requested RESET data link.
NLP_DATA_IN	Indication from NLP to CE of incoming data.
NLP_DACK_IN	Indication from NLP to CE of far-end window rotation.
NLP_EXP_IN	Indication from NLP to CE of received expedited data.

2.4 ME and X25_DP Communication Interface

The interface between the Management Entity and the X25_DP entity uses "acknowledged" mailboxes, as described in the inter-layer functional interfaces section. Primitives between the NLP (X.25 Data Phase) and the Management Entity are concerned with timer and buffer allocation/deallocation and LCI/CES/SAPI assignments.

2.4.1 ME to X25_DP Interface

The ME-to-NLP interface primitives are communicated via the ME_NLP_Mailbox. The ME_NLP_Mailbox is at a fixed location in RAM.

The structure of the ME_NLP_Mailbox is as follows:

	Length (Bytes)
Primitive Code	1
Receipt Code	1
Primitive Parameters	14

To initiate a primitive transfer, the ME writes a primitive code, any associated parameters, and a hex FF receipt code in the ME_NLP_Mailbox (pointed to by the "ME_NLP" mailbox pointer) and calls the "me_nlpservice()" function entry point in the the NLP entity. Once the NLP has finished processing the primitive, the NLP writes the receipt code in the mailbox and returns.

The following primitives are used for communication from the Management Entity to X25_DP:

MNLP_AS_RQ	Request from ME to NLP to assign LCI/SAPI.
MNLP_RM_RQ	Request from ME to NLP to disassociate LCI/SAPI from ConnID.
MNLP_TM_RS	Response from ME to NLP indicating timer expiration.

2.4.2 X25_DP to ME Interface

The NLP-to-ME interface primitives are communicated via the NLP_ME_Mailbox. The NLP_ME_Mailbox is at a fixed location in RAM.

The structure of the NLP_ME_Mailbox is as follows:

	Length (Bytes)
Primitive Code	1
Receipt Code	1
Primitive Parameters	14

To initiate a primitive transfer, the NLP writes a primitive code, any associated parameters, and a hex FF receipt code in the NLP_ME_Mailbox (pointed to by the "NLP_ME" mailbox pointer) and calls the "nlp_meservice()" function entry point in the the ME entity. Once the ME has finished processing the primitive, the ME writes the receipt code in the mailbox and returns.

The following primitives are used for communication from the X25_DP to Management Entity:

MNLP_TM_IN	Indication from NLP to ME indicating timer service needed.
MNLP_BF_IN	Indication from NLP to ME indicating buffer service needed.
MNLP_CID_IN	Request from the NLP to the ME to allocate a Connection ID.
MNLP_ER_IN	Indication from the NLP to the ME that an NLP Error has occurred.

3 Detailed Functional Specifications

In this section the NLP and MNLP primitives and their associated command/event codes and parameters are described in more individual depth.

The NLP primitives are used for communication interface between the Coordinating Entity and the NLP. The MNLP primitives are used between the Management Entity and the NLP.

In case of 2 byte long parameters, the least significant byte is loaded first. For example if LENGTH is a 2 byte long parameter the parameter field will have the following format:

Octet 1 Low order 8 bits of LENGTH (LSB)
Octet 2 High order 8 bits of LENGTH (MSB)

In case of address parameters, the parameter will occupy four bytes and will have the format as described below:

Octet 1 Low order 8 bits of the OFFSET
Octet 2 High order 8 bits of the OFFSET
Octet 3 Low order 8 bits of the SEGMENT
Octet 4 High order 8 bits of the SEGMENT

Each primitive description contains the following information:

PRIMITIVE:	The name of the primitive
CODE:	Mnemonic code representing the primitive
INPUT:	Input parameters required for this primitive
DESCRIPTION:	The function provided by this primitive
NOTES:	Special considerations or characteristics related for this primitive

In addition, MNLP primitives require two additional information sections, due to their use of "acknowledged" mailboxes:

RECEIPT CODES:	Codes entered into the receipt code field of the mailbox
OUTPUTS:	Output parameters as needed, placed into the mailbox by the requested entity, for use by the requestor upon acknowledgement of the primitive

3.1 CE to X25_DP Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the Coordinating Entity to X25_DP are described in this section.

3.1.1 NLP_RESET_REQUEST

PRIMITIVE: NLP_RESET_REQUEST

CODE: NLP_RES_RQ

INPUTS:

ConnID	1 byte
Cause/Diagnostic Code	2 bytes

DESCRIPTION: This primitive is used to request the resetting of an established data link.

NOTES:

3.1.2 NLP_RESET_RESPONSE

PRIMITIVE:	NLP_RESET_RESPONSE	
CODE:	NLP_RES_RS	
INPUTS:	ConnID	1 byte
DESCRIPTION:	This primitive is used to indicate that the resetting of a data link has been completed.	
NOTES:		

3.1.3 NLP_DATA_REQUEST

PRIMITIVE: NLP_DATA_REQUEST

CODE: NLP_DATA_RQ

INPUTS:

ConnID	1 byte
Data Length	2 bytes
Data Address	4 bytes
Data Refnum	2 bytes
Data Attributes	1 byte

DESCRIPTION: This primitive is used to request that a data block be transmitted.

NOTES: The ConnID is used as an identifier for a particular logical link/channel across the upper layers of the ISDN software.

User data is located at Data Address plus an offset of 7 bytes. This 7-byte "hole" is left at the "front" of the user data to allow room for the Layer 2 and Layer 3 protocol packet headers. Similarly, the Data Length parameter is equal to the actual user data length plus 7 bytes.

Data Attributes has the following values:

Bit
0 == 0, not D-bit Data;
== 1, D-bit Data
1 == 0, not M-bit Data;
== 1, M-bit Data
2 == 0, not Q-bit Data;
== 1, Q-bit Data

3.1.4 NLP_DATA_ACKNOWLEDGE_REQUEST

PRIMITIVE: NLP_DATA_ACKNOWLEDGE_REQUEST

CODE: NLP_DACK_RQ

INPUTS: ConnID 1 byte

DESCRIPTION: This primitive is used to acknowledge data that has been received (and rotate window).

NOTES:

3.1.5 NLP_EXPEDITED_DATA_REQUEST

PRIMITIVE: NLP_EXPEDITED_DATA_REQUEST

CODE: NLP_EXP_RQ

INPUTS:

ConnID	1 byte
Data Length	2 bytes
Data Address	4 bytes
Data Refnum	2 bytes

DESCRIPTION: This primitive is used to request that an expedited data block be transmitted.

NOTES: Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Refnum and Data Address inputs are not used.

If user data is present, it is located at Data Address plus an offset of 7 bytes. This 7-byte "hole" is left at the "front" of the user data to allow room for the Layer 2 and Layer 3 protocol packet headers. Similarly, the Data Length parameter is equal to the actual user data length plus 7 bytes.

3.2 X25_DP to CE Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the X25_DP to the Coordinating Entity are described in this section.

3.2.1 NLP_RESET_INDICATION

PRIMITIVE: NLP_RESET_INDICATION

CODE: NLP_RES_IN

INPUTS:

ConnID	1 byte
Cause/Diagnostic Code	2 bytes

DESCRIPTION: This primitive is used to indicate the resetting of the data link.

NOTES:

3.2.2 NLP_RESET_CONFIRMATION

PRIMITIVE:	NLP_RESET_CONFIRMATION	
CODE:	NLP_RES_CF	
INPUTS:	ConnID	1 byte
DESCRIPTION:	This primitive is used to confirm that a requested reset of the data link has been done.	
NOTES:		

3.2.3 NLP_DATA_INDICATION

PRIMITIVE: NLP_DATA_INDICATION

CODE: NLP_DATA_IN

INPUTS:

ConnID	1 byte
Data Length	2 bytes
Data Address	4 bytes
Data Refnum	2 bytes
Data Attributes	1 byte

DESCRIPTION: This primitive is used to indicate an incoming data block.

NOTES: The ConnID is used as an identifier for a particular logical link/channel across the upper layers of the ISDN software.

User data is located at Data Address. The Layer 2 and Layer 3 protocol packet headers have already been "stripped." Similarly, the Data Length parameter is equal to the actual user data length.

Data Attributes has the following values:

Bit
0 == 0, not D-bit Data;
== 1, D-bit Data
1 == 0, not M-bit Data;
== 1, M-bit Data
2 == 0, not Q-bit Data;
== 1, Q-bit Data

3.2.4 NLP_DATA_ACKNOWLEDGE_INDICATION

PRIMITIVE:	NLP_DATA_ACKNOWLEDGE_INDICATION	
CODE:	NLP_DACK_IN	
INPUTS:	ConnID	1 byte
DESCRIPTION:	This primitive is used to confirm that the far-end has acknowledged receipt of data.	
NOTES:		

3.2.5 NLP_EXPEDITED_DATA_INDICATION

PRIMITIVE: NLP_EXPEDITED_DATA_INDICATION

CODE: NLP_EXP_IN

INPUTS:

ConnID	1 byte
Data Length	2 bytes
Data Address	4 bytes
Data Refnum	2 bytes

DESCRIPTION: This primitive is used to indicate the arrival of expedited data.

NOTES: Optional User Data is not present if the Data Length parameter is zero. In this case, the Data Refnum and Data Address inputs are not used.

If user data is present, it is located at Data Address. The Layer 2 and Layer 3 protocol packet headers have already been "stripped." Similarly, the Data Length parameter is equal to the actual user data length.

3.3 ME to X25_DP Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the Management Entity to X25_DP are described in this section.

3.3.1 MNLP_ASSIGN_REQUEST

PRIMITIVE: MNLP_ASSIGN_REQUEST

CODE: MNLP_AS_RQ

INPUTS:

ConnID	1 byte
LCI (== 1 for DMI, dummy for CES_NLS)	2 bytes
CES	1 byte
SAPI (dummy parameter for nailed-up X.25)	1 byte
L3 Window size (W)	1 byte
L3 Max User Data Field Size	1 byte

OUTPUTS: none

RECEIPT CODES: FF by ME
00 by NLP to indicate successful receipt
11 by NLP to indicate error in command

DESCRIPTION: This primitive is used by the Management Entity to assign a particular channel as indicated by LCI/CES/SAPI to a ConnID

NOTES: The CES implies the Physical channel; CES == CES_NLS indicates a D-channel. Likewise, CES == CES_DNUPX25 implies use of the D-channel for data. Any other value indicates use of the B-channel.

3.3.2 MNLP_REMOVE_REQUEST

PRIMITIVE:	MNLP_REMOVE_REQUEST	
CODE:	MNLP_RM_RQ	
INPUTS:	ConnID	1 byte
OUTPUTS:	none	
RECEIPT CODES:	FF by ME 00 by NLP to indicate successful receipt 11 by NLP to indicate error in command	
DESCRIPTION:	This primitive is used to break the connection between the ConnID and a particular channel.	
NOTES:		

3.3.3 MNLP_TIMER_SERVICE_RESPONSE

PRIMITIVE: MNLP_TIMER_SERVICE_RESPONSE

CODE: MNLP_TM_RS

INPUTS: Timer Refnum 2 bytes

OUTPUTS: none

RECEIPT CODES: FF by ME
00 by NLP to indicate successful receipt
11 by NLP to indicate error in command

DESCRIPTION: This primitive is used to indicate that a previously requested timer has expired.

NOTES:

3.4 X25_DP to ME Interface Primitives and Parameters

The detailed functional specifications for the primitives directed from the X25_DP to the Management Entity are described in this section.

3.4.1 MNLP_TIMER_SERVICE_INDICATION

PRIMITIVE: MNLP_TIMER_SERVICE_INDICATION

CODE: MNLP_TM_IN

INPUTS:

Service Type (0 = start timer, 1 = stop timer, 2 = restart timer)	1 byte
Timer Refnum	2 bytes
Timeout Value	2 bytes

OUTPUTS: none

RECEIPT CODES: FF by NLP
00 by ME to indicate successful receipt
11 by ME to indicate error in command

DESCRIPTION: This primitive is used to request the Management Entity to start, stop, or restart timers needed for X.25.

NOTES:

3.4.2 MNLP_BUFFER_SERVICE_INDICATION

PRIMITIVE: MNLP_BUFFER_SERVICE_INDICATION

CODE: MNLP_BF_IN

INPUTS:

Service Type (0 = allocate, 1 = deallocate)	1 byte
Min Buf Size (alloc)	2 bytes
Buffer Refnum (dealloc)	2 bytes

OUTPUTS:

Buffer Address (alloc)	4 bytes
Buffer Refnum (alloc)	2 bytes

RECEIPT CODES:

FF by NLP
00 by ME to indicate request is granted
11 by ME to indicate request is denied

DESCRIPTION: This primitive is used to request buffer service -- either to allocate or deallocate a particular buffer.

NOTES: The output parameters are returned only if the request is granted.

The Buffer Refnum returned as an output during buffer allocation is used as an input during dealloction to specify to the ME which buffer to deallocate.

3.4.3 MNLP_CONNID_INDICATION

PRIMITIVE:	MNLP_CONNID_INDICATION	
CODE:	MNLP_CID_IN	
INPUTS:	none	
OUTPUTS:	ConnID	1 byte
RECEIPT CODES:	FF by NLP 00 by ME to indicate request is granted 11 by ME to indicate request is denied	
DESCRIPTION:	Request from the NLP to the ME to allocate a Connection ID.	
NOTES:	The output parameter is returned only if the request is granted.	

3.4.4 MNLP_ERROR_INDICATION

PRIMITIVE:	MNLP_ERROR_INDICATION	
CODE:	MNLP_ER_IN	
INPUTS:	NLP Error Code	1 byte
OUTPUTS:	none	
RECEIPT CODES:	FF by NLP 00 by ME to indicate request is granted 11 by ME to indicate request is denied	
DESCRIPTION:	Indication from the NLP to the ME that an NLP error has occurred.	
NOTES:	The NLP Error Codes are defined in the AmLink3 User's Guide.	

ISDN Terminal Coprocessor Board™

Technical Manual

AMD ISDN Systems Engineering

Revision 1.0 August 1, 1988

Publication #	Rev.	Amendment
11277	B	/0

Issue Date: October 1988

Table of Contents

1 ISDN Terminal Coprocessor Board™

- *o Member of AMD ISDN Applications Board family*
- *o Designed for use in PC-XT and PC-AT compatible computers*
- *o Simultaneous voice + data for ISDN applications*
- *o 10 MHz iAPX 188 microprocessor subsystem:*
  - *- 256 kbytes of dynamic RAM*
  - *- 64 kbytes of EPROM*
  - *- 32 kbytes of shared RAM*
  - *- DMA and interrupt-driven operation*
  - *- Digital breadboard area for customer circuits*
- *o 4-wire ISDN 'S' Interface:*
  - *- compliant with CCITT I.430*
  - *- implemented with Am79C30A Digital Subscriber Controller™*
  - *- supports point-to-point and point-to-multipoint*
  - *- D-channel protocol implemented using Am79C30A*
- *o Voice Function Interfaces:*
  - *- audio processing performed by Am79C30A*
  - *- two telephone connectors for handsets or loudspeakers*
  - *- carbon handset interface provided*
  - *- analog breadboard area for customer circuit development*
- *o 'B' Channel Support:*
  - *- Am79C401 provides 'B' channel data link controller*
  - *- DMA support for 'B' channel data transfer*
  - *- Am79C401 USART provides superset of 8250 USART*
  - *- RS232 port for a terminal, or connection to PC COMM port*
- *o Extensive Software Support:*
  - *- menu-driven control interface*
  - *- includes hex downloader program for user code*
  - *- designed to execute AMD's AmLink3™ software package, which provides ISDN voice and data (X.25) connection*

2 System Overview

The AMD ISDN Terminal Coprocessor Board™ (ITCB™) is an intelligent 80188-based plug-in board for PC-XT or PC-AT compatible personal computers, and was created to meet the following goals:

1) To provide sufficient coprocessing power to design and run ISDN software of level 3 and below on a PC.

2) To provide a demonstration and evaluation tool for the Am79C401 IDPC™ (Integrated Data Protocol Controller™) and Am79C30A DSC (Digital Subscriber Controller).

3) To realize a combination of AMD hardware and software that can be certified as compliant with ISDN switching systems.

4) To provide an environment for the development of Am79C401 and/or Am79C30A applications, such as terminal adaptation or X.25.

5) To provide an environment for the demonstration and evaluation of Am79C30A transmission performance. The board layout and design approaches utilized illustrate how to maximize audio performance in the PC environment.

3 Hardware Architecture

3.1 Block Diagram

Diagram 1 illustrates the main functional blocks of the ITCB. The "local" processor is a 10 MHz 80188, which has access to 64K bytes of EPROM and 256K bytes of dynamic RAM. The 80188 address bus is demultiplexed by a transparent latch enabled by ALE, and the ITCB address/data busses are completely separate from those of the "host" PC processor.

There are three peripheral devices available to the 80188: an Am79C30A Digital Subscriber Controller, an Am79C401 Integrated Data Protocol Controller, and a simple ALS374 providing a latched output port.

The DSC is wired to one eight-pin and two four-pin modular jacks. The eight-pin jack is dedicated as a CCITT 'S' interface connector, and one of the four-pin jacks is pre-wired as a carbon telephone handset interface. The other four-pin jack is uncommitted, and together with the analog breadboard area may be used to implement interfaces for dynamic microphones or loudspeakers. The serial port of the DSC is jumper-selectable to interface to the IDPC or to an external audio tester.

The IDPC serial port is wired to a DB9 connector through RS-232C drivers and receivers, to support the connection of external terminal devices. The IDPC also controls a 32K byte shared memory through which the local and host processors communicate.

The ALS374 latch is used by the 80188 to control the IDPC powerdown pin and DSC reset pin. TP1 through TP6 are test points which are connected to the unused outputs of the latch to simplify the addition of new board functions by the user.

The ITCB also includes buffers and decoders to interface to the PC bus. The PC processor views the ITCB as a 32K block of memory, four I/O addresses, and as an optional interrupt source for IRQ3 or IRQ4. The memory and I/O addresses are switch-selectable.

3.2 PC Interface

The PC interface consists primarily of straightforward buffering and decoding logic. Hysteresis is provided for all signal inputs, and no more than two LSTTL loads are placed on any one input.

DIP switches X29 pin 4 through X29 pin 8 are used to select the location in the PC memory space of the 32K block of shared RAM present on the ITCB. Switches 4 through 8 correspond to PC address lines A15 through A19 respectively, and are labelled next to the switches on the board. A switch setting of 'ON' designates a logical zero for the associated address bit, and a setting of 'OFF' designates a logical one. For example, setting switch 8 to OFF and switches 4 through 7 'ON' places the shared memory at $80000 to $87FFF.

DIP switches X24 pin 1 through X24 pin 8 are used to select the base address in PC I/O space of four address strobes. These strobes are only activated by a I/O READ of one of the four addresses. Switches 1 through 8 correspond to PC address lines A2 through A9 respectively, with 'ON' representing a logical zero for the associated address line as before.

The function of the I/O read strobes is as follows:

READ of I/O base +0	=	spare I/O strobe
READ of I/O base +1	=	Forces a pulse on the 80188 NMI input
READ of I/O base +2	=	Pulses the HNTAC input of the 79C401 to force the HINTOUT interrupt (ITCB to PC) inactive.
READ of I/O base +3	=	Pulses the HNTIN input of the 79C401 to force the LINTOUT interrupt (PC to ITCB) active.

Jumpers T5-T6-T7 are used to determine which, if either, of IRQ3 or IRQ4 are connected to the shared memory interrupt source HINTOUT. If the jumper is placed across T6-T7, IRQ4 is used, and if placed across T5-T6, IRQ3 is used.

Jumpers T1 through T4 are used to connect the serial port between the DSC and the IDPC. If these jumpers are removed, the serial port signals may be used to connect an audio tester to the DSC.

Example Memory Switch Settings (X29)

X29-4	X29-5	X29-6	X29-7	X29-8	Base Address
ON	ON	ON	ON	OFF	$80000
ON	OFF	ON	ON	OFF	$90000
ON	ON	OFF	ON	OFF	$A0000
ON	OFF	OFF	ON	OFF	$B0000
ON	ON	ON	OFF	OFF	$C0000

Example I/O Switch Settings (X24)

X24-1	X24-2	X24-3	X24-4	X24-5	X24-6	X24-7	X24-8	Address
ON	ON	ON	ON	ON	ON	OFF	OFF	$300
ON	ON	ON	ON	ON	OFF	OFF	OFF	$380

3.3 Microprocessor Subsystem

3.3.1 Interrupt Structure

The 80188 is provided with four level-triggered interrupts:

INT0 - Indicates an interrupt from the DSC.

INT1 - Indicates an interrupt from the IDPC Data Link Controller.

INT2 - Indicates an interrupt from the IDPC USART.

INT3 - Indicates an interrupt from the PC, routed thorough the IDPC. Its customary interpretation is that the PC processor has read a block of data in the dualport memory written by the 80188.

3.3.2 DMA Structure

The two DMA requests of the 80188 internal DMA controller are dedicated to servicing the transmit and receive FIFOs of the IDPC Data Link Controller. Of special note is the generation of the DMA acknowledge pulse IDDACK*. This pulse is active when the 80188 RD*, S6 (A19), and DEN* signals are active coincident with a 80188 access to shared RAM. Since S6 = 1 indicates a DMA cycle, IDDACK* is active when the processor is performing a DMA READ of the shared RAM area prior to writing this data to the IDPC. This ensures that the IDPC will have sufficient time to remove its DMA request in time to prevent an extra access by the DMA controller. Furthermore, this "early" acknowledge allows use of source-synchronized DMA transfers, which are approximately 20% faster than destination-synchronized DMA transfers, even though the destination is requesting the DMA.

Diagrams 10 and 11 illustrate destination and source-synchronized transfers respectively. The signal /DAC is the DMA acknowledge, DREQ is the DMA request, and /ID is the IDPC chip select. The two pulses seen on /ID preceding the activation of DREQ are writes to the IDPC transmit byte count register to initiate DMA.

Diagrams 12 and 13 illustrate finer detail of the setting of the DMA request and its subsequent removal by IDDACK*. Note that the IDPC only examines its acknowledge input when it is waiting for the last transmit byte.

3.3.3 READY Generation

The 80188 may be requested to insert wait states by two sources, the dualport memory or the DRAM state machine. In the case of the dualport memory, the 80188 will be forced to wait if it attempts to access the dualport memory after the PC has been granted a memory cycle by the IDPC dualport memory controller. This is indicated by the LRDY output of the IDPC. Similiarly, the 80188 will be forced to wait if it attempts to access DRAM after a refresh cycle has been started by the DRAM state machine. These two cases are detected by equations in the 20RA10 PAL® and an output is clocked by the 80188 CLKOUT signal in order to present the output to the 80188 synchronous ready input SRDY.

3.4 Memory Subsystem

3.4.1 Memory Map

The memory map of the ITCB is organized as follows:

$00000 through $3FFFF	256K bytes of dynamic RAM
$80000 through $87FFF	32K bytes of shared static RAM
$A0080 through $A00BF	IDPC address space
$A0100	Latch address
$A0200 through $A0208	DSC address space
$F0000 through $FFFFF	64K bytes of EPROM

The internal peripherals of the 80188 reside in I/O space, beginning at $FF00.

3.4.2 EPROM

The EPROM used is a 200 nsec or faster 64Kx8 Am27C512, and is directly connected to the demultiplexed 80188 system bus. The output enable is tied to RD* to prevent bus contention in case of an erroneous write to EPROM, and the chip enable is tied to the upper chip select UCS* of the 80188.

3.4.3 Dual-port RAM Subsystem

The operation of the IDPC dual-port memory controller (DPMC) is described in full detail in the IDPC technical manual. In essence, the DPMC samples the memory chip selects from the local and host processors, and initiates a memory access cycle if one or both of the selects are active. Simultaneous requests are arbitrated in favor of the local processor, and contention is resolved by pulling either the local or host READY line low, thereby injecting wait states.

A memory access consists of generating chip select, and output enable or write enable as appropriate to the 32K static RAM, as well as activating buffers and/or latches for either a host or local access. As can be seen on page 3 of the ITCB schematics, the IDPC provides HABE* and LABE* for the host and local address bus enables, DBE* and LDBE* for the host and local data bus enables, HDLOE* and LDLOE* for the host and local data latch output enables, and HDLE and LDLE for the host and local data latch enables.

3.4.4 Dynamic RAM Subsystem

The ISDN Terminal Coprocessor Board includes 256K bytes of dynamic memory, implemented in eight 256Kx1 120 nsec devices. The microprocessor interface to the DRAMs consists of an AMD 673102A DRAM controller and an AMD 20RA10 PAL.

The 673102A provides all of the basic support functions for the DRAMs, and interfaces cleanly to the multiplexed 80188 bus. Addresses are strobed into the 673102A by the 80188 ALE signal, whereupon a negative transition of the RASIN* input instructs the 673102 to provide the DRAMs with multiplexed addresses, row and column address strobes, and if necessary, a write enable. The 673102A is operated in auto-access mode, whereby the critical address, RAS*, and CAS* timing is internally generated. The 673102A outputs are designed to drive highly capacitive loads, and incorporate series damping resistors to reduce voltage undershoot. The DRAM controller also incorporates a refresh counter, and will generate RAS-only refresh cycles under control of its RFSH* and RASIN* inputs.

The 20RA10 PAL implements a state machine which arbitrates between normal microprocessor memory cycles and refresh cycles, and generates the appropriate refresh signal RFSH* and row address input RASIN* for the 673102A. The two DMA channels of the 80188 are dedicated to servicing the Am79C401 IDPC, and thus are not available to implement refresh as is often done. An alternative scheme is used whereby the Timer1 output of the 80188 sets a flip-flop in the 20RA10 to indicate that a request for a refresh cycle has been issued. Timer1 has been programmed to generate a periodic pulse which guarantees that the 256 rows are refreshed every 4 msec as specified by the DRAM manufacturer.

The state machine is clocked on the falling edge of the 10 MHz 80188 CLKOUT signal, and consists of eight states encoded by three state variables S2, S1, and S0. The inputs are the three state variables, the 80188 DRAM memory select DRAMEN*, and the output of the refresh request latch RLTCH. Furthermore, the state machine has the ability to drive the synchronous ready line SRDY of the

80188. The state numbers were chosen such that S2 and S1 implement the required output functions RFSH* and RASIN* directly. The state transitions are illustrated in diagram 2, and the truth table is as follows:

Inputs		**Present State**			**Next State**		
DRAMEN*	RLTCH	S2	S1	S0	S2'	S1'	S0'
0	x	1	1	1	1	0	1
1	1	1	1	1	0	1	0
1	0	1	1	1	1	1	1
x	x	0	1	0	0	0	1
x	x	0	0	1	0	0	0
x	x	0	0	0	0	1	1
0	x	0	1	1	1	1	0
1	x	0	1	1	1	1	1
x	x	1	1	0	1	0	1
x	x	1	0	1	1	0	0
x x	x	1	0	0	1	1	1

The idle state for the controller state machine is '111', which means that RFSH* and RASIN* are held HIGH. The controller will remain in state '111' until the 80188 DRAMEN* chip select is active or until the output of the flip-flop set by Timer1 is active.

When the DRAMEN* chip select is sampled active, indicating a normal memory cycle, it is known that the processor is in T2. The next state of the controller is state '101', which maintains RFSH* HIGH but drives RASIN* LOW. This is followed by state '100', which maintains RFSH* HIGH and RASIN* LOW as the processor moves to T3. Note that it is not necessary to generate CASIN* for the 673102A; in a refresh cycle the 673102A ignores CASIN*, thus it is sufficient to tie CASIN* to RASIN*. After '100' the controller returns to '111' as the processor moves to T4.

Diagram 3 illustrates the case of two back-to-back RAM accesses, and it may be seen that RASIN* is LOW for two processor clock cycles during memory access then HIGH for two processor cycles. This ensures that the 673102A will bring RAS* HIGH sufficiently long to guarantee DRAM precharge for the second memory access. For diagrams 3 through 9 signal REFR01 is the Timer1 refresh request, and REFR00 is the output of the refresh request latch RLTCH.

If the controller is in state '111' and samples DRAMEN* HIGH and RLTCH HIGH, it means that a refresh request is pending and that the processor is not currently requesting access to DRAM. In this case the controller prepares for a refresh cycle by moving to state '010', which sets RFSH* LOW to the 673102A but maintains RASIN* HIGH. It is not permissible to lower RFSH* and RASIN* simultaneously to the 673102A. The RFSH* output is tied to the reset of the refresh request latch RLTCH, which will clear the refresh request once the controller begins the refresh sequence. The next state transitions are '001' followed by '000', which lowers RASIN* for two processor cycles and initiates a refresh cycle by the 673102A. State '000' is followed by state '011', which sets RASIN* HIGH for one cycle. The need for this state could be eliminated by fast DRAMs with a small minimum RAS* HIGH time, but the improvement in processor throughput is marginal.

It is possible for the DRAMEN* chip select to be sampled active during any of states '010', '001', '000', or '011'. If this occurs, the controller drives SRDY of the 80188 LOW to insert wait states until the controller can complete the refresh cycle. These cases are illustrated in diagrams 6 through 9.

When in state '011' the controller will sample DRAMEN* to see if the processor requested DRAM access during any of the refresh states. If not, the controller returns to the idle state '111' which maintains RASIN* HIGH for a minimum of one additional cycle, and restores RFSH* to HIGH in preparation for ensuing memory accesses. If DRAMEN* is sampled LOW during '011', it means that the processor is requesting DRAM access. The controller then moves to state '110', which restores RFSH* HIGH to prepare the 673102A for a normal memory access. The controller cannot return to idle state '111' to service the memory request, because the processor would see its SRDY return HIGH too early and it would fall out of synchronization with the controller.

After state '110', the controller releases SRDY to HIGH and moves to state '101' to complete service of the memory access in the normal fashion.

If the controller is in state '111' and simultaneously samples DRAMEN* LOW and RLTCH HIGH, priority is given to the processor access to minimize the effect of refresh on processor throughput. Since the refresh request is latched, the controller will service it after completing the current memory access. This case is illustrated in diagram 5, which also serves to show that refresh is guaranteed to take place within a few cycles of the refresh request even if the processor is accessing the dynamic memory. Note that RASIN* is returned HIGH for two processor cycles between the memory accesses and refresh accesses in order to guarantee sufficient RAS* HIGH time for DRAM precharge.

3.5 Support Hardware

3.5.1 RESET Generation

The ITCB is reset upon power-up by an RC network, or it may be reset by activation of a manual pushbutton. Reset will affect the 80188, the 20RA10 PAL, and the IDPC reset. The DSC reset will be pulsed by the 80188 initialization routine, therefore it is indirectly reset upon activation of the pushbutton. The PAL will power-up with the state machine in IDLE, and with NMI inactive.

3.5.2 NMI Generation

The non-maskable interrupt of the 80188 may be pulsed by the PC processor's activation of the SETNMI* I/O strobe. The SETNMI* strobe is routed to the 20RA10 PAL, where it sets four flip-flops which are connected in series and clocked by the 80188 CLKOUT signal. The output of the last flip-flop drives the 80188 NMI input, and the input of the first flip-flop is connected to a logical '0'. After the SETNMI* signal returns to logical '1', the NMI output remains HIGH for four processor clocks. This scheme ensures that the NMI will be recognized by the processor independant of the width of the SETNMI* strobe.

3.5.3 DIP Switch Input

DIP switch X29 pin 3 is wired to a pullup resistor and to the Timer 0 input of the 80188. Since the internal Timer 0 can be enabled or disabled by the level present on the Timer 0 input, the processor has the capability in software of sensing the position of X29 pin 3. If X29 pin 3 is set to OFF, the Timer 0 input is HIGH and the board firmware interprets this condition to mean that the dual-port memory is the command console for pROBE. If X29 pin 3 is ON, the Timer 0 input is LOW and the board firmware assumes that the command console for pROBE is the USART port of the IDPC. If pROBE is not installed by the user, then the position of X29 pin 3 is irrelevant.

3.5.4 Breadboard Areas

The ITCB provides two breadboard areas where users may add additional circuitry. The area nearest the telephone connectors is the analog breadboard area, and is intended for the breadboarding of low-noise components such as operational amplifiers. Access is provided to Analog +5V, Analog ground, +12V, and -12V supplies, as well as AREF, AINB, LS1, LS2, and the pins of connector J3.

Another breadboard area is located at the opposite end of the board, and is intended for digital components. Only digital +5V and ground are provided. Note also that the DSC signals SFS, SCLK, SBOUT, and SBIN are available near the digital breadboard to simplify addition of hardware for half-channel audio testing. A DB-9 footprint labelled SPX1 is provided near the digital breadboard area as well.

3.5.5 Am79C30A Support Hardware

The only support hardware required by the DSC are the few discrete components needed in the LIU and audio paths. The LIU transformer locations have been pre-drilled to support three common transformer footprints. All necessary series resistors and load resistors have been included, and spare component positions have been provided for custom modifications.

Audio input AINA and jack J2 are pre-wired for a carbon handset interface. Audio input AINB, LS1, LS2, and signals from jack J3 are routed to the analog breadboard area where applications such as dynamic microphone circuits may be tested.

The serial port of the DSC is connected to the serial port of the IDPC when jumpers T7 through T10 are installed. If these jumpers are removed, then the DSC serial port may be accessed through the SFS, SCLK, SBIN, and SBOUT test points brought out next to the digital breadboard area.

3.5.6 Am79C401 Support Hardware

The only support hardware required by the Am79C401 are the RS-232C drivers and receivers connected to the DB-9 serial port connector. The drivers are powered from +5V and -5V to reduce noise on the ITCB.

3.6 Board Layout Considerations

3.6.1 General Design Considerations

Numerous techniques were utilized to enhance ITCB reliability and noise performance:

1) Decoupling capacitors for the DSC were placed as close to the power pins as possible, and connected directly to the power plane. AVCC is decoupled to AVSS, and DVCC to DVSS. All digital chips have 0.1 microfarad ceramic decoupling capacitors in close proximity.

2) The DSC crystal and startup capacitors were placed as close to the DSC XTAL pins as possible.

3) The RC network for the DSC MAP was placed as close to the DSC CAP1 and CAP2 pins as possible.

4) Wide traces were used for power supply connections from the edge connector.

5) The DRAM controller and DRAM areas required special attention for layout and decoupling.

6) High speed clock traces were kept as short as reasonably possible, and were isolated by ground tracks.

7) The LIU transformers were placed over a ground grid.

8) ALS logic was used instead of high speed CMOS to reduce sharp signal edges which may cause ground bounce.

9) Audio traces were separated from LIU traces in the analog area.

10) The RS232 transmitter is powered from +/-5V instead of +/-12V to reduce voltage swing and minimize interference.

3.6.2 Power Supply Layout Considerations

If the ITCB is held up to a light source, it is possible to view cuts in the Vcc and ground planes that divide the board into two unequally sized parts. The larger section is designated the digital area, and the smaller section near the telephone connectors is designated the analog area. The power plane connections made on the digital side are considered digital Vcc and Vss, whereas connections made on the other side are considered analog Vcc and Vss. The digital and analog power planes join near the PC edge connector.

This split power plane approach is ideal for optimizing the transmission performance of a mixed analog and digital device such as the AM79C30A DSC. Separate connections for analog and digital Vcc and Vss are provided on the DSC PLCC, but completely separate power supplies cannot be used since the analog and digital circuitry share a common substrate. In effect, the power plane cuts provide wide traces for the analog and digital supplies, which are joined at the edge connector. This greatly reduces the coupling of digital noise into the analog circuitry.

4 Software Architecture

4.1 System Reset

Upon reset, the 80188 first initializes the upper memory chip select range to cover the entire 64K EPROM area, since the 80188 default is not sufficient. Control is then transferred to the HW_INIT routine, which initializes the remaining 80188 registers, sets the stack pointer to $3FFFF, resets the DSC, takes the IDPC out of power-down, and prepares the DRAMs for use. (The manufacturers' datasheet for the DRAMs specifies a minimum wait of 100 microseconds after power-up followed by any 8 RAS cycles before operation is guaranteed). In addition, the HW_INIT routine determines whether the user is requesting the system console to be the DB-9 serial port or the PC interface, by sensing the position of switch X29 position 3. Since switch 3 is connected to the Timer 0 input, HW_INIT can tell if the input is "0" or "1" by programming Timer 0 to be internally clocked and externally enabled.

After completing initialization, control is transferred to the PC interface command processor, with no interrupts enabled.

4.2 PC Interface Command Processor

4.2.1 Overview

The PC Command Processor utilizes a simple semaphore scheme in predetermined locations of the shared memory to accept commands from and return status to the PC processor. The Command Processor polls location $80005 for the value $FF, which means that the PC has requested service for a command. After recognizing the $FF, the Command Processor reads location $80004 to identify the type of command requested, and then passes control to an appropriate service routine. The command types supported are as follows:

$00 - READ an IDPC register
$01 - WRITE an IDPC register
$02 - READ a DSC register
$03 - WRITE a DSC register
$04 - Transfer control to the pROBE monitor
$06 - Download an Intel MCS-86 Hex file to the ITCB
$08 - Return software revision code and signature
$0A - GOTO user code command

The format and nature of parameters passed to the service routine varies from one routine to the other, and are described individually in the ensuing sections.

Upon completion of a request, the service routine will write location $80005 with either a $00 (indicating success) or an error code. The polling loop is then repeated, except for cases "04" and "0A" which transfer control to other routines.

4.2.2 DSC/IDPC Register Server

The format of parameters passed to and from the Register Server is as follows:

$80004 - READ/WRITE field
$80005 - ACK field
$80006 - OFFSET field
$80007 - REPEAT field
$80008+ - beginning of DATA field

Upon entering the Register Server, the first task performed is to check that the OFFSET field is between 0 and 63. The offset field defines the offset of the desired register from its base address. If the offset is incorrect, an error code of "01" is written to location $80005 and control is returned to the Command Processor.

If the offset is verified, the next step is to check that the REPEAT field is between 1 and 32. Since the DSC and IDPC have FIFOs, the server supports the option of reading or writing up to 32 bytes in a single service request. If the offset is out of range, an error code of "02" is written to location $80005 and control is returned to the Command Processor.

If the REPEAT field is within range, the register server determines if the request is for the DSC or IDPC, and if it is a READ or WRITE request. In any case, a loop is initiated which performs the desired data transfer(s) from dualport memory beginning at the DATA field to the device, or in the reverse direction.

After data transfer is complete, a code of "00" is written to LOCATION $80005 and control is returned to the Command Processor.

4.2.3 pROBE Monitor

The ITCB software release 1.0 includes I/O routines for the pROBE monitor available from Software Components, but **does not include the pROBE code** due to licensing restrictions. If the user wishes to use pROBE he must purchase a copy from Software Components. Any ITCB users who wish to use pROBE in the ITCB environment are encouraged to contact AMD Telecom Systems Engineering for assistance.

This software release 1.0 provides configuration tables and I/O drivers for pROBE, with the PC port designated as console. The routines necessary to support a pROBE console on the serial port of the IDPC are relatively straightforward, but are not available at the time of this initial software release.

If the command for pROBE exit is detected, the Command Processor writes a "00" to location $80005, and then performs a long jump to location $F0040, which is the normal entry point for pROBE. At this same time, the software resident on the PC side transfers to a routine which services the pROBE console commands. The first four locations of dualport memory are reserved for the pROBE console, as follows:

$80000 TXDAT - Data from PC to pROBE
$80001 TXSEM - Semaphore location for TXDAT
$80002 RXDAT - Data from pROBE to PC
$80003 RXSEM - Semaphore location for RXDAT

The semaphore location is written to "00" after data is read by the receiving party, and written to "01" by the transmitting party.

The reader is directed to the manuals for pROBE if more information is required.

4.2.4 Program Loader

The Downloader is resident in EPROM on the ITCB, and is included so that the user can download Intel MCS-86 format Hex files from the PC to the coprocessor memory. The format of the parameter field for this program is as follows:

$80004 - COMMAND field
$80005 - ACK field
$80006 - START ls byte
$80007 - START ms byte

The START fields provide the offset address in shared RAM of where the download records begin, assuming a segment address of $8000.

The loader processes the MCS-86 formatted ascii data line-by-line, until either an end-of-block character "." or an error is encountered. The end-of-block character is inserted automatically by the loader software resident on the PC, and is necessary because the dualport memory is only 32K bytes but application programs may be much larger.

The main loader loop begins by checking if the first character of the input line is "." or ":". If a period is read, the loader writes "00" to location $80005 and passes control back to the Command Processor. If the MCS-86 colon start character is read, the task of processing a line is started, else an error code of "03" is written to location $80005 and control is returned to the Command Processor.

When processing a line, the loader first calculates the checksum of the line and compares it to the value included at the end of the line. Note that the download file is formatted in ASCII, and requires conversion to hex while processing. If the line is in error, a code of "04" is written to $80005 and control is returned to the Command Processor. If the checksum is verified correctly, the loader then examines the record field of the line to determine if the current line is a type 00, 01, 02, or 03 record.

A type 03 record is a "signon" record, and requires no processing by the loader other than to advance the SI pointer to the next line. Similiarly, a type 01 record is an "end-of-file" record and requires no processing since the loader must receive a "." to stop.

A type 02 record is an "extended address record", and changes the segment address of the data to be loaded. The loader updates its "current segment" stored in RAM, and advances the SI pointer to the next line.

A type 00 record is a data record, and includes both an offset address and the data to be loaded. The loader adds the offset to the current segment, transfers the data, and advances the SI pointer to the next line.

After processing a line, the main loader loop is repeated until either "." or a processing error is encountered.

4.2.5 Software ID Command

The purpose of the Software ID command is to provide a basic sanity check of the ITCB, and to provide the PC with the EPROM revision number. The format of the parameter field is as follows:

$80004 - COMMAND field
$80005 - ACK field
$80006 - SIG field
$80007 - REV field

The Software ID routine writes a signature of $2B to the SIG field, a revision value of $10 to the REV field, and $00 to the ACK field before returning control the the Command Processor.

4.2.6 GOTO Command

The GOTO Command is included so that control may be passed from the Command Processor to user software after downloading. The format of the parameter field is as follows:

$80004 - COMMAND field
$80005 - ACK field
$80006 - offset ls-byte
$80007 - offset ms-byte
$80008 - segment ls-byte
$80009 - segment ms-byte

The GOTO command writes $00 to location $80005, then performs a jump to the requested address.

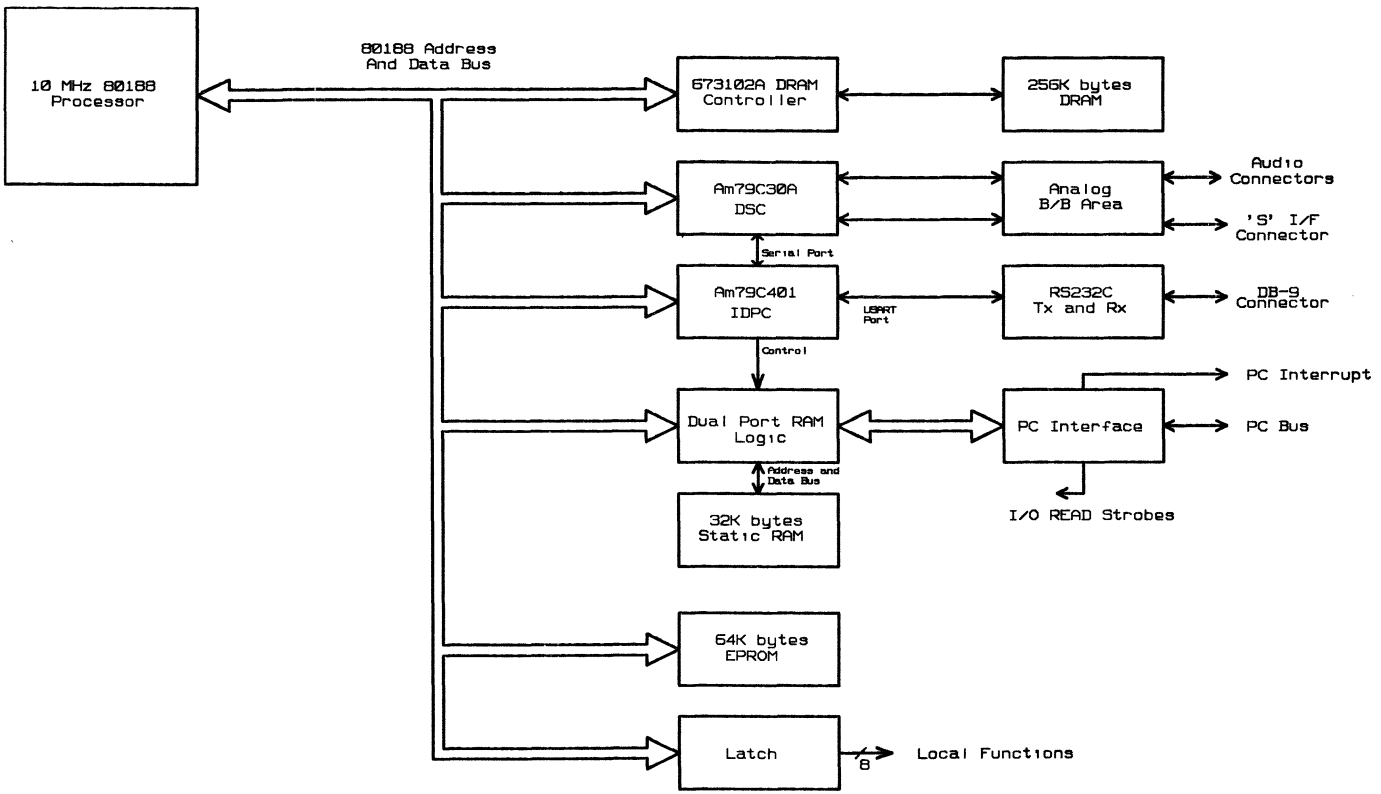


Diagram 1: ITCB Block Diagram

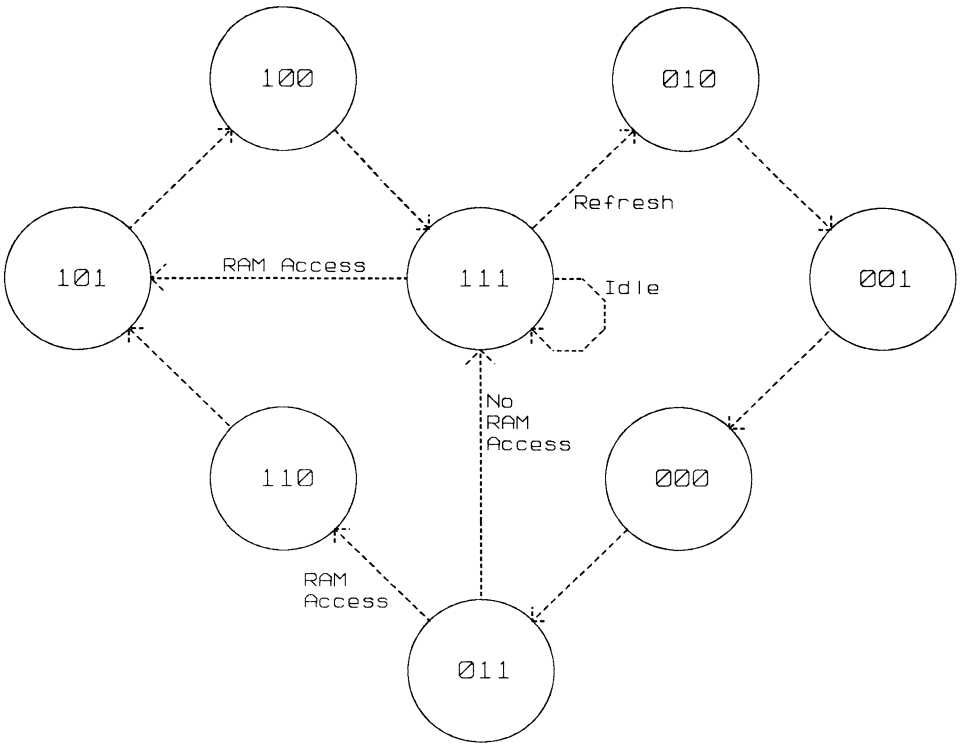


Diagram 2: DRAM Control State Transitions

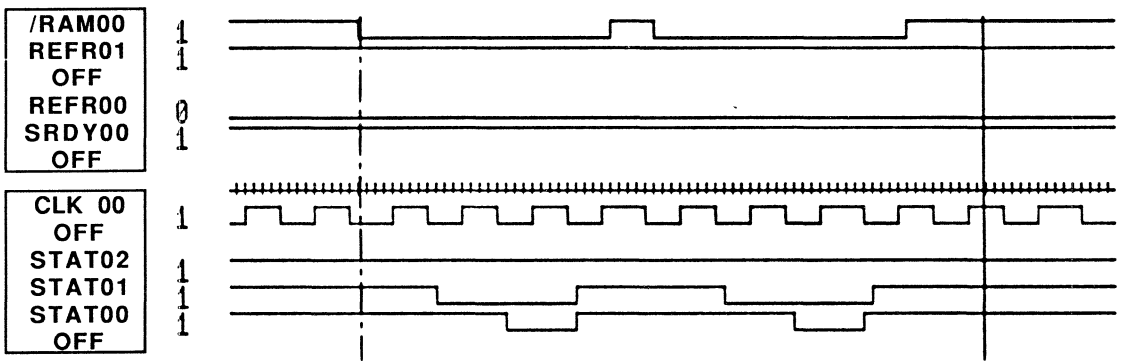


Diagram 3: Back-to-back RAM Accesses

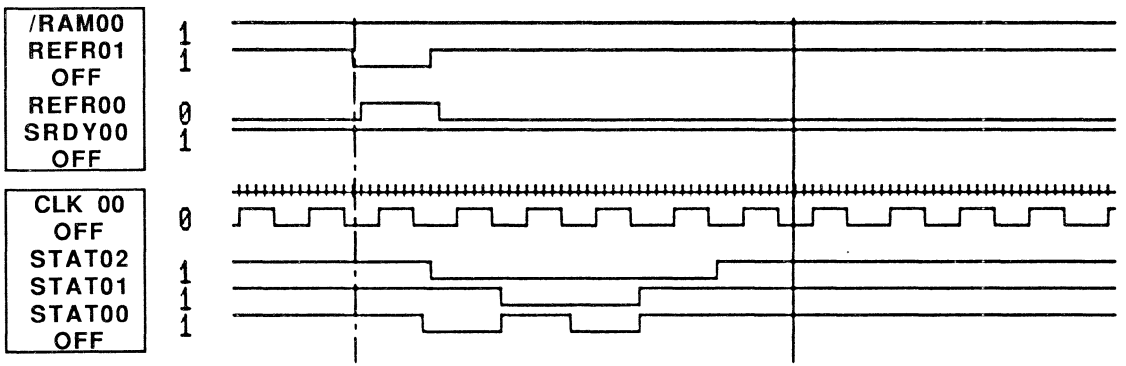


Diagram 4: Basic Refresh Cycle

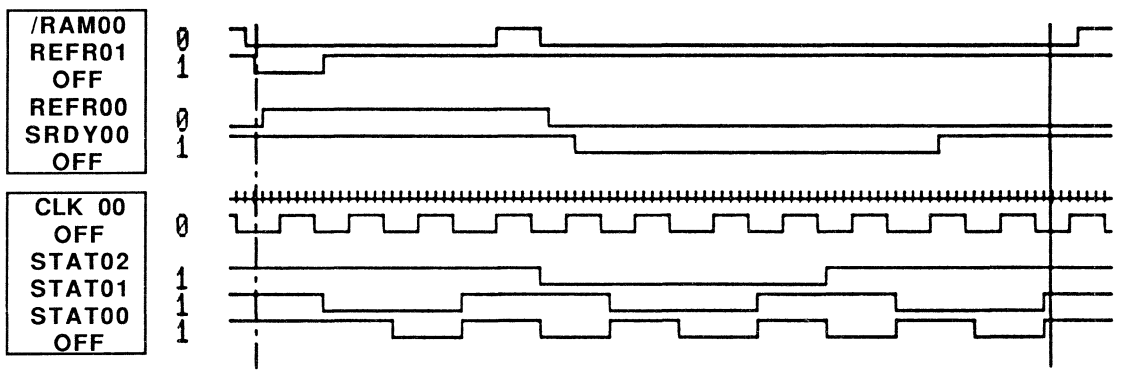


Diagram 5: Simultaneous RAM Access and Refresh Request

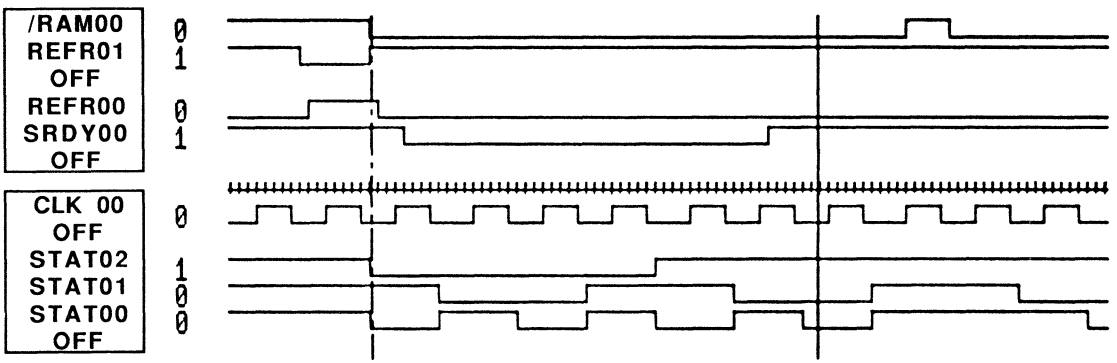


Diagram 6: RAM Access One Cycle After Refresh

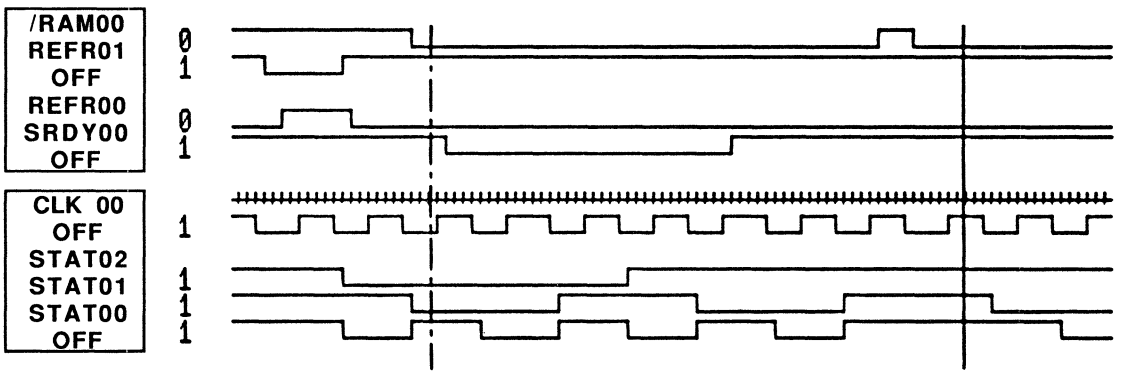


Diagram 7: RAM Access Two Cycles After Refresh

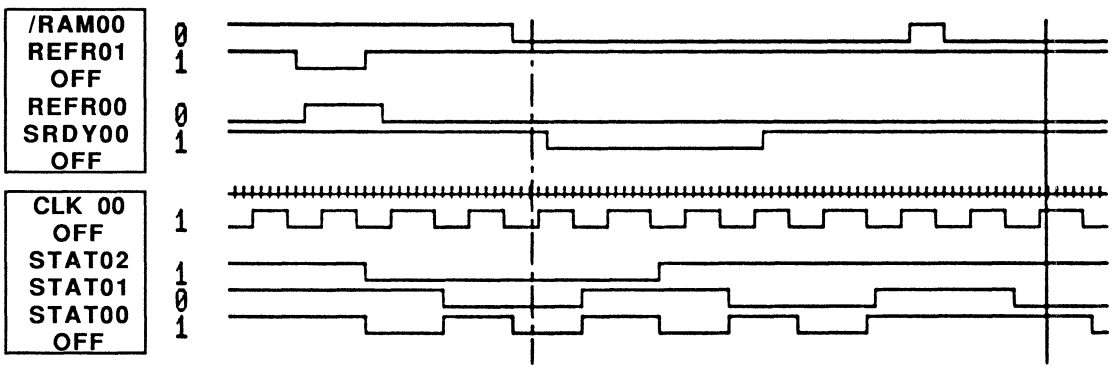


Diagram 8: RAM Access Three Cycles After Refresh

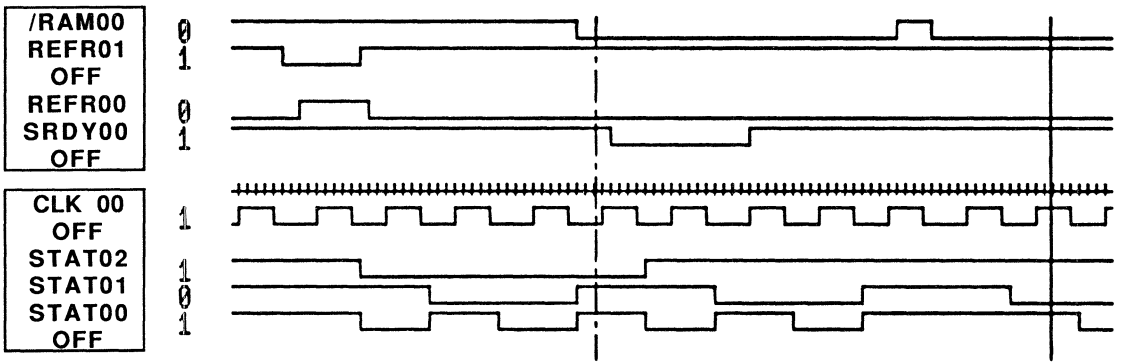


Diagram 9: RAM Access Four Cycles After Refresh

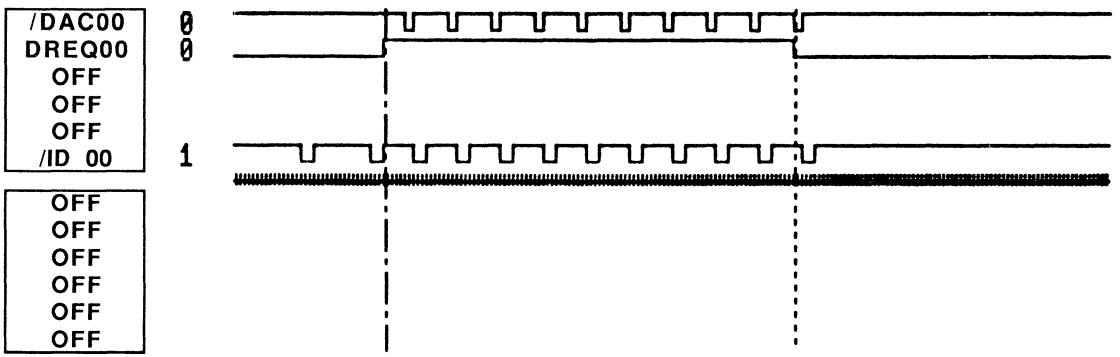


Diagram 10: Destination-synchronized DMA to IDPC

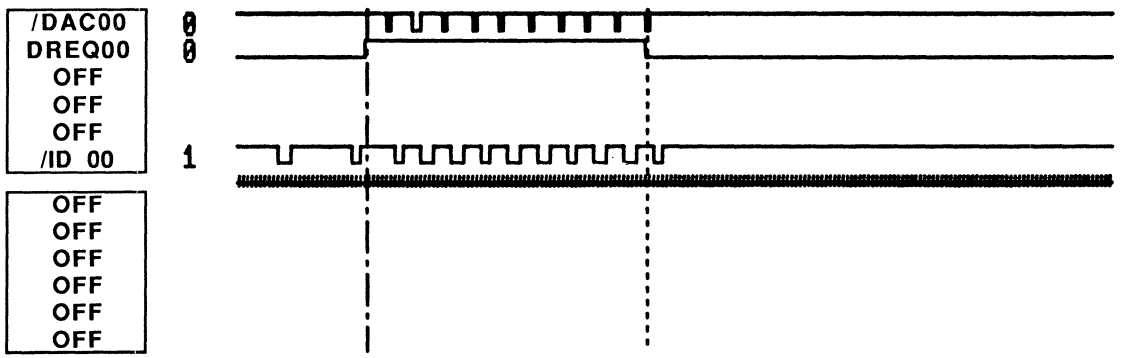


Diagram 11: Source-synchronized DMA to IDPC

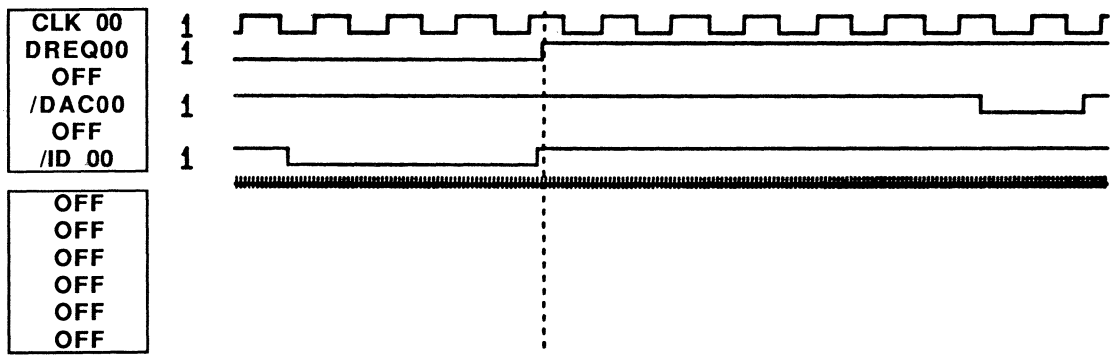


Diagram 12: Activation of DMA Request by IDPC

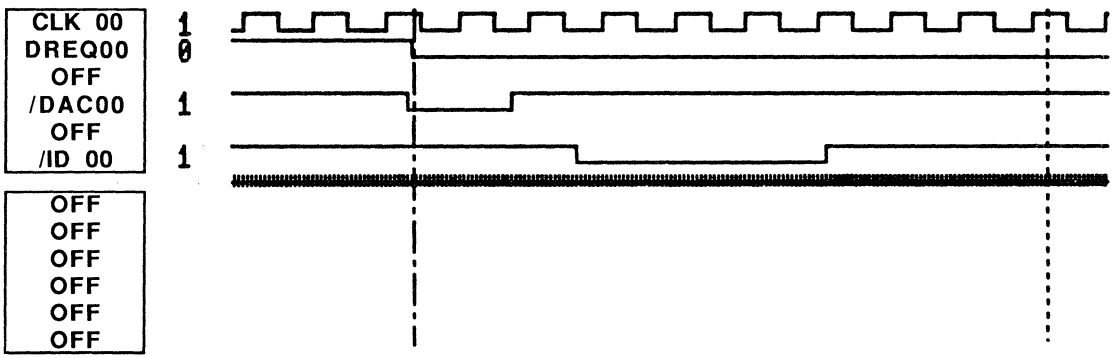


Diagram 13: Removal of DMA Request by IDPC

5 Appendix A: Parts List

Designator	Part Number & Function
X14	Saronix NYP-200-20, parallel resonant 20 MHz crystal
X34	Saronix NYP-122-20, parallel resonant 12.288 MHz crystal
X3	Augat 28-pin wide-DIP socket AMD Am27C512-200 EPROM
X5	LS08 quad AND
X6	LS14 invertor w. hysteresis
X7	LS138 decoder
X9, 10, 17, 33, 35, 36, 39	ALS244 octal buffer
X11, 15, 38	ALS373 8 trans. latch
X12	Augat 24-pin slim-DIP socket AMD 20RA10 PAL
X13	Advanced Interconnect CS068-01TG PGA socket AMD 80188-10 microprocessor
X16	ALS374 8 D-latch
X18	AMD 673102A DRAM controller
X19-22, 25-28	Micron MT1259-12 256K x 1 DIP DRAM
X23	RCA CDM62256E10 32K x 8 SRAM
X30	Textool 2-0068-06234-0x0-038-077 PLCC socket Am79C401 IDPC
X31, 32	ALS520 comparator
X37	Textool 2-0044-06232-0x0-038-077 PLCC socket Am79C30A DSC
X40	1489 RS232 rcvr
X41	1488 RS232 txr

Resistors, capacitors, diodes

X1, 2, 4, 8	Bourns 4308R101562 5.6K 8-pin bus RSIP
R1, 2, 3	1K ohm, 1%, 1/4W resistor
R4, 5	5.6 kohm, 1%, 1/4W resistor
R7, 8	39 ohm, 1%, 1/4W resistor
C4, 5, 6, 7	AVX SR211A220KAA 22 pF radial ceramic capacitor
C1, 2, 9, 10 11,12	Sprague 199D106x9010BA1 10 uF radial tantalum capacitor
C8	AVX SR211A222KAA 2.2 nF radial ceramic capacitor
C3, C13, all CX	Corning CAC02-Z5U-104-Z050A 0.1 uF 50V axial ceramic capacitors
D1	1N914 diode
Z9	510 ohm resistor
Z10	510 ohm resistor
Z11	100K ohm resistor
Z12	.22 uF radial ceramic capacitor
Z13	.01 uF radial ceramic capacitor
Z14	120 ohm resistor
Z15	200 ohm resistor
Z16	22 uF radial electrolytic capacitor
Z17	100 ohm resistor
Z18	100 ohm resistor

Mechanical, transformers

PB1	Alco TP11CG-RA0 pushbutton
X24, X29	AMP 3-435640-9 8-position rocker DIP switch
J1, J2	AMP 520249-2 4-pin modular telephone jack
J3	AMP 520252-4 8-pin modular telephone jack
P2	AMP 745112-2 9-pin female D-type connector
TF1 and TF2	'S' i/f transformer BH Electronics (part number TBD)
T1-T4	breakaway header, 2 rows of 4 posts
T5-T7	breakaway header, 1 row of 3 posts

6 Appendix B: PAL Equations

```
TITLE:      ISDN Co-processor PAL
REVISION:   001
AUTHOR:     DG
COMPANY:    Advanced Micro Devices
CHIP:       Controller PAL20RA10

;PINS   1       2       3       4       5       6
        /PL     A19     /RD     CLKOUT  DPRRDY  /RFRQ

;PINS   7       8       9       10      11      12
        /BRDRES /DEN    /RAM    /DPORT  /SETNMI GND

;PINS   13      14      15      16      17      18
        /OE     RLTCH   /IDDACK S1      S2      NMI

;PINS   19      20      21      22      23      24
        SRDY    DEL1    DEL2    S0      DEL3    VCC

STRING  ST0 ' /S2 * /S1 * /S0'
STRING  ST1 ' /S2 * /S1 *  S0'
STRING  ST2 ' /S2 *  S1 * /S0'
STRING  ST3 ' /S2 *  S1 *  S0'
STRING  ST4 '  S2 * /S1 * /S0'
STRING  ST5 '  S2 * /S1 *  S0'
STRING  ST6 '  S2 *  S1 * /S0'
STRING  ST7 '  S2 *  S1 *  S0'

EQUATIONS

/S0:=       ST7 * /RAM * RLTCH      + ST3 * RAM  + ST1 + ST5
/S1:=       ST7 *  RAM              + /S1 * S0   + ST6 + ST2
/S2:=       ST7 * /RAM * RLTCH      + ST2        + ST1 + ST0
 RLTCH:=    GND
 DEL1:=     GND
 DEL2:=     DEL1
 DEL3:=     DEL2
 NMI:=      DEL3
/SRDY:=  /  S2 * RAM                + /DPRRDY    + ST6
 IDDACK:=   DPORT * RD * A19 * DEN
```

```
S0.CLKF=                /CLKOUT
S1.CLKF=                /CLKOUT
S2.CLKF=                /CLKOUT
RLTCH.CLKF=             /BRDRES
NMI.CLKF=                CLKOUT
DEL1.CLKF=               CLKOUT
DEL2.CLKF=               CLKOUT
DEL3.CLKF=               CLKOUT
SRDY.CLKF=               CLKOUT
/IDDACK.CLKF=            GND

S0.RSTF=                 BRDRES
S1.RSTF=                 BRDRES
S2.RSTF=                 BRDRES
RLTCH.RSTF=              RFRQ
NMI.RSTF=                SETNMI
DEL1.RSTF=               SETNMI
DEL2.RSTF=               SETNMI
DEL3.RSTF=               SETNMI
SRDY.RSTF=               BRDRES
/IDDACK.RSTF=            VCC

S0.SETF=                 GND
S1.SETF=                 GND
S2.SETF=                 GND
RLTCH.SETF=              /S2
NMI.SETF=                BRDRES
DEL1.SETF=               BRDRES
DEL2.SETF=               BRDRES
DEL3.SETF=               BRDRES
SRDY.SETF=               GND
/IDDACK.SETF=            VCC
```

7 Appendix C: 80188 Software Listing

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                      1 '80188'
                      2 ;*************************************************************************
                      3 ;             ISDN Coprocessor PCB Software  Release EX B 1.0           *
                      4 ;                                                                      *
                      5 ;                                                                      *
                      6 ;*************************************************************************
                      7
                      8 ;*************************************************************************
                      9 ;                   RAM  Equates                                       *
                     10 ;*************************************************************************
                     11                     DATA
                     12
                     13                     ORG  000100000H
                     14
0000                 15 SC_DATA             DBS  16             ; System data area for pROBE
                     16
                     17                     ORG  000600000H
                     18
0000                 19 PROBE_DAT           DBS  2048           ; pROBE data area
                     20
0800                 21 LOAD_SEG            DBS  2              ; storage of current loader seg
                     22
                     23                     ORG  080000000H
0000                 24 DUALPORT                                ; Dualport RAM area
0000                 25 TXDAT               DBS  1              ; 4 bytes for pROBE I/O drivers
0001                 26 TXSEM               DBS  1
0002                 27 RXDAT               DBS  1
0003                 28 RXSEM               DBS  1
                     29
0004                 30 SCNBAS              DBS  1              ; COMMAND read/write area
0005                 31 SCNACK              DBS  1              ; COMMAND ACK area
0006                 32 SCNOFF              DBS  1              ; COMMAND offset area
0007                 33 SCNSIZ              DBS  1              ; COMMAND size area
0008                 34 SCNDAT              DBS  1              ; COMMAND data area begins
                     35
                     36 ;*************************************************************************
                     37 ;                   Define label for PROBE                             *
                     38 ;*************************************************************************
                     39                     ORG 0F0000000H
                     40
0000                 41 PROBE                                   ; equate for PROBE base address
                     42                                         ; PROBE code is inserted when
                     43                                         ; burning EPROM, not now
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                        45 ;*************************************************************************
                        46 ;                   Hardware Equates                                     *
                        47 ;*************************************************************************
                        48
                        49 ;*************************************************************************
                        50 ;                   Equates for 188 internal peripherals                 *
                        51 ;*************************************************************************
          <FFFE>        52 Reloc_Reg          EQU  0FFFEH         ; relocation register location
                        53
          <FFA2>        54 LMCS_Reg           EQU  0FFA2H         ; lower memory chip select reg
          <FFA6>        55 MMCS_Reg           EQU  0FFA6H         ; middle memory chip select reg
          <FFA0>        56 UMCS_Reg           EQU  0FFA0H         ; upper memory chip select reg
          <FFA4>        57 PACS_Reg           EQU  0FFA4H         ; peripheral chip select reg
          <FFA8>        58 MPCS_Reg           EQU  0FFA8H         ; memory/peripheral CS reg
                        59
          <FF56>        60 Tim_0_Mode         EQU  0FF56H         ; timer 0 mode register
          <FF52>        61 Tim_0_Max          EQU  0FF52H         ; timer 0 rollover register
          <FF50>        62 Tim_0_Val          EQU  0FF50H         ; timer 0 current value
          <FF5E>        63 Tim_1_Mode         EQU  0FF5EH         ; timer 1 mode register
          <FF5A>        64 Tim_1_Max          EQU  0FF5AH         ; timer 1 rollover register
          <FF58>        65 Tim_1_Val          EQU  0FF58H         ; timer 1 current value
                        66
          <FF3E>        67 Int_3_Ctrl         EQU  0FF3EH         ; interrupt 3 control reg
          <FF3C>        68 Int_2_Ctrl         EQU  0FF3CH         ; interrupt 2 control reg
          <FF3A>        69 Int_1_Ctrl         EQU  0FF3AH         ; interrupt 1 control reg
          <FF38>        70 Int_0_Ctrl         EQU  0FF38H         ; interrupt 0 control reg
                        71
          <FF36>        72 DMA_1_Int          EQU  0FF36H         ; DMA channel 1 interrupt reg
          <FF34>        73 DMA_0_Int          EQU  0FF34H         ; DMA channel 0 interrupt reg
          <FF32>        74 Timer_Int          EQU  0FF32H         ; timer interrupt register
                        75
                        76 ;*************************************************************************
                        77 ;                   Equates for microprocessor peripherals               *
                        78 ;*************************************************************************
                        79
                        80                    ORG  0A0000000H     ; base address for PCS* lines
     0000               81 PCSBAS
                        82                    ORG  0A0000080H     ; IDPC base is $A0080
     0080               83 En_IDPC            DBS  3FH
                        84                    ORG  0A0000100H
     0100               85 En_Latch           DBS  01H            ; Latch address is $A0100
                        86                    ORG  0A0000200H     ; DSC base is $A0200
     0200               87 En_DSC             DBS  08H
                        88
```

```
LOCATION OBJECT CODE LINE    SOURCE LINE

                     453 ;*************************************************************************
                     454 ;                    NEW_HEX_BYTE subroutine                             *
                     455 ;                                                                        *
                     456 ;     Converts the two ascii bytes at DS:SI into a single hex byte       *
                     457 ;     and leaves the result in AL.  SI is advanced 2 positions, and      *
                     458 ;     AH is trashed.  This routine assumes the direction bit has         *
                     459 ;     been set to increment.                                             *
                     460 ;*************************************************************************
                     461 NEW_HEX_BYTE       PROC
  0162 AC            462                    LODSB               ; fetch first string byte
  0163 2C41          463                    SUB  AL,#41H        ; check if range is A-F
  0165 7C02          464                    JL   CORR1          ; jump if range was 0-9
  0167 2C07          465                    SUB  AL,#07         ; otherwise range was A-F
  0169 0411          466 CORR1              ADD  AL,#11H        ; add correction factor
  016B C0C004        467                    ROL  AL,4           ; multiply by 16
  016E 86E0          468                    XCHG AH,AL          ; store in AH
  0170 AC            469                    LODSB               ; fetch second string byte
  0171 2C41          470                    SUB  AL,#41H        ; check if range is A-F
  0173 7C02          471                    JL   CORR2          ; jump if range was 0-9
  0175 2C07          472                    SUB  AL,#07         ; otherwise range was A-9
  0177 0411          473 CORR2              ADD  AL,#11H        ; add correction factor
  0179 02C4          474                    ADD  AL,AH          ; leave result in AL
  017B C3            475                    RET
                     476
                     477 ;*************************************************************************
                     478 ;                        FORM_ADDR subroutine                            *
                     479 ;                                                                        *
                     480 ;     Subroutine that forms a 16-bit address from the string pointed     *
                     481 ;     to by DS:SI.  The address is left in AX, and ES:BX is left         *
                     482 ;     pointing to the RAM location LOAD_SEG for convenience.  SI         *
                     483 ;     is incremented by 4, and it is assumed that the direction bit      *
                     484 ;     has been set to increment.                                         *
                     485 ;*************************************************************************
                     486 FORM_ADDR          PROC
  017C E8E3FF        487                    CALL NEW_HEX_BYTE   ; fetch first byte of address
  017F 8AD8          488                    MOV  BL,AL          ; save byte in BL
  0181 E8DEFF        489                    CALL NEW_HEX_BYTE   ; fetch second byte of address
  0184 8AE3          490                    MOV  AH,BL          ; AX contains new segment
  0186 BB6000        491                    MOV  BX,SEG LOAD_SEG
  0189 8EC3          492                    MOV  ES,BX          ; ES --> RAM work location
  018B BB0008        493                    MOV  BX,OFFSET LOAD_SEG
  018E C3            494                    RET
                     495
```

```
LOCATION OBJECT CODE LINE    SOURCE LINE

                     148
                     149 ;*************************************************************************
                     150 ;                    pROBE AND pSOS I/O Drivers                          *
                     151 ;*************************************************************************
                     152                      ORG  0F8570000H
                     153
                     154 IOINIT               PROC FAR            ; pROBE I/O initialization
0000 CB              155                      RET                 ; return to pROBE
                     156 ;*************************************************************************
                     157                      ORG  0F85B0000H
                     158
                     159 CONSTS               PROC FAR            ; pROBE console status routine
0000 9A1E0067F8      160                      CALL FAR PTR IOMODE ; PC or RS232 console?
0005 7305            161                      JNC  CONSTS1        ; if carry = 0, UART is console
0007 EA000067F8      162                      JMP  FAR PTR HSTSTS ; if carry = 1, PC is console
                     163
000C CB              164 CONSTS1              RET                 ; return to pROBE
                     165 ;*************************************************************************
                     166                      ORG  0F85F0000H
                     167
                     168 CONIN                PROC FAR            ; pROBE console input routine
0000 9A1E0067F8      169                      CALL FAR PTR IOMODE ; determine PC or UART console
0005 7305            170                      JNC  CONIN1         ; if carry = 0 UART is console
0007 EA000061F8      171                      JMP  FAR PTR HSTIN  ; if carry = 1 PC is console
000C CB              172 CONIN1               RET
                     173 ;*************************************************************************
                     174                      ORG 0F8610000H
                     175
                     176 HSTIN                PROC FAR            ; pROBE host input routine
0000 06              177                      PUSH ES             ; save current ES
0001 B80080          178                      MOV  AX,SEG DUALPORT
0004 8EC0            179                      MOV  ES,AX          ; ES points to dualport
0006 26A00000        180                      MOV  AL,ES:TXDAT    ; load input character
000A 26C6060100      181                      MOV  ES:TXSEM,$00   ; clear semaphore
0010 07              182                      POP  ES
0011 CB              183                      RET                 ; return to pROBE
                     184 ;*************************************************************************
                     185                      ORG  0F8630000H
                     186
                     187 CONOUT               PROC FAR            ; pROBE console output routine
0000 50              188                      PUSH AX
0001 9A1E0067F8      189                      CALL FAR PTR IOMODE ; PC or RS232 console?
0006 58              190                      POP  AX
0007 7305            191                      JNC  CONOUT1        ; if carry = 0, UART is console
0009 EA000065F8      192                      JMP  FAR PTR HSTOUT ; if carry = 1, PC is console
                     193
000E CB              194 CONOUT1              RET
                     195 ;*************************************************************************
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                    497 ;*************************************************************************
                    498 ;                          SUM_CHECK subroutine                          *
                    499 ;                                                                        *
                    500 ;     Subroutine to verify the checksum of the line pointed to by        *
                    501 ;     DS:SI.  The carry bit will be returned as "1" if the checksum      *
                    502 ;     at the end of the line matches the calculated sum, otherwise       *
                    503 ;     carry will be returned as "0".  The byte count is extracted        *
                    504 ;     from the line, and returned in DL.  SI is incremented by 2         *
                    505 ;     to point to the address field of the current line.  DI returns     *
                    506 ;     a pointer to the next line.  It is assumed that the direction      *
                    507 ;     bit has been set to increment.                                     *
                    508 ;*************************************************************************
                    509 SUM_CHECK          PROC
   018F 53          510                    PUSH BX
   0190 51          511                    PUSH CX
   0191 B90400      512                    MOV  CX,#04H        ; there are 4 overhead bytes
                    513
   0194 E8CBFF      514                    CALL NEW_HEX_BYTE   ; fetch #bytes
   0197 8AD0        515                    MOV  DL,AL          ; save #bytes in DL
   0199 8AD8        516                    MOV  BL,AL          ; first part of sum
   019B 02C8        517                    ADD  CL,AL          ; CX now contains # bytes to sum
   019D 56          518                    PUSH SI             ; save SI
                    519
   019E E8C1FF      520 SUM_LP             CALL NEW_HEX_BYTE   ; get next byte
   01A1 02D8        521                    ADD  BL,AL          ; add to sum
   01A3 E2F9        522                    LOOP SUM_LP         ; repeat until done
                    523
   01A5 F9          524                    STC                 ; carry=1 for 'correct'
   01A6 7401        525                    JZ   SUM_DONE       ; exit if OK
   01A8 F5          526                    CMC                 ; otherwise carry=0 for 'error'
   01A9 8BFE        527 SUM_DONE           MOV  DI,SI          ; save next line pointer
   01AB 5E          528                    POP  SI             ; restore SI pointer
   01AC 59          529                    POP  CX
   01AD 5B          530                    POP  BX
   01AE C3          531                    RET
                    532
```

```
LOCATION OBJECT CODE LINE    SOURCE LINE

                     361
                     362 ;*************************************************************************
                     363 ;                  PC Interface Command Processor                       *
                     364 ;                                                                       *
                     365 ;      This routine processes commands received from the PC through     *
                     366 ;      the dualport RAM.  Operational details are provided in the       *
                     367 ;      ISDN Terminal Coprocessor Board users' manual.                   *
                     368 ;*************************************************************************
                     369
    00D5 FC          370 SCANIN           CLD                    ; set direction flag for inc
                     371
    00D6 33C0        372 SCANRP           XOR  AX,AX             ; clear AX
    00D8 BB0080      373                  MOV  BX,SEG DUALPORT
    00DB 8EDB        374                  MOV  DS,BX             ; DS points to dualport RAM
    00DD BE0500      375                  MOV  SI,OFFSET SCNACK  ; SI --> ACK field
    00E0 8804        376                  MOV  [SI],AL           ; return 'done' status
    00E2 BF0800      377                  MOV  DI,OFFSET SCNDAT  ; DI --> DAT field of DPR
                     378
    00E5 803CFF      379 SCANLP           CMP  DS:BYTE PTR [SI],#0FFH
    00E8 75FB        380                  JNE  SCANLP
                     381
    00EA 8A1E0400    382                  MOV  BL,DS:SCNBAS   ; load service request byte
    00EE 81E30E00    383                  AND  BX,#00EH       ; save only 3 bits
    00F2 46          384                  INC  SI             ; SI --> offset descriptor
    00F3 2EFFA7F800  385                  JMP  CS:WORD PTR JMPTAB[BX] ; execute jump
                     386
    00F8 0E01        387 JMPTAB           DW   OFFSET REGS    ; IDPC/DSC register R/W server
    00FA 0E01        388                  DW   OFFSET REGS    ; IDPC/DSC register R/W server
    00FC 0401        389                  DW   OFFSET PROBEX  ; prepare to return to pROBE
    00FE AF01        390                  DW   OFFSET LOADER  ; I-HEX downloader
    0100 1002        391                  DW   OFFSET SWID    ; Software revision ID
    0102 2202        392                  DW   OFFSET GOTO    ; GOTO command
                     393
                     394 ;*************************************************************************
                     395 ;                  Routine for returning to pROBE                       *
                     396 ;*************************************************************************
    0104 C606050000  397 PROBEX           MOV  DS:SCNACK,#00  ; return 'done' to PC
    0109 EA400000F0  398                  JMP  FAR PTR PROBE+40H  ; return to pROBE
                     399
```

```
LOCATION OBJECT CODE LINE    SOURCE LINE

                     401 ;*************************************************************************
                     402 ;                    IDPC and DSC Register R/W Server                    *
                     403 ;*************************************************************************
    010E AC          404 REGS             LODSB               ; load offset descriptor
    010F A8C0        405                  TEST AL,#0C0H       ; 0 <= offset <= 64
    0111 753F        406                  JNZ  OFFERR         ; process offset error
    0113 8BD0        407                  MOV  DX,AX          ; save offset in DX
                     408
    0115 AC          409                  LODSB               ; load repeat descriptor
    0116 FEC8        410                  DEC  AL             ; (to simplify range check)
    0118 A8E0        411                  TEST AL,#0E0H       ; 1 <= repeat <= 32
    011A 753E        412                  JNZ  REPERR         ; repeat is max of 32
    011C FEC0        413                  INC  AL             ; correct repeat
    011E 8AC8        414                  MOV  CL,AL          ; CL contains repeat counter
                     415
    0120 A00400      416                  MOV  AL,DS:SCNBAS   ; load service descriptor
    0123 BE8000      417                  MOV  SI,OFFSET En_IDPC     ; SI points to IDPC
    0126 A802        418                  TEST AL,#02H        ; IDPC or DSC request?
    0128 7403        419                  JZ   PNTOK          ; SI points to IDPC
    012A BE0002      420                  MOV  SI,OFFSET En_DSC      ; SI points to DSC
                     421
    012D 03F2        422 PNTOK            ADD  SI,DX          ; SI --> device + reg offset
    012F D0D8        423                  RCR  AL,01          ; check LSB for read or write
    0131 7210        424                  JC   WRITOP
                     425
    0133 8CDB        426 READOP           MOV  BX,DS          ; correct source and dest
    0135 8EC3        427                  MOV  ES,BX
    0137 BB00A0      428                  MOV  BX,SEG PCSBAS
    013A 8EDB        429                  MOV  DS,BX
                     430
    013C 8A04        431 READLP           MOV  AL,DS:[SI]     ; load data from device
    013E AA          432                  STOSB               ; write data to RAM
    013F E2FB        433                  LOOP READLP         ; repeat until done
    0141 EB93        434                  JMP  SCANRP         ; resume scanning
                     435
    0143 87F7        436 WRITOP           XCHG SI,DI          ; correct source and dest
    0145 BB00A0      437                  MOV  BX,SEG PCSBAS
    0148 8EC3        438                  MOV  ES,BX          ; ES points to peripherals
                     439
    014A AC          440 WRITLP           LODSB               ; get data from RAM
    014B 268805      441                  MOV  ES:[DI],AL     ; write data to device
    014E E2FA        442                  LOOP WRITLP         ; repeat until done
                     443
    0150 EB84        444                  JMP  SCANRP         ; resume scanning
                     445
    0152 C606050001  446 OFFERR           MOV  DS:SCNACK,#01  ; return 'offset error' to PC
    0157 E97CFF      447                  JMP  SCANRP         ; resume scanning
                     448
    015A C606050002  449 REPERR           MOV  DS:SCNACK,#02  ; return 'size error'
    015F E974FF      450                  JMP  SCANRP         ; resume scanning
                     451
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     90
                     91                    PROG
                     92                    ASSUME CS:PROG,DS:DATA
                     93 ;*********************************************************************
                     94 ;                  Power-on reset vector                             *
                     95 ;*********************************************************************
                     96                    GLB START
                     97                    ORG  0FFFF0000H
                     98
0000 BAA0FF          99                    MOV  DX,UMCS_Reg    ; point DX to upper CS reg
0003 B83CF0         100                    MOV  AX,0F03CH      ; F038 means F0000 base
                    101                                        ; 0 waits, ignore ext ready
0006 EF             102                    OUT  DX,AX          ; expand UCS default range
0007 EA280067F8     103                    JMP  FAR PTR START  ; Jump to START upon reset
                    104
                    105 ;*********************************************************************
                    106 ;                  Define System Tables for pROBE and pSOS           *
                    107 ;*********************************************************************
                    108                    ORG  0F84E0000H
                    109
0000 00001000       110 SYS_CON_TAB        DD   SC_DATA        ; system data area
0004 01000000       111                    DD   01H            ; processor type is 80188
0008 01000000       112                    DD   01H            ; pROBE starts first
000C 00000000       113                    DD   00H            ; reserved
0010 00000000       114                    DD   00H            ; pRISM not installed
0014 00000000       115                    DD   00H            ; pRISM not installed
0018 000053F8       116                    DD   SC_PROBE_C     ; address of pROBE config table
001C 000000F0       117                    DD   PROBE          ; address of pROBE code
0020 00000000       118                    DD   00H            ; pSOS not installed
0024 00000000       119                    DD   00H            ; pSOS not installed
0028 00000000       120                    DD   00H            ; pHILE not installed
002C 00000000       121                    DD   00H            ; pHILE not installed
0030 00000000       122                    DD   00H            ; reserved
0034 00000000       123                    DD   00H            ; reserved
0038 00000000       124                    DD   00H            ; reserved
003C 00000000       125                    DD   00H            ; reserved
0040 00004EF8       126 SYS_ANCHOR         DD   SYS_CON_TAB    ; anchor points to sys con table
                    127
                    128                    ORG  0F8530000H
                    129
0000 000000F0       130 SC_PROBE_C         DD   PROBE          ; pROBE code pointer
0004 00006000       131                    DD   PROBE_DAT      ; pROBE data area
0008 000057F8       132                    DD   IOINIT         ; initialization routine
000C 00005BF8       133                    DD   CONSTS         ; console status routine
0010 00005FF8       134                    DD   CONIN          ; console input routine
0014 000063F8       135                    DD   CONOUT         ; console output routine
0018 000067F8       136                    DD   HSTSTS         ; host status routine
001C 000061F8       137                    DD   HSTIN          ; host input routine
0020 000065F8       138                    DD   HSTOUT         ; host output routine
0024 00000000       139                    DD   00000000H      ; no symval routine
0028 00000000       140                    DD   00000000H      ; no valsym routine
002C 00000000       141                    DD   00000000H      ; no urcom routine
0030 0000           142                    DW   00000000H      ; disable interrupts in pROBE
0032 0100           143                    DW   00000001H      ; enable trace refresh
0034 0000           144                    DW   00000000H      ; normal pROBE startup
0036 00000000       145                    DD   00000000H      ; reserved
003A 00000000       146                    DD   00000000H      ; reserved
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     293                              ; priority = 5
    0069 EF          294                  OUT  DX,AX          ; load interrupt 1 control word
                     295
    006A BA38FF      296                  MOV  DX,Int_0_Ctrl  ; point DX to interrupt 0 reg
    006D B81C00      297                  MOV  AX,001CH       ; level trigger
                     298                                      ; interrupt masked
                     299                                      ; priority = 4
    0070 EF          300                  OUT  DX,AX          ; load interrupt 0 control word
                     301
    0071 BA36FF      302                  MOV  DX,DMA_1_Int   ; point DX to DMA 1 int. reg
    0074 B80900      303                  MOV  AX,0009H       ; interrupt masked
                     304                                      ; priority = 1
    0077 EF          305                  OUT  DX,AX          ; load DMA 1 int. reg
                     306
    0078 BA34FF      307                  MOV  DX,DMA_0_Int   ; point DX to DMA 0 int. reg
    007B B80800      308                  MOV  AX,0008H       ; interrupt masked
                     309                                      ; priority = 0
    007E EF          310                  OUT  DX,AX          ; load DMA 0 int. reg
                     311
    007F BA32FF      312                  MOV  DX,Timer_Int   ; point DX to timer int. reg
    0082 B80B00      313                  MOV  AX,000BH       ; interrupt masked
                     314                                      ; priority = 3
    0085 EF          315                  OUT  DX,AX          ; load timer interrupt reg
                     316
    0086 BA56FF      317                  MOV  DX,Tim_0_Mode  ; point DX to timer 0 mode
    0089 B801C0      318                  MOV  AX,0C001H      ; timer enabled, internal clock,
                     319                                      ; no int, continuous,
                     320                                      ; external enable
    008C EF          321                  OUT  DX,AX          ; load timer 0 control word
    008D BA50FF      322                  MOV  DX,Tim_0_Val   ; point to counter value
    0090 ED          323                  IN   AX,DX          ; load timer value
    0091 8BD8        324                  MOV  BX,AX          ; save timer value
    0093 90          325                  NOP                 ; do nothing, let counter run
    0094 90          326                  NOP
    0095 90          327                  NOP
    0096 ED          328                  IN   AX,DX          ; load new timer value
    0097 3BC3        329                  CMP  AX,BX          ; has timer been running?
    0099 BA56FF      330                  MOV  DX,Tim_0_Mode
    009C B80040      331                  MOV  AX,04000H      ; prepare to turn off timer
    009F 7403        332                  JE   INIT1          ; console is serial port?
    00A1 B80240      333                  MOV  AX,04002H      ; otherwise console is PC
    00A4 EF          334 INIT1            OUT  DX,AX          ; store result in bit 1
                     335
    00A5 BA5AFF      336                  MOV  DX,Tim_1_Max   ; point to rollover register
    00A8 B82500      337                  MOV  AX,0037        ; max count is 37 clock cycles
    00AB EF          338                  OUT  DX,AX          ; load rollover register
                     339
    00AC BA5EFF      340                  MOV  DX,Tim_1_Mode  ; Timer 1 is DRAM refresh timer
    00AF B801C0      341                  MOV  AX,0C001H      ; Timer enabled, internal clock,
                     342                                      ; no int, continuous,
                     343                                      ; external enable
    00B2 EF          344                  OUT  DX,AX          ; load Timer 1 control word
                     345
    00B3 B800A0      346                  MOV  AX,SEG PCSBAS
    00B6 8EC0        347                  MOV  ES,AX          ; point ES to peripheral block
    00B8 26C6060001  348                  MOV  ES:En_Latch,02H ; 30A reset, 401 p. down
    00BE 26C6060001  349                  MOV  ES:En_Latch,01H ; 30A not reset, 401 not p. down
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     236 ;*************************************************************************
                     237 ;                Hardware Initialization Routine                        *
                     238 ;*************************************************************************
                     239
                     240 START
0028                 241 HW_INIT           MOV  DX,Reloc_Reg   ; point DX to relocation reg
0028 BAFEFF          242                   MOV  AX,00FFH       ; no ESC trap
002B B8FF00          243                                       ; non-IRMX interrupt mode
                     244                                       ; register block in I/O
                     245                                       ; register block at $FF00
002E EF              246                   OUT  DX,AX          ; load relocation register
                     247
002F BAA2FF          248                   MOV  DX,LMCS_Reg    ; point DX to lower CS reg
0032 B8F83F          249                   MOV  AX,3FF8H       ; 3FF8 means 256K range
                     250                                       ; 0 waits, use ext ready
0035 EF              251                   OUT  DX,AX          ; load lower CS reg
                     252
0036 BAA6FF          253                   MOV  DX,MMCS_Reg    ; point DX to middle CS reg
0039 B8F881          254                   MOV  AX,81F8H       ; 81F8 means 80000 base
                     255                                       ; 0 waits, use ext ready
003C EF              256                   OUT  DX,AX          ; load middle CS reg
                     257
003D BAA8FF          258                   MOV  DX,MPCS_Reg    ; point DX to mem/periph CS reg
0040 B8FD90          259                   MOV  AX,90FDH       ; 32K select size for MCS0:3
                     260                                       ; 7 PCS* lines
                     261                                       ; peripherals in mem space
                     262                                       ; 1 wait state for PCS4:6
                     263                                       ; no external ready for PCS4:6
0043 EF              264                   OUT  DX,AX          ; load mem/periph CS register
                     265
0044 BAA4FF          266                   MOV  DX,PACS_Reg    ; point DX to peripheral CS reg
0047 B83CA0          267                   MOV  AX,0A03CH      ; peripheral base is $A0000
                     268                                       ; no wait states for PCS0:3
                     269                                       ; no external ready for PCS0:3
004A EF              270                   OUT  DX,AX          ; load peripheral CS register
                     271
004B BA3EFF          272                   MOV  DX,Int_3_Ctrl  ; point DX to interrupt 3 reg
                     273
004E B80030          274                   MOV  AX,$3000H      ; set stack pointer
0051 8ED0            275                   MOV  SS,AX
0053 B8FFFF          276                   MOV  AX,$0FFFFH
0056 8BE0            277                   MOV  SP,AX
                     278
0058 B81F00          279                   MOV  AX,001FH       ; level trigger
                     280                                       ; interrupt masked
                     281                                       ; priority = 7
005B EF              282                   OUT  DX,AX          ; load interrupt 3 control word
                     283
005C BA3CFF          284                   MOV  DX,Int_2_Ctrl  ; point DX to interrupt 2 reg
005F B81E00          285                   MOV  AX,001EH       ; level trigger
                     286                                       ; interrupt masked
                     287                                       ; priority = 6
0062 EF              288                   OUT  DX,AX          ; load interrupt 2 control word
                     289
0063 BA3AFF          290                   MOV  DX,Int_1_Ctrl  ; point DX to interrupt 1 reg
0066 B81D00          291                   MOV  AX,001DH       ; level trigger
                     292                                       ; interrupt maked
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     588 ;*************************************************************************
                     589 ;                   Software Revision ID Routine                        *
                     590 ;*************************************************************************
                     591
    0210 C606070010  592 SWID              MOV  DS:SCNSIZ,#10H ; software revision 1.0
    0215 C60606002B  593                   MOV  DS:SCNOFF,#2BH ; signature
    021A C606050000  594                   MOV  DS:SCNACK,#00  ; 'done' code
    021F E9B4FE      595                   JMP  SCANRP         ; resume scanning
                     596
                     597 ;*************************************************************************
                     598 ;                   GOTO Routine                                        *
                     599 ;*************************************************************************
    0222 C606050000  600 GOTO              MOV  DS:SCNACK,#00          ; 'done' code
    0227 FF2C        601                   JMP  DS:DWORD PTR [SI]      ; close eyes and jump
                     602
                     603                   END

Errors=    0
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     350
    00C4 B9E803      351                  MOV  CX,$1000      ; DRAMs need delay after powerup
    00C7 E2FE        352 DRAMWT           LOOP DRAMWT        ; do nothing for a while
    00C9 8ED9        353                  MOV  DS,CX         ; point to DRAM
    00CB 8BF1        354                  MOV  SI,CX         ; point to DRAM
    00CD B90800      355                  MOV  CX,$08
    00D0 F3AC        356                  REP  LODSB         ; ensure 8 RAS cycles for DRAM
                     357                                     ; as per manufact. datasheet
    00D2 E90000      358                  JMP  SCANIN        ; go to command server
                     359
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     534 ;*************************************************************************
                     535 ;                    Intel Hex Downloader Routine                        *
                     536 ;*************************************************************************
    01AF 8B34        537 LOADER           MOV  SI,[SI]        ; SI ---> start of data area
    01B1 AC          538 LOADRP           LODSB               ; load first byte of next line
    01B2 3C2E        539                  CMP  AL,#2EH        ; check for end character "."
    01B4 7452        540                  JE   LOADDN         ; exit if load done
    01B6 3C3A        541                  CMP  AL,#3AH        ; first character should be ":"
    01B8 753E        542                  JNE  COLERR         ; go process error
                     543
    01BA E8D2FF      544                  CALL SUM_CHECK      ; process checksum of line
    01BD 7341        545                  JNC  SUMERR         ; process checksum error
                     546
    01BF 8A4405      547                  MOV  AL,DS:BYTE PTR 05[SI]  ; load record type
    01C2 2C30        548                  SUB  AL,#30H        ; ASCII correction
    01C4 7419        549                  JE   REC0           ; process record type 00
    01C6 FEC8        550                  DEC  AL
    01C8 7404        551                  JE   REC1OR3        ; process record type 01
    01CA FEC8        552                  DEC  AL
    01CC 7404        553                  JE   REC2           ; process record type 02
                     554
    01CE 8BF7        555 REC1OR3          MOV  SI,DI          ; point to next line
    01D0 EBDF        556                  JMP  LOADRP         ; process next line
                     557
    01D2 83C606      558 REC2             ADD  SI,#06         ; point SI to data area
    01D5 E8A4FF      559                  CALL FORM_ADDR      ; form load segment address
    01D8 268907      560                  MOV  ES:[BX],AX     ; store load segment in RAM
    01DB 8BF7        561                  MOV  SI,DI          ; point to next line
    01DD EBD2        562                  JMP  LOADRP         ; process next line
                     563
    01DF 57          564 REC0             PUSH DI             ; save next line address
    01E0 E899FF      565                  CALL FORM_ADDR      ; form offset address
    01E3 268B1F      566                  MOV  BX,ES:WORD PTR [BX]  ; fetch load seg
    01E6 8EC3        567                  MOV  ES,BX          ; current loader segment
    01E8 8BF8        568                  MOV  DI,AX          : ES:DI points to load dest.
    01EA 83C602      569                  ADD  SI,#02         ; point SI to data area
    01ED 8ACA        570                  MOV  CL,DL          ; CX contains byte xfer count
                     571
    01EF E870FF      572 XFERLP           CALL NEW_HEX_BYTE   ; form a data byte
    01F2 AA          573                  STOSB               ; load to RAM
    01F3 E2FA        574                  LOOP XFERLP         ; repeat until done
    01F5 5E          575                  POP  SI             ; restore next line pointer
    01F6 EBB9        576                  JMP  LOADRP         ; process next line
                     577
    01F8 C606050003  578 COLERR           MOV  DS:SCNACK,#03  ; return 'colon error'
    01FD E9D6FE      579                  JMP  SCANRP         ; resume scanning
                     580
    0200 C606050004  581 SUMERR           MOV  DS:SCNACK,#04  ; return 'checksum error'
    0205 E9CEFE      582                  JMP  SCANRP         ; resume scanning
                     583
    0208 C606050000  584 LOADDN           MOV  DS:SCNACK,#00  ; return 'done'
    020D E9C6FE      585                  JMP  SCANRP         ; resume scanning
                     586
```

```
LOCATION OBJECT CODE LINE    SOURCE LINE

                     197                 ORG  0F8650000H
                     198
                     199 HSTOUT          PROC FAR            ; pROBE host output routine
0000 06              200                 PUSH ES
0001 50              201                 PUSH AX             ; save char to be output
0002 B80080          202                 MOV  AX,SEG DUALPORT
0005 8EC0            203                 MOV  ES,AX          ; ES points to dualport RAM
0007 26F6060300      204 WAIT2           TEST ES:RXSEM,$01H  ; test output semaphore
000D 75F8            205                 JNZ  WAIT2          ; wait if previous char not read
000F 58              206                 POP  AX             ; restore char to be output
0010 26A20200        207                 MOV  ES:RXDAT,AL    ; write output character
0014 26C6060300      208                 MOV  ES:RXSEM,$01   ; set output semaphore
001A 07              209                 POP  ES
001B CB              210                 RET
                     211 ;*****************************************************************************
                     212                 ORG  0F8670000H
                     213
                     214 HSTSTS          PROC FAR            ; pROBE host status routine
0000 06              215                 PUSH ES
0001 B80080          216                 MOV  AX,SEG DUALPORT
0004 8EC0            217                 MOV  ES,AX          ; point ES to dualport RAM
0006 33C0            218                 XOR  AX,AX          ; indicate "no char"
0008 26F6060100      219                 TEST ES:TXSEM,$01H  ; check semaphore value
000E 740C            220                 JZ   OVER           ; exit if no character present
0010 FEC0            221                 INC  AL             ; indicate "character present"
0012 26803E0000      222                 CMP  ES:TXDAT,025   ; check if data is CTRL_Y
0018 7502            223                 JNE  OVER           ; exit if not CTRL_Y
001A FEC0            224                 INC  AL             ; exit with AL=2 if CTRL_Y
001C 07              225 OVER            POP  ES
001D CB              226                 RET
                     227 ;*****************************************************************************
                     228 IOMODE          PROC FAR            ; subroutine to determine
001E 52              229                 PUSH DX             ; if console is PC or RS232
001F BA56FF          230                 MOV  DX,Tim_0_Mode  ; point DX to timer 0 mode reg
0022 ED              231                 IN   AX,DX          ; load mode word
0023 C0D802          232                 RCR  AL,2           ; rotate ALT bit into carry
0026 5A              233                 POP  DX
0027 CB              234                 RET                 ; console flag is the CARRY bit
```

Advanced Micro Devices

Am79C30A LOW-LEVEL DEVICE DRIVER REFERENCE GUIDE

3

Publication #	Rev.	Amendment
10101	A	/0

Issue Date: 1987

Am79C30A LOW-LEVEL DEVICE DRIVER
REFERENCE GUIDE
TABLE OF CONTENTS

1. Distinctive Characteristics

The following list summarizes the main features of the Am79C30A DSC LLD. The DSC LLD has been implemented using 'C' to maximize portability and readability with a minimum effect upon performance. The DSC LLD provides support for all major DSC functions. These include the Line Interface Unit (LIU) and Data Link Controller (DLC) which are required to support AmLink LAPD. Other functions included support the Multiplexer (MUX) and the Main Audio Processor (MAP).

Table 1 - Summary of DSC LLD Features

- Written primarily in ANSI Compatible 'C'.
- Less than 5% written in Microsoft Assembler.
- Minimum operating system and processor dependencies.
- Supports all DSC major components:
  - Line Interface Unit (LIU).
  - Data Link Controller (DLC).
  - B-Channel Multiplexor (MUX).
  - Main Audio Processor (MAP).
- Mailbox interface to/from layer 2 (L2) and management entity (ME) routines.
- Queues up to 10 events to L2 and 10 events for ME.
- Compatible w/ AmLink (LAPD) layer 2 software.

For additional related information, the user may refer to any of the following documents:

Am79C30A Digital Subscriber Controller (DSC) Preliminary Data Sheet, PID # 09893, Rev. A, 10/87, Advanced Micro Devices.

ISDN AmLink Interface Preliminary Reference Guide, PID # 09529A, 6/87, Advanced Micro Devices.

2. General Description

This document describes the Low-Level Device Driver (LLD) for the Am79C30A Digital Subscriber Controller. The document includes a description of the programming interface to the DSC LLD services and a description of the implementation architecture.

2.1. Purpose

The DSC LLD is intended to be used as a general purpose example of DSC access procedures. The DSC LLD source code contains examples which illustrate how to use and access the many features of the Am79C30A DSC hardware.

The DSC LLD is intended for use with AmLink, AMDs LAPD implementation. The DSC LLD is used to support LAPD for the D-channel. The Am79C401 Integrated Data Protocol Controller (IDPC) LLD provides the same services for the B-channel. The interface provided by the IDPC and DSC LLDs use the same primitives so that both B-channel and D-channel can use the same layer 2 software. The IDPC and DSC LLDs provide a hardware independent interface to upper layer protocols such as LAPD.

2.2. System Requirements

The DSC LLD places relatively few requirements on the target system. The DSC LLD requires that the OS provide a method for requesting and returning memory buffers. No other OS services are required. The system requirements are summarized below:

- 16 KBytes of RAM/ROM for object code.
- 64 bytes of shared RAM for the configuration table.
- 256 bytes of shared RAM for mailboxes & initialization parameter blocks.
- Two or more shared memory buffers of MAXPACKSZ for data transfers.
- 512 bytes of RAM for the DSC LLD private use.
- Memory allocation service from the OS.
- Event handler routines.

2.3. Architecture

Communications with the DSC LLD by Layer 2 (L2) or Management Entity (ME) routines are performed via mailboxes. As shown in Figure 1, the DSC LLD uses four mailboxes. A pair of mailboxes is required for the L2 interface and a second pair for the ME interface. Each interface pair includes a command and an event mailbox. The mailbox structure is described in a later section.

The DSC performs the physical layer and part of the link layer protocols internally. The DSC LLD provides a well-defined software access to the DSC's services. The DSC LLD provides layer 2- services. The DSC LLD is designed to easily interface to any layer 2+ and management entity. The DSC LLD can be combined with the AmLink layer 2+ software to provide a complete Q.921 LAPD system.

The ME is a set of routines which are provided by the user to perform connection and inter-layer management functions. The ME functions and services are defined in the CCITT Q.940 recommendations. The ME routines are generally system dependent functions that are used to tie the various D-channel and B-channel protocols layers together in the user's system drivers. ISDN end-products are differentiated by their management entity implementations.

Command mailboxes are used to allow commands to be sent from the L2 or ME routines to the DSC LLD. Commands are loaded into the mailbox by the calling routine then control is passed to the routines in the DSC LLD that processes commands. The DSC LLD acknowledges the receipt of the command through the same mailbox.

Event mailboxes are used to send status information from the DSC LLD to the L2 or ME routines. Events are asynchronous messages passed to either the L2 or ME. Events are generated as the result of a DSC hardware interrupt. Events are placed on a queue prior to reporting. Separate queues exist for the ME and L2.

Event reporting may be enabled and disabled by issuing an EVENT ENABLE primitive which provides a method for enabling or disabling events to the queues. If a particular event occurs but is disabled, the event is not placed on the queue.

The user may issue an L2_EVENT_CTS or ME EVENT_CTS primitive with a DISABLE parameter to prevent queued events from being reported. This allows new events to be queued; however, it delays the reporting of events until the L2 EVENT CTS or ME EVENT_CTS are issued with an ENABLE parameter.

When enabled, the DSC LLD loads the event information into the proper mailbox then generates a signal which alerts the L2 or ME layer that an event has occurred and is available. The receiving routine, L2 or ME, acknowledges the event in the same mailbox.

The L2-LLD interface (mailbox pair) is used primarily for data transfer primitives. Most DSC LLD primitives are accessed via the ME-LLD interface. The ME-LLD interface is used to pass control and status information.

2.4. Target Environment

The DSC LLD can be used in either a single processor environment or a multi-processor system. In the single processor system, inter-layer communications are signalled using software interrupts. In a multi-processing system, hardware interrupts are used. In either case, the operation of the DSC LLD is the same. The interrupt handler used to process the mailbox message (command or event) is the same for both software and hardware based systems.

The DSC DLL has initially been implemented on an 80188/86 processor. The DSC LLD is implemented primarily in 'C' so that porting it to other types of processors is as easy as possible. The two primary processor dependencies are:

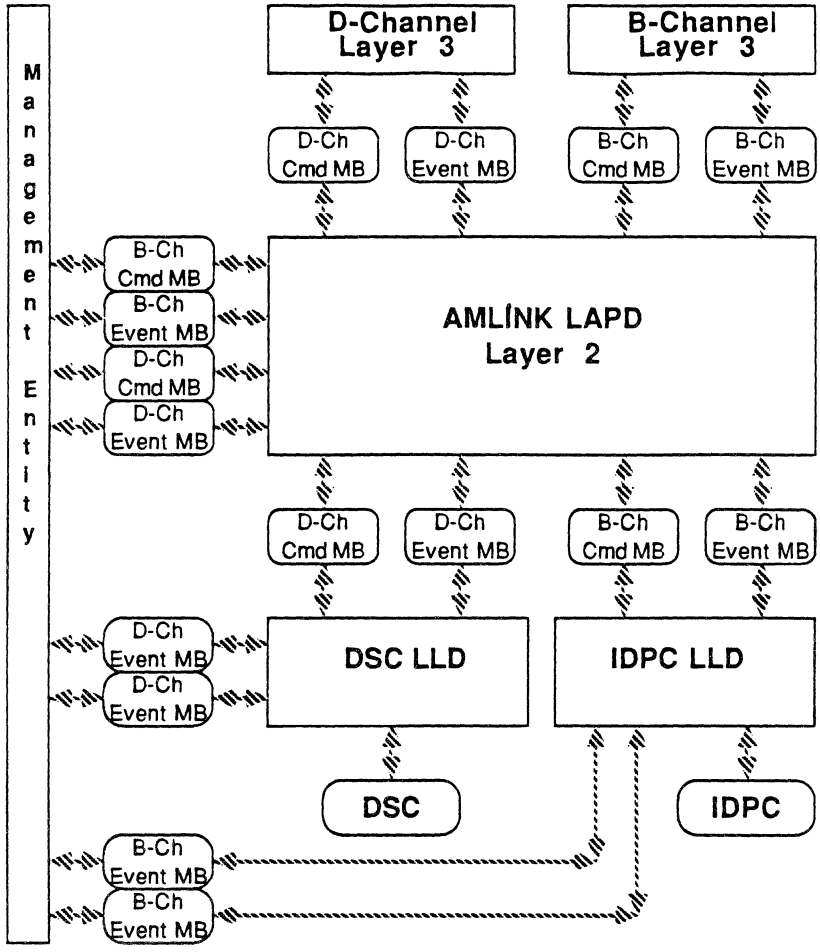
- Word (16-bits) and long word (32-bits) byte order
- Address segmentation

2.5. Development Environment

The DSC LLD is implemented using Microsoft 'C' Compiler Version 4.0 and the Microsoft Macro Assembler Version 4.0. Note that the byte order for all addresses specified in this document are in Microsoft 'C "FAR" format:

Lowest Address Byte:	Low-Order Byte	OFFSET
.	High-Order Byte	OFFSET
.	Low-Order Byte	SEGMENT
Highest Address Byte:	High-Order Byte	SEGMENT

FIGURE 1
INTER-LAYER MAILBOX INTERFACE



3. Functional Description

This section describes the DSC LLD interfaces and services which are accessible to the user. In general, this chapter describes how to use the DSC LLD. Refer to the chapter "Detailed Functional Description" for information regarding the actual implementation of the DSC LLD.

This chapter begins by describing the DSC LLD configuration block and related data structures used during initialization and operation. The remaining sections describe the command and event mailbox sequences used to access the DSC LLD services.

3.1. POR Configuration and Initialization

Prior to using the DSC LLD, the user is required to perform several initialization tasks. These are generally executed during the system Power-On Reset (POR) initialization sequence. In order to perform these tasks, the user must know the following information about the DSC LLD:

- Address of DSC LLD code
- Offset of POR initialization routine
- Offset of DSC HW interrupt handler
- Offset of Layer 2 command input handler
- Offset of Management Entity command input handler
- Structure of RAM Interface Block (RIB)

The DSC LLD code base address depends upon the user's system design. It may be located in firmware or in RAM. The DSC LLD is position independent.

In order to simplify the access of the DSC LLD hardware interrupt service routine and command input handlers, a set of entry points are provided. The location of this table will depend upon the order in which the DSC LLD has been linked. The table is located at the beginning of the isr.obj object code module. The table has the following structure:

Offset	Description
0	L2 Command Handler Subroutine
5	ME Command Handler Subroutine
A	L2 Command Handler Interrupt Service Routine
F	ME Command Handler Interrupt Service Routine
14	DSC Hardware Interrupt Service Routine
19	DSC Power-On Reset Initialization Subroutine

The DSC hardware interrupt handler routine address must be installed in the processor vector table prior to calling the DSC POR initialization routine.

Access to the Command handlers may be made using either a subroutine call or by issuing a software interrupt instruction. If a software interrupt is used, a vector to the command handler ISR must be installed for that software interrupt prior to issuing any DSC LLD commands.

The RAM Interface Block (RIB) is the structure through which the DSC LLD is accessed by either Layer 2 or the Management Entity. The DSC POR initialization routine installs default values into the RIB when it is called.

Using the above information, the user must perform or provide the following functions:

- Install the interrupt vectors for the DSC LLD.
- Provide routines to service L2 & ME events.
- Build or provide the DSC LLD configuration table.
- Execute the dsc_init() routine.

The user must install interrupt vectors for the DSC hardware interrupt handler. The user may also be required to install vectors to the the LLD command input handlers. This is required only if interrupts are to be used to pass control to the LLD command handlers.

L2/ME event input processes reside in the L2 (AmLink or other user-written layer 2) code and the user-written ME code. The user is required to provide routines which will be called by the LLD to service L2 or ME events. The user-provided event handler is to be a subroutine whose function is to pass the event message to the appropriate destination, the L2 or ME event input process. Additional processing may be performed in the user-supplied event handler prior to passing the event message to the L2 or ME.

The user is also required to service requests for memory allocation by the DSC LLD. The DSC LLD requests two buffers of MAXPACKSZ to support data reception. When a packet is received without an error, the DSC LLD returns the buffer to the L2 routine and requests a new buffer. The L2 or higher layers are responsible for the eventual deallocation of the buffer. If an error occurs, the DSC LLD re-uses the buffer.

The DSC LLD accesses the configuration table (see Table 2) via an internal pointer. In other words, the DSC LLD expects to find the pointer to the configuration table at a known address. The configuration table itself can be located in RAM or in firmware.

The dsc_init() routine uses information from the 64 byte user-supplied configuration table (see Table 2) to initialize the DSC hardware, install default values into the initialization parameter blocks, and initialize the DSC LLD Private RAM area.

After the dsc_init() routine is executed, the DSC LLD is in an idle state. Before using any of the DSC LLD services, the user must perform any remaining initialization via ME command and event primitives. This might include enabling the LIU receiver, requesting line activation, enabling the B-channel MUX, and starting MAP tones. This additional initialization is performed by using the DSC LLD primitives.

Table 2 - DSC LLD Configuration Table Format

Offset	Description	Size (bytes)
00	Addr of DSC LLD Private RAM.	4
04	Addr of LLD RAM Interface Block.	4
08	Addr of L2 Event Handler.	4
0C	Addr of ME Event Handler.	4
10	Addr of DSC Hardware.	4
14	Reserved	44

ADDRESS OF THE DSC LLD PRIVATE DATA RAM

This field contains the base address of the DSC LLD Private RAM area. This RAM is 512 bytes in length. This area is not required to reside in shared memory.

ADDRESS OF DSC LLD RAM INTERFACE BLOCK

This field contains the address of the shared RAM where the mailboxes and initialization parameter blocks are located. This area must be at least 256 bytes in length. The structure of the RAM Interface Block (RIB) is described in Table 3.

ADDRESS OF THE LAYER 2 EVENT HANDLER ROUTINE

This field contains a pointer to a user-supplied routine which is called by the DSC LLD to service an L2 event. No parameters are passed; and, the routine should return via a return from subroutine instruction.

ADDRESS OF THE MANAGEMENT ENTITY EVENT HANDLER ROUTINE

This field contains a pointer to a user-supplied routine which is called by the DSC LLD to service an ME event. No parameters are passed; and, the routine should return via a return from subroutine instruction.

ADDRESS OF DSC DEVICE HARDWARE

This field contains the base address of the Am79C30A DSC hardware (registers). The DSC LLD uses this pointer to access the Am79C30A device. Note that in I/O mapped systems, only the least significant 16-bits are used.

Table 3 - RAM Interface Block Structure

Offset	Description	Size (Bytes)
00	L2-LLD Command Mailbox	18
12	LLD-L2 Event Mailbox	18
24	ME-LLD Command Mailbox	18
36	LLD-ME Event Mailbox	18
48	DSC Initialization Parameter Block	8
50	DLC Initialization Parameter Block	32
70	MAP Initialization Parameter Block	64
B0	Reserved	80

LAYER 2 TO LLD COMMAND MAILBOX

This block of RAM is used as the L2-LLD mailbox which is used to pass commands from the Layer 2 protocol to the DSC LLD. The structure of this mailbox is described in Table 6. This mailbox is primarily used for D-channel data transmissions. The DSC LLD L2 command input handler services the commands passed in this mailbox.

LLD TO LAYER 2 EVENT MAILBOX

This block of RAM is used as the LLD_L2 event mailbox which is used to pass event information from the DSC LLD to Layer 2. Mailbox structures are described in Table 6. This mailbox is used primarily for receiving D-channel data.

MANAGEMENT ENTITY TO LLD COMMAND MAILBOX

This block of RAM is used as the ME-LLD mailbox which is used to pass commands from the Management Entity to the DSC LLD. Mailbox structures are described in Table 6. This

mailbox is used to pass commands for DSC LLD setup and initialization. The DSC LLD ME command input handler services the commands passed in this mailbox.

LLD TO MANAGEMENT ENTITY EVENT MAILBOX

This block of RAM is used as the LLD-ME mailbox which is used to pass event information from the DSC LLD to the management entity. Mailbox structures are described in Table 6. This mailbox is used primarily to pass DSC LLD status information back to the management routines.

DSC INITIALIZATION PARAMETER BLOCK

This field is not used.

DLC INITIALIZATION PARAMETER BLOCK

The data in this block (see Table 4) provides control information for the DSC DLC module. Default values are loaded and installed by the dsc_init() routine. The user may modify the values in the IPB; however, these are not installed until the user also calls the dlc_init() routine. The data in this block is generally not changed after the POR initialization sequence.

Table 4 - DLC Initialization Parameter Block Structure

Offset	Description	Size (Bytes)
00	Maximum Packet Size	2
02	L2 Address Length	1
03	L2 Address Select	1
04	CRC Check Enable	1
05	CRC Pass-Through Enable	1
06	CRC Generator Enable	1
07	Mark or Flag Idle Select	1
08	C/R Address Bit Ignore Enable	1
09	B-channel Select	1
0A	Invert Enable	1
0B	Minimum Packet Size	1
0C	Transmit FIFO Threshold	1
0D	Receive FIFO Threshold	1
0E	Auto-Receive Buffer Request	1
0F	Reserved	17

The DLC IPB contains fields that are not used by the Am79C30A DSC. The unused fields are used with the DLC on the Am79C401 IDPC, Am79C31A DEC, and/or Am79C313 M/S DEC. The default values for the fields which are used are:

- Max Packet Size (Default = 266)
- L2 Addr Length (Default = 2)
- L2 Addr Select (Default = 2nd)
- Receive FIFO Threshold (Default = 1)
- Transmit FIFO Threshold (Default = 8)
- CRC Pass-Through Enable (Default = Disable)
- Mark or Idle Fill (Default = Mark)
- C/R Bit Ignore Enable (Default = Disable)
- Auto-Buffer Request (Default = Enable)

MAXIMUM PACKET SIZE

This field indicates the larget packet size which can be received. If a packet is received larger than this size, the packet will be aborted and an overflow error reported via an ERR_STAT event.

L2 ADDRESS LENGTH

This field indicates that address recognition, if enabled, will recognise one or two byte addresses. If the contents are '1' then single byte address recognition is enabled; other wise, two byte address recognition is enabled.

L2 ADDRESS SELECT

This field is used to define which of byte is to be tested during single byte address recognition. If the contents are '1' then the first byte is used; otherwise, the second byte is used. This field is not used if two byte address recognition is enabled.

RECEIVE FIFO THRESHOLD

The contents of this field defines the DLC receive FIFO interrupt threshold. Valid values are 1, 2, 4, and 8. Any other value causes the threshold to be programmed to the default value of '1'. A value of '1' provides the largest possible receiver interrupt latency.

TRANSMIT FIFO THRESHOLD

The contents of this field defines the DLC transmitter FIFO interrupt threshold. Valid values are 1, 2, 4, and 8. Any other value causes the threshold to be programmed to the default value of '8'. A value of '8' provides the largest possible transmitter interrupt latency.

CRC PASS-THROUGH ENABLE

This field defines whether or not the FCS bytes are passed to the receive buffer. A value of '1' enables the pass-through; all other values disable pass-through.

MARK OR FLAG IDLE

This field defines whether MARK or FLAG idle is to be used for interframe fill. If this field is non-zero, MARK idle is used; otherwise, FLAG idle is used.

C/R BIT ENABLE

This field defines whether or not the C/R bit is to be used in address recognition. If the field is non-zero, the C/R bit is used in address recognition.

AUTO-RECEIVE BUFFER REQUEST ENABLE

This field defines whether the DSC LLD will request new receive buffers as needed. A value of '1' causes the DSC LLD to make requests for receive buffer space as needed using the BUFFSERV primitive. Any other value requires that buffers be manually posted by issung a READ_REQ primitive.

MAP INITIALIZATION PARAMETER BLOCK

This area is used to store the DSC MAP filter coefficients. The dsc_init() routine loads default values into this area. The MAP X and R filter coefficient are initially loaded with a set of coefficients which have been found to be useful for most handsets. Other default values provide 0dB gain on all filters and -18dB on Sidetone Gain.

Table 5 - MAP Initialization Parameter Block

Offset	Description	Size (Bytes)
00	X Filter Coefficients	16
10	R Filter Coefficients	16
20	GX Filter Coefficients	2
22	GER Filter Coefficients	2
24	Sidetone Gain Coefficients	2
25	Reserved	26

3.2. Mailbox Interfaces

As described earlier, the primary interface to the DSC LLD services is implemented using mailboxes. This section describes the procedure for using the mailboxes. Mailboxes consist of an 18 byte structure which contains the format shown in Table 6.

A mailbox is used to transfer commands and event information between the DSC LLD and either the L2 protocols or the ME routines. These routines may be in separate tasks or processes when using a multi-tasking operating system. This means that the mailboxes must be accessible to both the DSC LLD and L2/ME. In a single processor implementation, this is generally no problem; however, in a multi-processing system or a system with memory-management, the mailboxes must be placed in shared-memory.

Table 6 - Mailbox Structure

Offset	Component	Size (Bytes)
0	Command/Event Code	1
1	Receipt Code	1
2	Parameters	16

Command/Event Code

The first byte in a mailbox is the Command/Event Code. This byte determines what command is to be performed or what event has occurred. Each DSC LLD primitive is assigned a unique Command/Event Code. Table 8 lists a summary of the DSC LLD command codes; and, Table 9 provides a summary of the DSC LLD event codes.

Receipt Code

The second byte in a mailbox is the receipt code. This byte is used to indicate to the routine issuing the command that command has been received and validated.

Upon issuing a command/event, the issuing routine places the value 0xFF into the Receipt Code. The issuing routine then monitors the Receipt Code to determine if the command/event has been received. A value of '00' indicates that the command/event has been received or, for some commands, executed. Any other value indicates an error condition.

Table 7 - Valid Mailbox Receipt Codes

Code	Description
00	Command/event received or complete.
01	Illegal command/event.
02	Illegal parameter(s).
03	Transmitter is busy.
04	Transmitter is inactive.
05-10	Reserved.
11	Buffer request cannot be serviced.
12	Event overrun has occurred.
13-FE	Reserved.
FF	Command/event not received or complete.

Parameters

The last 16 bytes of a mailbox are reserved for parameters. The contents of these bytes are dependent upon the actual command or event issued.

3.3. Command Sequences

Commands are passed from either the Management Entity (ME) or the Layer 2 (L2) protocol to the DSC LLD. Figure 2 shows a typical command sequence. Note that the Receipt Code returned by the DSC LLD may indicate that the command has simply been received or that execution is complete. This depends upon the particular command issued.

Table 8 lists the valid command codes to which the DSC LLD will respond. Each command is described in detail in a later section.

Figure 2 - Typical Command Sequence

L2 or ME	LLD
o Write an 0xFF to the Receipt Code in the command mailbox.	
o Write the Command Code to the command mailbox.	
o Write any command parameters to the command mailbox.	
o Generate a command interrupt to the DSC LLD.	
-----> INTERRUPT -----> (or function call)	
o	Read the Command Code from the command mailbox.
o	Process the command using the Parameters from the command mailbox.
o	Write the appropriate Receipt Code to the command mailbox.
o	Execute a return from interrupt (or return from subroutine) instruction.
o When the Receipt Code in the command mailbox is not 0xFF, the sequence is complete.	

Table 8 - DSC LLD Command Codes Summary

Code (Hex)	Description	MB I/F	Module
00	Transmit a Buffer	L2	DLC
01	Initialize the DLC	ME	DLC
03	Update Addr Recognition	ME	DLC
04	Abort the Current Transmit	ME	DLC
05	Load a New Event Enables	ME	All
06	Begin Remote Loopback	ME	DLC
07	End Remote Loopback	ME	DLC
08	Begin Local Loopback	ME	DLC
09	End Local Loopback	ME	DLC
0A	Get a Random Number	ME	DLC
0B	Enable D-Channel Backoff	ME	DLC
0C	Post a Read Request	L2	DLC
80	Initialize the LIU	ME	LIU
81	Get Current LIU Status	ME	LIU
82	Get Hookswitch Status	ME	LIU
83	Activate 'S' I/F	ME	LIU
84	Set D-Channel Priority	ME	LIU
85	LIU B-Channel Control	ME	LIU
86	LIU Receiver Control	ME	LIU
90	Initialize the MAP	ME	MAP
91	Begin a Tone	ME	MAP
92	End a Tone	ME	MAP
93	Begin Digital Loopback	ME	MAP
94	End Digital Loopback	ME	MAP
95	MAP Output Control	ME	MAP
96	Load X-Filter Coefficients	ME	MAP
97	Load R-Filter Coefficients	ME	MAP
98	Load GX Coefficients	ME	MAP
99	Load GR Coefficients	ME	MAP
9A	Load GER Coefficients	ME	MAP
9B	Load SideTone Gain Coeff.	ME	MAP
A0	Set the Clock Mode	ME	INIT
A1	Set the Power-On Mode	ME	INIT
A8	Make a B-Channel Connection	ME	MUX
B0	ME Event Clear To Send	ME	All
B1	L2 Event Clear To Send	L2	All

3.4. Event Sequences

Events are messages passed from the DSC LLD back to either the L2 protocol or the ME. Event reporting can be disabled on an event by event basis using the LOAD_EVENT ENABLES primitive. In this case, disabled events are lost. Event reporting can be disabled using the L2_EVENT_CTS or the

ME_EVENT CTS primitives. These primitives prevent the LLD from reporting events but continues to queue up to 10 events each for the L2 and the ME. Event reporting can be enabled by issuing the L2 EVENT_CTS or ME_EVENT_CTS as appropriate.

Figure 3 shows a typical event sequence assuming that event reporting is enabled. Note that the Receipt Code returned by the L2 or ME may indicate that the command has simply been received or that some data is available. This depends upon the particular event issued.

Table 9 lists the valid event codes which the DSC LLD will generate. Each event is described in detail in a later section.

Figure 3 - Typical Event Sequence

L2 or ME	LLD
o	Write an 0xFF to the Receipt Code in the event mailbox.
o	Write the Event Code to the event mailbox.
o	Write any event parameters to the event mailbox.
o	Call the Event Generator routine for the target mailbox.
<----- INTERRUPT <----- (or function call)	
o Read the Event Code from the event mailbox.	
o Process the event using the Parameters from the event mailbox.	
o Write the appropriate Receipt Code to the event mailbox.	
o Execute a return from interrupt (or return from subroutine) instruction.	
o	When the Receipt Code in the event mailbox is not 0xFF, the sequence is complete.

Table 9 - DSC LLD Event Codes Summary

Code (Hex)	Description	MB I/F	Module
0	Transmission Complete	L2	DLC
1	Packet Received	L2	DLC
C	Buffer Allocation Request	ME	All
10	DLC Error	ME	DLC
20	LIU/HSW Status	ME	LIU

4. Programming

This section describes the command and event primitives that are used with the DSC LLD. This section is organized by DSC module (DLC, LIU, MAP, & MUX). Within each module subsection, all commands and events for that module are described.

All primitives, except those noted in Table 8 and 9, use the ME mailbox. The L2 mailbox is only used by a few DLC primitives.

Each primitive description contains the following information:

PRIMITIVE: Name of command/event.

CODE: Command or event code for the primitive.

MAILBOX: Mailbox to be used with this primitive.

INPUTS: Input parameters for this primitive. These are identified as mailbox parameter bytes 0 to 15.

OUTPUTS: Output parameters for this primitive. These are identified as mailbox parameter bytes 0 to 15.

RECEIPT CODES: Possible Receive Codes for this primitive.

DESCRIPTION: Describes the functions and services provided by this primitive.

NOTES: Describes any special considerations related to this primitive.

4.1. Line Interface Unit (LIU)

This section describes the primitives used to access the LIU services. Primarily, the LIU services deal with activation on the 'S' interface and hookswitch status.

Commands	Description
LIU_INIT	Initialize the LIU.
GET_LIU_STATE	Get the current activation state.
GET_HSW_STATE	Get the current hookswitch state.
REQ_LIU_ACTIVATION	Activate the 'S' I/F.
ENABLE_LIU	Enable/disable the LIU receiver.
SET_DCH_PRIORITY	Set the D-channel priority.
ENABLE_BCH	Enable/disable the LIU B-channels.

Events	Description
LIU_STATUS	LIU &/or HSW change of state.

PRIMITIVE: LIU_INIT

COMMAND CODE: 0x80

MAILBOX: ME

INPUTS: None.

OUTPUTS: None.

RECEIPT CODES:
00 = Initialization is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: Upon exit the LIU will be in the idle mode with the B1 & B2 channels and the receiver disabled.

NOTES: To achieve line activation, the user should issue the ENABLE_LIU and REQ_LIU_ACTIVATION primitives. Notification of activation state changes will be given via LLD to ME events.

PRIMITIVE: GET_LIU_STATE

COMMAND CODE: 0x81

MAILBOX: ME

INPUTS: None

OUTPUTS:

status	Parameter byte 0 reflects the current LIU activation state plus 2 as read from the DSC LSR. This value reflects the current F-state of the LIU.

RECEIPT CODES:

00 = Command is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive returns the current LIU 'S' interface F-state.

NOTES:

PRIMITIVE: GET_HSW_STATE

COMMAND CODE: 0x82

MAILBOX: ME

INPUTS: None

OUTPUTS: Status — Parameter byte 0 reflects the current state of the HSW pin. A 0x00 indicates a LOW signal and a 0x01 indicates a HIGH signal on the HSW pin.

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive returns the current state of the DSC HSW pin.

NOTES:

PRIMITIVE: REQ_LIU_ACTIVATION

COMMAND CODE: 0x83

MAILBOX: ME

INPUTS: Activate/Deactivate (parameter byte 0) - Activate = 1; Deactivate = Not 1

OUTPUTS: None

RECEIPT CODES:

00 = Command has been received.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet received.

DESCRIPTION: This primitive allows the user to request that the LIU attempt to establish line activation. This primitive is equivalent to the PH-ACTIVE request primitive defined by CCITT.

The deactivate option does not remove deactivation since a TE may not request deactivation. If the NT causes a deactivation, the DSC will normally immediately request re-activation. If the deactivate option is used, the DSC will not request re-activation. The deactivate option should not be called until after activation is achieved.

NOTES: The transition from the present activation state to the request state will be reported using the LIU_STATUS event.

This primitive should only be issued when the LIU in state F3, F7, or F7 and the LIU receiver has been enabled for a minimum of 250 mS.

PRIMITIVE: SET_DCH_PRIORITY

COMMAND CODE: 0x84

MAILBOX: ME

INPUTS: Priority (parameter byte 0) - Contains a four-bit priority (0 - 0xF). Only the least significant nibble is used.

OUTPUTS: None

RECEIPT CODES:
00 = Initialization is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This primitive sets the D-channel priority level. This is used in passive bus configurations.

NOTES: The default value is ZERO.

PRIMITIVE: ENABLE_BCH

COMMAND CODE: 0x85

MAILBOX: ME

INPUTS: B1 (parameter byte 0) -
Enabled = 1; Disabled = Not 1

B2 (parameter byte 1) -
Enabled = 1; Disabled = Not 1

OUTPUTS: None

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This primitive allows the user to enable or disable either or both of the LIU B-channels.

NOTES:

PRIMITIVE: ENABLE_LIU

COMMAND CODE: 0x85

MAILBOX: ME

INPUTS: Enable/Disable (parameter byte 0) -
Enabled = 1; Disabled = Not 1

OUTPUTS: None

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This primitive allows the user to enable or disable the LIU receiver. If the receiver is disabled, 'S' interface line activation will be lost.

NOTES:

PRIMITIVE: LIU_STATUS

EVENT CODE: 0x20

MAILBOX: ME

INPUTS: STATUS (parameter byte 0) - A ONE in bit 0 indicates that the LIU has changed state. A ONE in bit 1, indicates that the HSW pin has changed state.

F-STATE (parameter byte 1) - Contains the F-state value as of the last LIU change of state.

HSW-STATE (parameter byte 2) - Contains the value of the HSW pin as of the last HSW change of state.

OUTPUTS: None

RECEIPT CODES:

00 = Event has been received.
01 = Illegal event.
02 = Illegal Parameter.
FF = Event not yet received.

DESCRIPTION: This primitive is used to notify the ME that an LIU and/or hookswitch change of state has occurred. This event also returns the current LIU F-state and HSW pin level.

NOTES: The LIU_STAT primitive is the equivalent to the PH-STATUS indication primitive.

4.2. Data Link Controller (DLC)

This section describes the primitives available to access the DLC services. These primitives are compatible with those used by AmLink LAPD and those provided by the IDPC LLD.

Commands	Description
DLC_INIT	Initialize the DLC.
XMIT_BUFFER	Transmit a buffer.
READ_REQUEST	Request a packet read.
UPDATE_ADDR_RECOGNITION	Update address recognition parameters.
XMIT_ABORT	Abort the current buffer transmission.
BEGIN_REMOTE_LOOP	Begin remote loopback.
END_REMOTE_LOOP	End remote loopback.
BEGIN_LOCAL_LOOP	Begin local loopback.
END_LOCAL_LOOP	End local loopback.
GET_RANDOM_NUM	Get a random number.
D_CH_BACKOFF	Enable/disable D-Channel backoff.

Events	Description
PACKET_RCVD	Packet received w/o error.
XMIT_DONE	Buffer transmitted w/o error.
ERROR_STATUS	DLC error status.

PRIMITIVE: DLC_INIT

COMMAND CODE: 1

MAILBOX: ME

INPUTS: None, uses the DLC IPB in the RIB.

OUTPUTS: None.

RECEIPT CODES:
00 = Initialization is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive places the DSC DLC into a known state. The primitive installs the DLC IPB parameters, disables address recognition, gets two buffers for the DLC receiver, and enables all DLC interrupts.

NOTES: When this primitive is issued, any DLC operation in-progress is aborted. Normally, the parameters in the DLC IPB are not changed after the original initialization.

PRIMITIVE: XMIT_BUFFER

COMMAND CODE: 0

MAILBOX: L2

INPUTS:
Buffer Address (parameter bytes 0-3)
Packet Length (parameter bytes 4-5)
Buffer Length (parameter bytes 6-7)
Buffer Reference Number (parameter bytes 8-9)

OUTPUTS: None

RECEIPT CODES:
00 = Command was received.
01 = Illegal command.
02 = Illegal parameter.
03 = Transmitter busy.
04 = Transmitter not available.
FF = Command not yet received.

DESCRIPTION:
This primitive initiates a buffer transmission on the D-channel. This primitive is equivalent to the PH-DATA request primitive.

The DSC LLD does not use the 'Packet Length' parameter. The DSC LLD will transmit a packet of 'Buffer Length'.

If this primitive is issued while the transmitter is busy from a previously issued, but unfinished, XMIT_BUFF, a Transmitter Busy receipt code is returned. The user may re-issue the primitive after the current transmission is complete as indicated by a XMIT_DONE.

If an XMIT_BUFF primitive is issued and either the LIU is not active or the DLC is not in a local loopback mode, a Transmitter Inactive Receipt Code is issued and the command is ignored.

NOTES:

PRIMITIVE: READ_REQUEST

COMMAND CODE: 0x0C

MAILBOX: L2

INPUTS: Buffer Address (parameter bytes 0-3)
Buffer Reference Number (parameter bytes 4-5)

OUTPUTS: None

RECEIPT CODES:
00 = Command was received.
01 = Illegal command.
05 = Receive buffer queue is full.
FF = Command not yet received.

DESCRIPTION: This primitive allows receive buffers to be manually posted to the DLC. By default, the DLC is configured to automatically request new receive buffers as they are needed. If the DLC IPB has been initialized to disable the auto-receive buffer requests, this primitive must be used to post a buffer in which to receive a packet.

NOTES: If ME event reporting has been disabled via the ME_CTS primitive, auto-receive buffer requests can be lost. If this occurs, the LLD may be stuck waiting for a buffer which will never arrive. For this reason, it is important that the system be "tuned" to prevent ME queue overflows or READ_REQ primitives be used to manually post receive buffers instead of enabling auto-receive buffer requests.

PRIMITIVE: UPDATE ADDR_RECOGNITION

COMMAND CODE: 0x03

MAILBOX: ME

INPUTS: Enable/Disable (parameter byte 0) - A ZERO indicates disable address recognition for the specified address recognition register.

Address Register Number (parameter byte 1) - Identifies address recognition register 0, 1, 2, or 3 to be affected by this command.

Address (parameter bytes 2-3) - Value to be loaded into the specified address recognition register. If single byte addresses are enabled in the DLC IPB, parameter byte 2 contains that address regardless of which address byte (1st or 2nd) is enabled.

OUTPUTS: None

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This primitive allows the DLC address recognition services to be used. Address recognition can be used as a bandpass filter, allowing only packets with the specified addresses to be received.

Three types of address recognition can be used. The first is a two byte address. The second is single byte recognition, 1st byte only. The third is single byte recognition, 2nd byte only. The DSC DLC supports up to three unique addresses.

The DLC IPB contains fields which define the length of the address to be used. For single addresses, the DLC IPB also contains a field which defines which address is to be used.

NOTES:

PRIMITIVE: XMIT_ABORT

COMMAND CODE: 4

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive causes any buffer transmission that was previously issued to be aborted. An XMITDONE event with a non-zero Status parameter is sent to the L2 following an aborted transmission.

NOTES:

PRIMITIVE: BEGIN_REMOTE_LOOP

COMMAND CODE: 6

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES:

00 = Command is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive places the D-channel on the LIU into a remote loopback (echo) mode. Information arriving at the LIU receiver is immediately sent on the LIU transmitter.

NOTES:

PRIMITIVE: END_REMOTE_LOOP

COMMAND CODE: 7

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive removes the D-channel remote loopback on the LIU if enabled. If the remote loopback is not enabled, the command does nothing.

NOTES:

PRIMITIVE: BEGIN_LOCAL_LOOP

COMMAND CODE: 8

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES:

00 = Command is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive places the D-channel into a local loopback mode. In this mode, data sent for transmission by the local processor is returned to the local processor on the DLC receiver.

NOTES:

PRIMITIVE: END_LOCAL_LOOP

COMMAND CODE: 9

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive removes the D-channel local loopback, if enabled.

NOTES:

PRIMITIVE: GET_RANDOM_NUMBER

COMMAND CODE: 0x0a

MAILBOX: ME

INPUTS: None

OUTPUTS: Random Number (parameter byte 0 & 1)

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive returns a 16-bit psuedo-random number. The number is generated from a free running counter within the DSC.

NOTES:

PRIMITIVE: D_CH_BACKOFF

COMMAND CODE: 0x0b

MAILBOX: ME

INPUTS: None

OUTPUTS: MODE -
Enable = 1; Disable = Not 1

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This primitive allows packet transmission which detect a D-channel collision to abort transmission (backoff enabled) or to continue transmitting (backoff disabled).

NOTES:

PRIMITIVE: XMIT_DONE

EVENT CODE: 0

MAILBOX: L2

INPUTS: Status (parameter bytes 0-1)

OUTPUTS: None

RECEIPT CODES:
00 = Event has been received.
01 = Illegal event.
FF = Event not yet recognized.

DESCRIPTION: This primitive is used to notify the L2 code that the last buffer transmission command issued has completed. The Status parameter indicates whether the transmission was successful or not (0 = successful, non-zero = aborted).

An XMIT DONE is also issued as the result of a D-Channel collision or a Transmit Underrun error or as the result of an XMIT_ABORT primitive.

NOTES: The XMITDONE primitive is equivalent to the CCITT PH-DATA Response primitive.

PRIMITIVE: PACKET_RCVD

EVENT CODE: 1

MAILBOX: L2

INPUTS: Buffer Address (parameter bytes 0-3)

Packet Length (parameter bytes 4-5)

Buffer Reference Number (parameter bytes 6-7)

OUTPUTS: None

RECEIPT CODES:

00 = Event has been received.
01 = Illegal command.
02 = Illegal parameter.
FF = Event not yet received.

DESCRIPTION: This primitive is used to notify the L2 code that a packet has been successfully received.

The DSC LLD is double-buffered. This means that the DSC LLD can accept two packets before losing a packet. After receiving the first packet, the DSC LLD will request a new buffer from the ME. If a new buffer is not received prior to the beginning of a third packet, that packet will be lost.

A BUFFSERV event is issued for each RCVDONE to replace the buffer passed to the L2. This BUFFSERV event should be serviced prior to the need to remove a data for a third packet from the receive FIFO. If this has not occurred, that third packet will be lost.

NOTES: The PACKET_RCVD primitive is equivalent to the CCITT PH-DATA Indication primitive.

PRIMITIVE: ERROR_STATUS

EVENT CODE: 0x10

MAILBOX: ME

INPUTS: ERROR (parameter byte 0) - A ONE in any bit position indicates that error occurred:

Bit 0: Abort Sequence Received.
1: Non-Integer Number of Bytes Received.
2: D-Channel Collision Occurred.
3: FCS Error on Received Packet.
4: 'Long' Packet Received.
5: 'Short' Packet Received.
6: Receiver Overrun Occurred.
7: Transmitter Underrun Occurred.

OUTPUTS: None

RECEIPT CODES:
00 = Event has been received.
01 = Illegal event.
02 = Illegal Parameter.
FF = Event not yet received.

DESCRIPTION: This primitive is used to notify the ME that a D-channel error has occurred. If a Receiver error is detected, the DSC LLD will abort the rest of the packet. If a D-channel transmitter error is detected, the DSC LLD will also send an XMIT DONE event with a non-zero Status parameter to the L2.

NOTES: The ERR_STAT primitive is the equivalent to a combination of the PH-ERROR primitive.

4.3. Multiplexor (MUX)

The DSC LLD supports up to three B-channel connections using the DSC MUX. These allow a number of different connections within the DSC.

Commands	Description
ALLOCATE_BCH_CONNECT	Allocate a B-channel connection via the MUX.

PRIMITIVE: ALLOCATE_BCH_CONNECT

COMMAND CODE: 0xA8

MAILBOX: ME

INPUTS: Connection (parameter byte 0) - Corresponds to one of the three MUX control registers (0 to 2)

Side #1 (parameter byte 1) -
- 0x00 = Deallocate channel.
- 0x01 = B1 on 'S' I/F
- 0x02 = B2 on 'S' I/F
- 0x03 = MAP (Audio port)
- 0x04 = Bb from MPI
- 0x05 = Bc from MPI
- 0x06 = SBP channel 1
- 0x07 = SBP channel 2
- 0x08 = SBP channel 3

Side #2 (parameter byte 2) -
- 0x00 = Deallocate channel.
- 0x01 = B1 on 'S' I/F
- 0x02 = B2 on 'S' I/F
- 0x03 = MAP (Audio port)
- 0x04 = Bb from MPI
- 0x05 = Bc from MPI
- 0x06 = SBP channel 1
- 0x07 = SBP channel 2
- 0x08 = SBP channel 3

OUTPUTS: None

RECEIPT CODES:
- 00 = Command is complete.
- 01 = Illegal command.
- 02 = Illegal parameter.
- FF = Command not yet complete.

DESCRIPTION: This primitive allows the user to make B-channel connections to the MAP, Serial Bus Port (SBP), or to the processor interface.

NOTES: For this primitive to be effective on the 'S' interface, the ENABLE_BCH primitive should be issued.

4.4. Main Audio Processor (MAP)

The MAP commands are used to allow the user to control call progress, ringing, and DTMF tones. The MAP commands also allows the user the ability to perform analog or digital loopback within the MAP.

Commands	Description
MAP_INIT	Initialize the MAP.
BEGIN_TONE	Begin a tone.
END_TONE	End a tone.
BEGIN_DIGITAL_LOOP	Begin digital loopback.
END_DIGITAL_LOOP	End digital loopback.
MAP_OUTPUT_CTRL	MAP audio output control.
MAP_OUTPUT_GAIN	MAP audio output gain.
LOAD_X_FILTER COEFFICIENTS	Load X-Filter Coefficients.
LOAD_R_FILTER COEFFICIENTS	Load R-Filter Coefficients.
LOAD_GX GAIN COEFFICIENTS	Load GX Gain Coefficients.
LOAD_GR GAIN COEFFICIENTS	Load GR Gain Coefficients.
LOAD_GER GAIN COEFFICIENTS	Load GER Gain Coefficients.
LOAD_STG GAIN COEFFICIENTS	Load Side Tone Gain Coefficients.

PRIMITIVE: MAP_INIT

COMMAND CODE: 0x90

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES:
00 = Initialization is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive installs the MAP IPB and initializes the MAP hardware in the DSC. Upon exit, the MAP call-progress, DTMF, and ringing tones are disabled. The MAP high-pass filter and ADC auto-zero functions are enabled.

The default MAP filter coefficients are sufficient for most handsets using a carbon-based microphone.

NOTES:

PRIMITIVE: BEGIN_TONE

COMMAND CODE: 0x91

MAILBOX: ME

INPUTS:

Command (parameter byte 0-1) -
0 = Ringing tone.
1 = Single frequency tone.
2 = DTMF tone.

Frequency #1 (parameter byte 2-3) -
(in Hz)

Amplitude Gain #1 (parameter byte 4-5) -
(in absolute value dB, from -18 to 0 dB in 2 dB steps)

Frequency #2 (parameter byte 6-7) -
(in Hz)

Amplitude Gain #2 (parameter byte 8-9) -
(in absolute value dB, from -18 to 0 dB in 2 dB steps)

OUTPUTS: None

RECEIPT CODES:

00 = Command is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This primitive starts a ringing, single, or DTMF tone with the specified frequencies and amplitudes. The output is directed to the MAP audio output port: LS1/LS2 or EAR1/EAR2.

The specified frequency(ies) must correspond to those listed in the Am79C30A data sheet. Any other values will return with an Illegal Parameter Receipt Code.

NOTES:

PRIMITIVE: END_TONE

COMMAND CODE: 0x92

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive turns-off the tone generators if enabled.

NOTES:

PRIMITIVE: BEGIN_DIGITAL_LOOP

COMMAND CODE: 0x93

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive places the MAP into a digital loopback. In other words, data routed to the MAP via the MUX is not output but is returned to the MUX.

NOTES: In order to perform an analog loopback for the MAP, use the ALLOCATE_BCH_CONNECT primitive to connect the MAP input to the MAP output.

PRIMITIVE: END_DIGITAL_LOOP

COMMAND CODE: 0x94

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive disables the MAP digital loopback.

NOTES:

PRIMITIVE: MAP_OUTPUT_CTRL

COMMAND CODE: 0x95

MAILBOX: ME

INPUTS: Ear/LS (parameter byte 0) -
- 0 = Ear output enabled
- 1 = Loudspeaker output enabled

OUTPUTS: None

RECEIPT CODES:
00 = Initialization is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This primitive allows the user to specify the MAP audio output port. The default port is the EAR1/EAR2 port.

NOTES:

PRIMITIVE: LOAD X-FILTER COEFFICIENTS

COMMAND CODE: 0x96

MAILBOX: ME

INPUTS: COEFFICIENTS (parameter bytes 0-15) -

OUTPUTS: None

RECEIPT CODES:
00 = Command Complete
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive allows the user to load the X-Filter coefficients. The coefficients are organized as: h0 LSB, h0 MSB, h1 LSB ... h7 MSB.

NOTES:

PRIMITIVE: LOAD R-FILTER COEFFICIENTS

COMMAND CODE: 0x97

MAILBOX: ME

INPUTS: COEFFICIENTS (parameter bytes 0-15) -

OUTPUTS: None

RECEIPT CODES:
00 = Command Complete
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive allows the user to load the R-Filter coefficients. The coefficients are organized as: h0 LSB, h0 MSB, h1 LSB ... h7 MSB.

NOTES:

PRIMITIVE: LOAD GX GAIN COEFFICIENTS

COMMAND CODE: 0x98

MAILBOX: ME

INPUTS: COEFFICIENTS (parameter bytes 0-1) -
Parameter Byte 0 - LSB
PArameter Byte 1 - MSB

OUTPUTS: None

RECEIPT CODES:
00 = Initialization is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive allows the user to load the GX gain coefficients.

NOTES:

PRIMITIVE: LOAD GR GAIN COEFFICIENTS

COMMAND CODE: 0x99

MAILBOX: ME

INPUTS: COEFFICIENTS (parameter bytes 0-1) -
Parameter Byte 0 - LSB
Parameter Byte 1 - MSB

OUTPUTS: None

RECEIPT CODES:
00 = Initialization is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive allows the user to load the GR gain coefficients.

NOTES:

PRIMITIVE: LOAD GER GAIN COEFFICIENTS

COMMAND CODE: 0x9A

MAILBOX: ME

INPUTS: COEFFICIENTS (parameter bytes 0-1) -
Parameter Byte 0 - LSB
Parameter Byte 1 - MSB

OUTPUTS: None

RECEIPT CODES:
00 = Initialization is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive allows the user to load the GER gain coefficients.

NOTES:

PRIMITIVE: LOAD STG GAIN COEFFICIENTS

COMMAND CODE: 0x9B

MAILBOX: ME

INPUTS: COEFFICIENTS (parameter bytes 0-1) -
Parameter Byte 0 - LSB
Parameter Byte 1 - MSB

OUTPUTS: None

RECEIPT CODES: 00 = Initialization is complete.
01 = Illegal command.
FF = Command not yet complete.

DESCRIPTION: This primitive allows the user to load the STG gain coefficients.

NOTES:

4.5. Misc Service Requests

This section describes several additional commands which are available in the DSC LLD. The DSC POR initialization routine is included in this section. The POR Initialization routine is not a mailbox primitive but a simple subroutine call. Others dealing with the buffer allocation/deallocation request, updating the event mask, and setting the DSC clock & power modes are mailbox primitives.

Commands	Description
UP_EVENT_MASK	Update the event mask.
SET_CLK_MODE	Set the DSC clock mode.
SET_PWR_MODE	Set the DSC power mode.
LOAD_EVENT_ENABLES	Enable/Disable specific events.
L2_EVENT_CLEAR TO SEND	Enable/Disable L2 event reporting.
ME_EVENT_CLEAR TO SEND	Enable/Disable ME event reporting.

Events	Description
BUFF_SERVICE	Buffer allocation request.

PRIMITIVE: DSC LLD Initialization

COMMAND CODE: Not Applicable

MAILBOX: Not Applicable

INPUTS: None, uses Configuration Table information.

OUTPUTS: None

RECEIPT CODES: 00 = Initialization is complete.

DESCRIPTION: This routine is used to initialize the DSC hardware, IPBs, and Private RAM area. This routine must be called prior to using any DSC LLD mailbox services. Normally, the routine should be part of the system POR initialization sequence.

After the dsc_init() routine is executed, all DSC registers are loaded with '00'. This value disables all functions. The DSC Private RAM area is cleared. The DLC & MAP IPBs are loaded with their default values.

The DLC will request new DLC receive buffers. The DLC will not deallocate its old receive buffers. This is the responsibility of the management entity.

NOTES: The user should issue the DLC_INIT, MAP_INIT, and LIU_INIT primitives to install the IPBs into the DSC hardware. The user may modify the IPB values prior to issuing the appropriate initialization primitive.

PRIMITIVE: BUFF_SERVICE

EVENT CODE: 0x0C

MAILBOX: ME

INPUTS: MODE (parameter byte 0) - A ZERO indicates that an allocation is requested.

SIZE (parameter byte 1 & 2) - The size in bytes of the buffer requested.

OUTPUTS: ADDR (parameter bytes 3-6) - Contains the base address of the buffer requested.

REFNO (parameter bytes 7-8) - Contains the reference number of the buffer provided.

RECEIPT CODES:
00 = Initialization is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This primitive is used to request that a buffer be allocated to the DSC LLD. This primitive is used to obtain a buffer for the DLC receiver. The deallocate option is not used by the DSC LLD. The REFNO is used when passing a buffer back to the ME or to the L2 code.

NOTES: The ME must be able to supply a buffer of at least MAXPACKETSZ as defined in the DLC IPB.

PRIMITIVE: LOAD_EVENT_ENABLES

COMMAND CODE: 5

MAILBOX: ME

INPUTS: Event Enables (parameter bytes 0-1) -

Byte 0, bit 0	-	Enable XMIT_DONE Events.
Byte 0, bit 1	-	Enable PACKET_RCVD Event.
Byte 0, bit 2	-	Enable BUFF_SERV Events.
Byte 0, bit 3	-	Enable LIU_STATUS Events.
Byte 0, bit 4	-	Enable HSW_STATUS Events.
Byte 1, bit 0	-	Enable Abort Rvcd Events.
Byte 1, bit 1	-	Enable Non-Integer # of Bytes Rcvd Events.
Byte 1, bit 2	-	Enable D-Ch. Collision Events.
Byte 1, bit 3	-	Enable FCS Error Events.
Byte 1, bit 4	-	Enable 'Long Packet' Rcvd Error Events.
Byte 1, bit 5	-	Enable 'Short Packet' Rcvd Error Events.
Byte 1, bit 6	-	Enable Receiver Overrun Error Events.
Byte 1, bit 7	-	Enable Transmitter Underrun Error Events.

OUTPUTS: None.

RECEIPT CODES:

00 = Update is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This primitive causes a new event mask to be used by the DSC LLD. The event mask allows event reporting to be individually enabled or disabled.

NOTES: Events may still occur even though they are not reported.

PRIMITIVE: SET_CLOCK_MODE

COMMAND CODE: 0xA0

MAILBOX: ME

INPUTS: Selector (parameter byte 0) -
- 0 = divide by 2
- 1 = divide by 1
- 2 = divide by 4
- 3 = divide by 3

OUTPUTS: None

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This command sets the DSC clock mode.

NOTES: The DSC will power up in the divide by two mode.

PRIMITIVE: SET_PWR_MODE

COMMAND CODE: 0xA1

MAILBOX: ME

INPUTS: Mode (parameter byte 0) -
- 0 = Idle
- 1 = Active
- 2 = Active without the MAP

OUTPUTS: None

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This primitive sets the DSC power mode.

NOTES: The DSC is in the Idle mode after a reset.

PRIMITIVE: L2_EVENT_CTS

COMMAND CODE: 0xB1

MAILBOX: L2

INPUTS:
Mode (parameter byte 0) -
- 0 = Disable
- 1 = Enable

OUTPUTS:
Status (parameter byte 1) -
- 0 = Empty Queue
- -1 = Queue overrun has occurred

RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This primitive allows event reporting from the LLD to the L2 to be controlled. When this primitive is issued with a DISABLE parameter, the LLD will continue to queue events but will not report them to the L2 until the primitive is re-issued with the ENABLE parameter.

NOTES:

PRIMITIVE: ME_EVENT_CTS

COMMAND CODE: 0xB0

MAILBOX: ME

INPUTS: Mode (parameter byte 0) -
- 0 = Disable
- 1 = Enable

OUTPUTS: Status (parameter byte 1) -
- 0 = Empty Queue
- -1 = Queue overrun has occurred

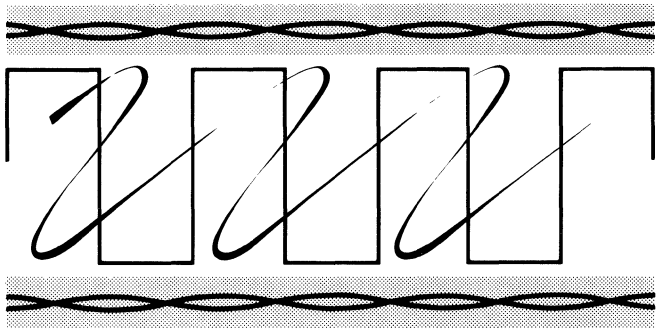
RECEIPT CODES:
00 = Command is complete.
01 = Illegal command.
02 = Illegal parameter.
FF = Command not yet complete.

DESCRIPTION: This primitive allows event reporting from the LLD to the ME to be controlled. When this primitive is issued with a DISABLE parameter, the LLD will continue to queue events but will not report them to the L2 until the primitive is re-issued with the ENABLE parameter.

NOTES:

DLC
Low-Level Device Driver

Reference Guide



I S D N

ADVANCED MICRO DEVICES

Publication # 09783 | Rev. A | Amendment /0
Issue Date: May 1988

TABLE OF CONTENTS

1. DISTINCTIVE CHARACTERISTICS

This document describes the user interface to the Low-Level Device Driver (LLD) for the Am79C401 Integrated Data Protocol Controller (IDPC) Data Link Controller (DLC). The IDPC DLC LLD has been implemented using the 'C' programming language to maximize portability and readability with a minimum effect upon performance.

Table 1 - Summary of IDPC DLC LLD Features

- Written primarily in ANSI 'C'.
- Less than 5% written in Microsoft 8088 Macro Assembler.
- Minimum operating system and processor dependencies.
- Uses 80188/80186 DMA.
- Supports optional logging to a file.
- Interrupt-driven mailbox interface to/from Layer 2 (L2) and Management Entity (ME) routines.
- Compatible with AmLink™ (LAPD/LAPB) Layer 2 software.

2. GENERAL DESCRIPTION

This document describes the user interface to the Low-Level Device Driver (LLD) for the Am79C401 Integrated Data Protocol Controller (IDPC) Data Link Controller (DLC).

2.1 Purpose

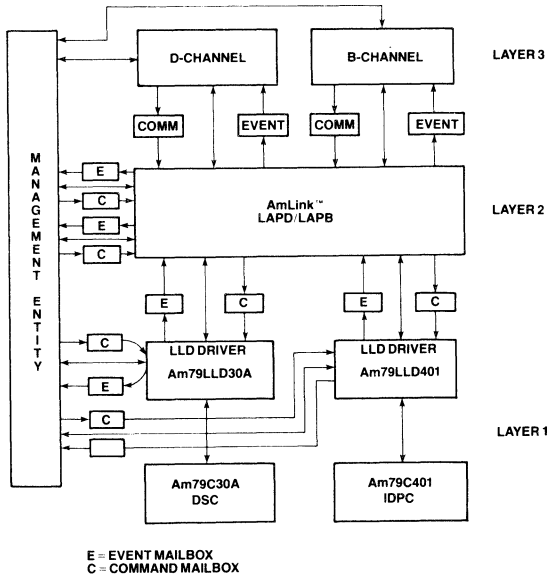
The IDPC DLC LLD is intended to be used as a general purpose example of IDPC DLC programming. The IDPC DLC LLD source code contains examples illustrating how to use and access the many features of the Am79C401 IDPC DLC hardware.

The IDPC DLC LLD can be used with any bit-oriented protocol (BOP) including AmLink™, AMD's LAPD/LAPB implementation. In Integrated Services Digital Network (ISDN) applications, the IDPC DLC LLD is used to support the packet protocol for the B-Channel. (The Am79C30A Digital Subscriber Controller (DSC) LLD provides the same services for the D-Channel.) The interfaces provided by the DSC and IDPC DLC LLDs use the same primitives so that both D-Channel and B-Channel can use the same Layer 2 software. The DSC and IDPC DLC LLDs provide a hardware independent interface to upper layer protocols such as LAPD.

2.2 System Requirements

The IDPC DLC LLD places relatively few requirements on the target system. The IDPC DLC LLD requires that the Operating System (OS) provide a method for requesting and returning memory buffers. No other OS services are required. The system requirements are summarized below:

- 4 kBytes of RAM/ROM for object code.
- 64 bytes of shared RAM for the configuration table.
- 256 bytes of shared RAM for mailboxes and initialization parameter blocks.
- One or more shared memory buffers of MAXPACKSZ (Maximum Packet Size) for data transfers.
- 128 bytes of stack RAM.
- Memory allocation service from the OS.
- Event interrupt generation routines. 80188/80186 DMA hardware (both channels).



E = EVENT MAILBOX
C = COMMAND MAILBOX

Figure 1. Inter-Layer Mailbox Interface

2.3 Architecture

Communications with the IDPC DLC LLD by Layer 2 (L2) or Management Entity (ME) routines are performed via mailboxes and interrupts. As shown in Figure 1, the IDPC DLC LLD includes four mailboxes. One pair of mailboxes is required for the L2 interface and a second pair is required for the ME interface. Each interface pair includes a command and an event mailbox. The mailbox structure is described in a later section.

The ME is a set of routines which are provided by the user to perform connection and layer management functions. These are generally system dependent functions that are used to tie the various D-Channel and B-Channel protocol layers together in the user's system drivers.

Command mailboxes are used to allow commands to be sent from the L2 or ME routines to the IDPC DLC LLD. Commands are loaded into the mailbox by the calling routine an interrupt is then generated to inform the IDPC DLC LLD that a command is available for processing. The IDPC DLC LLD acknowledges the receipt of the command through the same mailbox.

Event mailboxes are used to send status information from the IDPC DLC LLD to the L2 or ME routines. The IDPC DLC LLD loads the event information into the proper mailbox then, generates an interrupt which alerts the L2 or ME layer that an event has occurred and is available. The receiving routine, L2 or ME, acknowledges the event in the same mailbox.

The L2-DLC LLD interface (mailbox pair) is used primarily for data transfer primitives. The ME-DLC LLD interface is used to pass control and status information.

2.4 Target Environment

The IDPC DLC LLD can be used in either a single processor environment or a multi-processor system. In the single processor system, inter-layer communications are signaled using software interrupts. In a multi-processing system, hardware interrupts are used. In either case, the operation of the IDPC DLC LLD is the same. The interrupt handler used to process the mailbox message (command or event) is the same for both software and hardware based systems.

The IDPC LLD is initially implemented on an 80188/86 processor, using the 'C' programming language to enhance portability. The two primary processor dependencies are:

- Word (16-bits) and long word (32-bits) byte order
- Address segmentation

2.5 Development Environment

The IDPC DLC LLD is implemented using Microsoft 'C' Compiler Version 4.0 and the Microsoft Macro Assembler Version 5.0. Note that the byte order for all addresses specified in this document are in Microsoft 'C' "FAR" format:

Lowest Address Byte:	Low-Order Byte	OFFSET
	High-Order Byte	OFFSET
	Low-Order Byte	SEGMENT
Highest Address Byte:	High-Order Byte	SEGMENT

3. FUNCTIONAL DESCRIPTION

This section shows the IDPC DLC LLD interfaces and services accessible to the user and describes how to use the IDPC DLC LLD.

3.1 POR Configuration and Initialization

Prior to using the IDPC DLC LLD, several initialization tasks must be performed. These are generally executed during the system Power-On Reset (POR) initialization sequence. In order to perform these tasks, the user must know the following information about the IDPC DLC LLD:

- Address of IDPC DLC LLD code.
- Offset of POR initialization routine.
- Offset of IDPC DLC hardware interrupt handler.
- Offset of 80188 DMA hardware interrupt handler.
- Offset of Layer 2 command input handler.
- Offset of Management Entity command input handler.
- Structure of RAM Interface Block (RIB).

The IDPC DLC LLD code base address depends upon the user's system design. It may be located in firmware or in RAM. The IDPC DLC LLD is position independent.

The IDPC POR initialization routine (referred to as illdinit()) is located at offset TBD from the base address of the IDPC DLC LLD code. This routine must be called to initialize the IDPC DLC LLD.

The IDPC DLC hardware interrupt handler is located at offset TBD from the IDPC DLC LLD code base address. This address must be installed in the processor vector table prior to calling the IDPC POR initialization routine.

The 80188 DMA hardware interrupt handler is located at offset TBD from the IDPC DLC LLD code base address. This address of this routine must be installed in the processor vector table prior to calling the IDPC POR initialization routine.

The Layer 2 command input handler is located at offset TBD from the IDPC DLC LLD code base address. This address of this routine must be installed in the processor vector table prior to calling the IDPC POR initialization routine.

The Management Entity input command handler is located at offset TBD from the IDPC DLC LLD code base address. This address of this routine must be installed in the processor vector table prior to calling the IDPC POR initialization routine.

The RAM Interface Block (RIB) is the structure through which the IDPC DLC LLD is accessed by either Layer 2 or the Management Entity. The IDPC POR initialization routine installs default values into the RIB when it is called.

Using the above information, the user must perform or provide the following functions:

- Install the interrupt vectors for the IDPC DLC LLD.
- Provide routines to generate L2 and ME event interrupts.
- Build or provide the IDPC DLC LLD configuration table.
- Execute the illdinit( ) routine.

The user must install interrupt vectors for the IDPC DLC hardware interrupt handler, the 80188/86 DMA hardware interrupt handler, L2 & ME command input handlers, and L2 and ME event handlers. The L2/ME event handlers reside in the L2 (AmLink™ or other user-written Layer 2) code and the user-written ME code.

The user is required to provide routines which will be called by the DLC LLD to generate the interrupts (software or hardware) for L2 or ME events. Each routine should be in the form of a subroutine which passes no parameters. This allows the IDPC DLC LLD to be independent of the type of interrupt used to generate the event interrupt.

The user is also required to service requests for memory allocation by the IDPC DLC LLD. The IDPC DLC LLD requests memory of size specified by the Maximum Packet Size field (in the DLC Initialization Parameter Block (IPB)) to support data reception. When a packet is received without an error, the IDPC DLC LLD passes the buffer to Layer 2; if a packet is received with error, the LLD reuses the buffer for the next packet reception.

The IDPC DLC LLD accesses the configuration table (see Table 2) via a pointer to the configuration table located at the known address TBD. The configuration table itself can be located in RAM or in firmware.

The 'illdinit()' routine uses information from the 64 byte user-supplied configuration table (see Table 2) to initialize the IDPC hardware, install default values into the IDPC DLC Initialization Parameter Block, and initialize the IDPC DLC LLD Private RAM area.

After the 'illdinit()' routine is executed, the IDPC DLC LLD is in an idle state. Before transmitting or receiving any packets, the user must execute the DLC Init and DLC Control ME Command primitives.

Table 2. IDPC DLC LLD Configuration Table Format

Offset	Description	Size (bytes)
00	Addr of IDPC DLC LLD Private RAM	4
04	Addr of DLC LLD RAM Interface Block	4
08	Addr of L2 Event Generator	4
0C	Addr of ME Event Generator	4
10	Addr of IDPC DLC Hardware Registers	4
14	Addr of 80188/86 DMA Hardware Registers	4
18	Reserved	40

3.1.1 Address of the IDPC DLC Private Data RAM

This RAM is 64 bytes in length. This area is not required to reside in shared memory.

3.1.2 Address of IDPC DLC LLD RAM Interface Block

This field contains the address of the shared RAM where the mailboxes and DLC Initialization Parameter Block are located. This area must be at least 256 bytes in length. The structure of the RAM Interface Block (RIB) is described in Table 3.

3.1.3 Address of the LAYER 2 Interrupt Generator Routine

This field contains a pointer to a user-supplied routine which is called by the IDPC DLC LLD to generate an L2 event interrupt. No parameters are passed; the routine should return via a return from subroutine instruction. It is required that this routine return with all processor registers in the same state as when the routine was called.

3.1.4 Address of the MANAGEMENT ENTITY Interrupt Generator Routine

This field contains a pointer to a user-supplied routine which is called by the IDPC DLC LLD to generate an ME event interrupt. No parameters are passed; the routine should return via a return from subroutine instruction. It is required that this routine return with all processor registers in the same state as when the routine was called.

3.1.5 Address of IDPC DLC Device Hardware

This field contains the base address of the Am79C401 IDPC DLC hardware (registers). The IDPC DLC LLD uses this pointer to access the Am79C401 device.

3.1.6 Address of 80188/86 DMA Device Hardware

This field contains the base address of the 80188/86 DMA hardware (registers). The IDPC DLC LLD uses this pointer to access the DMA device.

Table 3. RAM Interface Block Structure

Offset	Description	Size (Bytes)
00	L2-DLC LLD Command Mailbox	18
12	DLC LLD -L2 Event Mailbox	18
24	ME-DLC LLD Command Mailbox	18
36	DLC LLD -ME Event Mailbox	18
48	Reserved	8
50	DLC Initialization Parameter Block	14
5E	Reserved	162

3.1.7 LAYER 2 to LLD Command Mailbox

This block of RAM is used as the L2-DLC LLD mailbox to pass commands from the Layer 2 protocol to the IDPC DLC LLD. This mailbox is primarily used for B-Channel data transmissions. The IDPC DLC LLD L2 command input handler services the commands passed in this mailbox.

3.1.8 LLD to LAYER 2 Event Mailbox

This block of RAM is used as the DLC LLD L2 event mailbox to pass event information from the IDPC DLC LLD to Layer 2. Mailbox structures are described in Table 5. This mailbox is used primarily for receiving B-Channel data.

3.1.9 MANAGEMENT ENTITY to LLD Command Mailbox

This block of RAM is used as the ME-DLC LLD mailbox to pass commands from the Management Entity to the IDPC DLC LLD. Commands are passed for IDPC DLC LLD setup and initialization. The IDPC DLC LLD ME command input handler services the commands passed in this mailbox.

3.1.10 LLD to MANAGEMENT ENTITY Event Mailbox

This block of RAM is used as the DLC LLD-ME mailbox which is used to pass event information from the IDPC DLC LLD to the management entity. This mailbox is used primarily to pass IDPC DLC LLD status information back to the management routines.

3.1.11 DLC Initialization Parameter Block

The data in this block (see Table 4) provides control information for the IDPC DLC module. Default values are loaded and installed by the 'illdinit()' routine. The user may modify the values in the IPB; however, these are not installed until the user executes the DLC Init Command.

The L2 Address Length, L2 Address Select and C/R Address Bit Ignore Enable fields are also used by the LLD during execution of the Update Address Recognition Command.

Table 4. DLC Initialization Parameter Block Structure

Offset	Description	Size (Bytes)
00	Maximum Packet Size	2
02	L2 Address Length	1
03	L2 Address Select	1
04	CRC Check Enable	1
05	CRC Pass-Through Enable	1
06	CRC Generator Enable	1
07	Mark or Flag Idle Select	1
08	C/R Address Bit Ignore Enable	1
09	B-Channel Select	1
0A	Invert Enable	1
0B	Minimum Packet Size	1
0C	Transmit FIFO Threshold	1
0D	Receive FIFO Threshold	1

MAXIMUM PACKET SIZE - Maximum length packet (including CRC bytes, if any) that is legal for the Layer 2+ protocol in use.

LAYER 2 ADDRESS LENGTH - Number of bytes in the packet address field for the Layer 2+ protocol in use.

SINGLE ADDRESS BYTE SELECT - Selects which packet address byte to compare during address recognition when the Layer 2 Address Length is equal to one. Set this parameter to 1 for first packet address byte select, 2 for second byte select.

CRC CHECK ENABLE - Set to 1 to enable CRC checking during packet reception, 0 for CRC Check Disable.

CRC PASS ENABLE - Set to 1 for pass CRC bytes to Layer 2+ as last two bytes of each received packet. Set to 0 to disable passing any CRC bytes.

CRC GENERATE ENABLE - Set to 1 to enable CRC Generation during packet transmission, 0 for CRC Generate Disable.

MARK IDLE/FLAG IDLE SELECT - Set to 1 to generate mark idle pattern (all 1 bits) when not transmitting a packet, 0 to generate flag idle pattern.

IGNORE C/R ADDRESS BIT ENABLE - Set to 1 to ignore the C/R bit in the Layer 2 packet address during address recognition. Set to 0 to also compare the C/R bit during address recognition.

B-CHANNEL SELECT - Set to a value from 0 to 30 (decimal) to select multiplexed B-Channel 0 through 30 respectively. Set to 31 (decimal) to select non-multiplexed operation (e.g. SNA).

INVERT ENABLE - Set to 1 to enable inversion of the transmitted and received serial bit streams. Set to 0 to disable inversion.

MINIMUM PACKET SIZE - Minimum length packet (including CRC bytes, if any) that is legal for the Layer 2+ protocol in use.

TRANSMIT FIFO THRESHOLD - Set to a value from 0 to 15 (decimal) to set the fullness threshold (0 to 15 bytes) at which the DLC Transmitter requests service from the LLD software or DMA for additional bytes to be loaded into the transmit FIFO.

RECEIVE FIFO THRESHOLD - Set to a value 0, 1, 2 . . . 15 (decimal) to set the fullness threshold (32, 2, 4 . . . 30 bytes) at which the DLC Receiver requests service from the LLD software or DMA for bytes to be unloaded from the receive FIFO.

3.2 Mailbox Interfaces

As described earlier, the primary interface to the IDPC DLC LLD services is implemented using mailboxes. This section describes the procedure for using the mailboxes which consist of an 18-byte structure containing the format shown in Table 5.

A mailbox is used to transfer commands and event information between the IDPC DLC LLD and either the L2 protocols or the ME routines. These routines may be in separate tasks or processes when using a multi-tasking operating system. This means that the mailboxes must be accessible to both the IDPC DLC LLD and L2/ME. In a single processor implementation, this is generally no problem; however, in a multi-processing system or a system with memory-management, the mailboxes must be placed in shared-memory.

Table 5. Mailbox Structure

Offset	Component	Size (Bytes)
0	Command/Event Code	1
1	Receipt Code	1
2	Parameters	16

3.2.1 Command/Event Code

The first byte in a mailbox is the Command/Event Code. This byte determines what command is to be performed or what event has occurred. Each IDPC DLC LLD primitive is assigned a unique Command/Event Code. Table 7 lists the IDPC DLC LLD command codes and Table 8 provides a summary of the IDPC DLC LLD event codes.

3.2.2 Receipt Code

The second byte in a mailbox is the receipt code. This byte indicates to the routine issuing the command that command has been received and validated.

Upon issuing a command/event, the routine places the value 0xFF into the Receipt Code. The issuing routine then monitors the Receipt Code to determine if the command/event has been received. A value of '00' indicates that the command/event has been received or, for some commands, executed. Any other value indicates an error condition.

Table 6. Valid Mailbox Receipt Codes

Code	Description
00	Command/event received or complete.
01	Illegal command/event.
02	Illegal parameter(s).
03-FE	Reserved.
FF	Command/event not received or complete.

3.2.3 Parameters

The last 16 bytes of a mailbox are reserved for parameters. The contents of these bytes are dependent upon the actual command or event issued.

3.3 Command Sequences

Commands are passed from either the Management Entity (ME) or the Layer 2 (L2) protocol to the IDPC DLC LLD. Figure 2 shows a typical command sequence. Note that the Receipt Code returned by the IDPC DLC LLD may indicate that the command has simply been received or that execution is complete. This depends upon the particular command issued.

Table 7 lists the valid command codes to which the IDPC DLC LLD will respond. Each command is described in detail in a later section.

Figure 2. Typical Command Sequence

L2 or ME	LLD
• Write an 0xFF to the Receipt Code in the command mailbox.	
• Write the Command Code to the command mailbox.	
• Write any command parameters to the command mailbox.	
• Generate a command interrupt to the IDPC DLC LLD.	
---› INTERRUPT ---›	
•	Read the Command Code from the command mailbox.
•	Process the command using the Parameters from the command mailbox.
•	Write the appropriate Receipt Code to the command mailbox.
•	Execute a return from interrupt instruction.
• When the Receipt Code in the command mailbox is not 0xFF, the sequence is complete.	

Table 7. IDPC DLC LLD Command Codes Summary

Code (Hex)	Description	MB I/F	Module
00	Transmit a Buffer	L2	DLC
01	Initialize the DLC	ME	DLC
02	DLC Control	ME	DLC
03	Update Address Recognition	ME	DLC
04	Abort the Current Transmit	ME	DLC
05	Load a New Event Enables	ME	DLC
06	Begin Remote Loopback	ME	DLC
07	End Remote Loopback	ME	DLC
08	Begin Local Loopback	ME	DLC
09	End Local Loopback	ME	DLC

3.4 Event Sequences

Events are messages passed from the IDPC DLC LLD back to either the L2 protocol or the ME. Figure 3 shows a typical event sequence. Note that the Receipt Code returned by the L2 or ME may indicate that the command has simply been received or that some data is available. This depends upon the particular event issued.

Table 8 lists the valid event codes which the IDPC DLC LLD will generate. Each event is described in detail in a later section.

Figure 3. Typical Event Sequence

L2 or ME	LLD
•	• Write an 0xFF to the Receipt Code in the event mailbox.
•	• Write the Event Code to the event mailbox.
•	• Write any event parameters to the event mailbox.
•	• Call the Event Generator routine for the target mailbox.
‹– – – INTERRUPT ‹– – –	
• Read the Event Code from the event mailbox.	
• Process the event using the Parameters from the event mailbox.	
• Write the appropriate Receipt Code to the event mailbox.	
• Execute a return from interrupt instruction.	
•	• When the Receipt Code in the event mailbox is not 0xFF, the sequence is complete.

Table 8. IDPC DLC LLD Event Codes Summary

Code (Hex)	Description	MB I/F	Module
0	Transmission Complete	L2	DLC
1	Packet Received	L2	DLC
B	Error Status	ME	DLC
C	Buffer Allocation Request	ME	DLC

4. PROGRAMMING

This section describes the command and event primitives used with the IDPC DLC LLD. These primitives are compatible with those used by AmLinkTM LAPD/LAPB and those provided by the DSC LLD.

Each primitive description contains the following information:

PRIMITIVE CODE:	Command or event code for the primitive.
MAILBOX:	Mailbox to be used with this primitive.
INPUTS:	Input parameters for this primitive. These are identified as mailbox parameter bytes 0 to 15.
OUTPUTS:	Output parameters for this primitive. These are identified as mailbox parameter bytes 0 to 15.
RECEIPT CODES:	Possible Receive Codes for this primitive.
DESCRIPTION:	Describes the functions and services provided by this primitive.
NOTES:	Describes any special considerations related to this primitive.

The command primitives are:

Commands	Description
XMITBUF	Transmit a buffer.
DLC_INIT	Initialize the DLC.
DLC_CONTROL	Enable/disable DLC transmitterand/orreceiver.
UPDATE ADDR_RECOG()	Update address recognition parameters.
XMIT_ABORT	Abort the current buffer transmission.
LOAD EVENT ENABLES	Load event reporting enable/disable bit array.
BEGIN-REMOTE_LOOP	Begin remote Loopback.
END_REMOTE_LOOP	End remote Loopback.
BEGIN_LOCAL_LOOP	Begin local Loopback.
END_LOCAL_LOOP	End local Loopback.

The event primitives are:

Events	Description
PACKET_RCVD	Packet received without error.
XMIT_DONE	Buffer transmitted without error.
ERROR STATUS	A valid address or an end-of address has been received.
BUFFER SERVICE	Buffer allocation/deallocation service.

4.1 PRIMITIVE: **XMITBUF**

COMMAND CODE: 0

MAILBOX: L2

INPUTS: Buffer Address (parameter bytes 0-3)

Packet Length (parameter bytes 4-5)

Buffer Length (parameter bytes 6-7)

Buffer Reference Number (parameter bytes 8-9)

OUTPUTS: None

RECEIPT CODES: 00 = Command was received.

01 = Illegal command.

02 = Illegal parameter.

03 = Transmitter busy.

FF = Command not yet received.

DESCRIPTION: This primitive initiates a buffer transmission via the IDPC DLC. If the buffer size is larger than the packet size, the IDPC DLC LLD will automatically send multiple packets until the entire buffer is sent. The buffer size MUST be an integral multiple of the packet size or an illegal parameter receipt code will be returned.

This primitive is equivalent to the PH-DATA request primitive.

If this primitive is issued while the transmitter is busy from a previously issued, but unfinished, XMITBUF, a Transmitter Busy receipt code is returned. The user may re-issue the primitive after the current transmission is complete as indicated by an XMITDONE event.

NOTES:

4.2 PRIMITIVE: **DLC_INIT**

COMMAND 1 CODE:

MAILBOX: ME

INPUTS: None, uses the DLC Initialization Parameter Block (IPB) in the LLD RAM Interface Block.

OUTPUTS: None.

RECEIPT CODES: 00 = Initialization is complete.

01 = Illegal command.

FF = Command not yet complete.

DESCRIPTION: This primitive places the IDPC DLC into a known state. The primitive installs the DLC IPB parameters, disables address recognition, gets a buffer for the DLC receiver, enables 80188/86 receive DMA Channel 0, and enables all DLC interrupts.

NOTES:

4.3 PRIMITIVE: DLC Control

COMMAND CODE: 0x02

MAILBOX: ME

INPUTS: DLC Transmitter Enable/Disable (parameter byte 0) Enable = 1; Disable = 0

DLC Receiver Enable/Disable (parameter byte 1) Enable = 1; Disable = 0

OUTPUTS: None

RECEIPT CODES: 00 = Command is complete.

01 = Illegal command.

02 = Illegal parameter.

FF = Command not yet complete.

DESCRIPTION: This primitive allows the user to enable or disable the DLC transmitter and/or receiver.

NOTES:

4.4 PRIMITIVE: UP_ADDR_RECOGNITION

COMMAND CODE: 0x03

MAILBOX: ME

INPUTS: Parameter byte 0 - Enable (= 1)/Disable (= 0) address recognition for the specified Address Register Number (parameter byte 1).

Parameter byte 1 - Address Register Number. Parameter byte 1 = 0, 1, 2, 3 for Address Recognition Register 0, 1, 2, or 3. Parameter byte 1 = 4 for Broadcast Address Recognition. Parameter bytes 2-3 - Address. Contents to be loaded into the specified address recognition register. If single byte addresses are enabled in the DLC IPB, parameter byte 2 contains that address regardless of which address byte (1st or 2nd) is enabled. Parameter bytes 2-3 are ignored if parameter byte 1 = 4 (Broadcast Address Recognition).

OUTPUTS: None

RECEIPT CODES: 00 = Command is complete.

01 = Illegal command.

02 = Illegal parameter.

FF = Command not yet complete.

DESCRIPTION: This primitive allows the DLC address recognition services to be used. Address recognition can be used as a bandpass filter, allowing only packets with the specified addresses to be received.

The IDPC DLC supports up to four programmable addresses plus a fixed Broadcast Address (all 1's address).

The DLC IPB contains the following fields which condition address recognition:

Address length (single byte or two bytes)

Single byte address select (1st or 2nd)

Ignore C/R address bit

NOTES: See section on DLC IPB for further information.

4.5 PRIMITIVE: TRANSMIT ABORT

COMMAND CODE: 4

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES: 00 = Command is complete.

01 = Illegal command.

FF = Command not yet complete.

DESCRIPTION: This primitive causes any buffer transmission that was previously issued to be aborted.

Execution of this primitive causes the IDPC DLC LLD to issue an XMITDONE event primitive to the Layer 2.

NOTES:

4.6 PRIMITIVE: LOAD_EVENT_ENABLES

COMMAND CODE: 5

MAILBOX: ME

INPUTS: Event Enables. This is an eight byte array. Each bit in the array represents one of 64 events. The bit position for a particular event mask corresponds to the event code for the event. A ONE in a bit position enables the corresponding event to be reported via interrupt to the layer (L2 or ME) appropriate for the particular event. A ZERO in a bit position disables that event from being reported. For instance, the event mask bit for the "Error Status" event (Event Code 0x0B) is byte #1, bit #3.

OUTPUTS: None.

RECEIPT CODES: 00 = Update is complete.

01 = Illegal command.

02 = Illegal parameter.

FF = Command not yet complete.

DESCRIPTION: This primitive causes a new event mask to be used by the IDPC DLC LLD. The event mask allows event reporting to be individually enabled or disabled.

NOTES: Events may still occur even though they are not reported if event reporting is disabled.

4.7 PRIMITIVE: BEGIN_REMOTE_LOOP

COMMAND CODE: 6

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES: 00 = Command is complete.

01 = Illegal command.

FF = Command not yet complete.

DESCRIPTION: This primitive places the IDPC DLC in Remote Loopback. While Remote Loopback is enabled, all received packets will be looped back to the far end transmitter. Received packets will also be received by the local DLC if the DLC Receiver is enabled. Any packets transmitted by the local DLC Transmitter will not be transmitted but will be thrown in the bit bucket.

NOTES:

4.8 PRIMITIVE: END_REMOTE_LOOP

COMMAND CODE: 7

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES: 00 = Command is complete.

01 = Illegal command.

FF = Command not yet complete.

DESCRIPTION: This primitive disables IDPC DLC Remote Loopback. If the Remote Loopback is not enabled, the command does nothing.

NOTES:

4.9 PRIMITIVE: BEGIN_LOCAL_LOOP

COMMAND CODE: 8

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES: 00 = Command is complete.

01 = Illegal command.

FF = Command not yet complete.

DESCRIPTION: This primitive places the IDPC DLC into a Local Loopback mode. In this mode, data transmitted by the IDPC DLC is looped back to the IDPC DLC receiver.

NOTES:

4.10 PRIMITIVE: **END_LOCAL_LOOP**

COMMAND CODE: 9

MAILBOX: ME

INPUTS: None

OUTPUTS: None

RECEIPT CODES: 00 = Command is complete.

01 = Illegal command.

FF = Command not yet complete.

DESCRIPTION: This primitive disables IDPC DLC Local Loopback, if enabled.

NOTES:

4.11 PRIMITIVE: **XMITDONE**

EVENT CODE: 0

MAILBOX: L2

INPUTS: None

OUTPUTS: None

RECEIPT CODES: 00 = Event has been received.

01 = Illegal event.

FF = Event not yet recognized.

DESCRIPTION: This primitive is used to notify the L2 code that the last Transmit Buffer LLD Command issued has been completed and the IDPC DLC transmitter is ready to transmit another buffer.

This event is issued both when normal buffer transmission is finished, when the Transmit Underrun event occurs or when the Transmit Abort command is executed.

NOTES:

4.12 PRIMITIVE: **PACKET_RCVD**

EVENT CODE: 1

MAILBOX: L2

INPUTS: Buffer Address (parameter bytes 0-3)

Packet Length (parameter bytes 4-5)

Buffer Reference Number (parameter bytes 6-7)

OUTPUTS: None

RECEIPT CODES: 00 = Event has been received.

01 = Illegal command.

02 = Illegal parameter. FF = Event not yet received.

DESCRIPTION: This primitive is used to notify the L2 code that a packet has been successfully received.

NOTES: The user should acknowledge this primitive with as little delay as possible. This routine will request a new receiver buffer immediately after the event is acknowledged. If acknowledgment is delayed too long, a receiver overrun error may occur.

4.13 PRIMITIVE: ERROR STATUS

EVENT CODE: 0x0B

MAILBOX: ME

INPUTS: None

OUTPUTS: Parameter byte 0 -

Bit 0 - Receiver abort condition.
1 - Receiver non-integer # of bytes.
2 - Reserved.
3 - Receiver CRC error.
4 - Receiver Long Packet error.
5 - Receiver Short Packet error.
6 - Receiver overrun error.
7 - Transmitter Underrun error.

RECEIPT CODES: 00 = Event has been received.

01 = Illegal event.

FF = Event not yet received.

DESCRIPTION: This primitive is used to notify the ME that one or more DLC error or exceptional conditions has occurred.

All of the receiver error/exceptions are mutually exclusive for the IDPC DLC receiver. In other words, only one of bits 0-6 of parameter byte 0 may be set in any single occurrence of this event. However, the Transmitter Underrun error bit may be set simultaneously with one of the receive error/exception bits.

When one of the received error/exception conditions occurs, Layer 2 is not notified. If the Transmitter Underrun condition occurs, a Transmit Done event to Layer 2 is executed in addition to the Error Status event.

NOTES: The Non-Integer Number of Bytes condition indicates that the last byte of a received packet contains less than 8 bits.

The Long Packet error occurs when the number of bytes in a received packet exceeds the Maximum Packet Size field of the DLC IPB.

The Short Packet error occurs when the number of bytes in a received packet is less than the Minimum Packet Size field of the DLC IPB.

4.14 PRIMITIVE: IDPC DLC LLD Initialization

COMMAND CODE: Not Applicable

MAILBOX: Not Applicable

INPUTS: None, uses Configuration Table information.

OUTPUTS: None

RECEIPT CODES: 00 = Initialization is complete.

DESCRIPTION: This routine 'illdinit()' is used to initialize the IDPC DLC Initialization Parameter Block (IPB) and IDPC DLC LLD Private RAM area. This routine must be called prior to using any IDPC DLC LLD mailbox services. Normally, the routine should be part of the system POR initialization sequence.

NOTES: The user should issue the DLC INIT command primitive to install the DLC IPB into the IDPC DLC hardware. The user may modify the IPB fields prior to issuing the DLC Init command.
Once the DLC Init command has been executed, the DLC CONTROL command must be executed to enable the IDPC transmitter and receiver.

4.15 PRIMITIVE: BUFF_SERVICE

EVENT CODE: 0x0C

MAILBOX: ME

INPUTS: MODE (parameter byte 0) - A ZERO indicates that an allocation is requested.

SIZE (parameter byte 1 and 2) - The size in bytes of the buffer requested.

OUTPUTS: ADDR (parameter bytes 3-6) - Contains the base address of the buffer requested.

REFNO (parameter byte 7-8) - Contains the reference number of the buffer provided.

RECEIPT CODES: 00 = Initialization is complete.

01 = Illegal command.

02 = Illegal parameter.

FF = Command not yet complete.

DESCRIPTION: This primitive is used to request that a buffer be allocated to the IDPC DLC LLD. This primitive is used to obtain a buffer for the DLC receiver. The deallocate option is not used by the IDPC DLC LLD. The REFNO is an arbitrary integer associated with the buffer by the ME when the buffer is allocated.

NOTES: The ME must be able to supply a buffer of at least the size programmed in the Maximum Packet Size field in the IDPC DLC Initialization Parameter Block.

Advanced
Micro
Devices

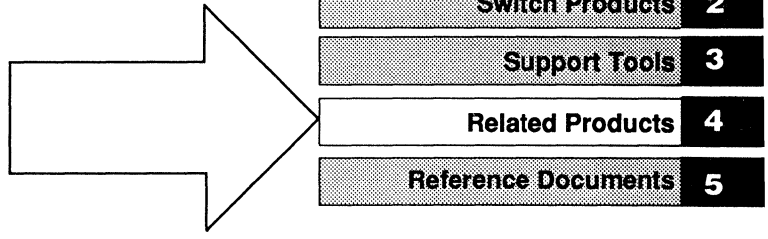


Table of Contents

Advance Information

Am79C04/A

IOM-2-DSLAC Specification

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Performs all the functions of a SLAC™
- Two independent channels
- Low power CMOS
- SLIC clock output
- A-law or mu-law coding
- External hardware reset
- Extended programmable filters
- Analog Impedance Scaling Network (AISN)
- IOM-2™ interface operation up to 8.192 MHz
- Compatible with 1.544 and 2.048 MHz primary rate interfaces
- Adaptive B filter
- 2.048 or 4.096 MHz Master Clock
- Analog output will drive transformer SLICs directly
- Control and voice PCM combined on single IOM-2 highway
- Some of the SLIC inputs are de-bounced with addressable timers
- Interface directly to Am795X or Am79M5X series SLICs

SYSTEM ARCHITECTURE

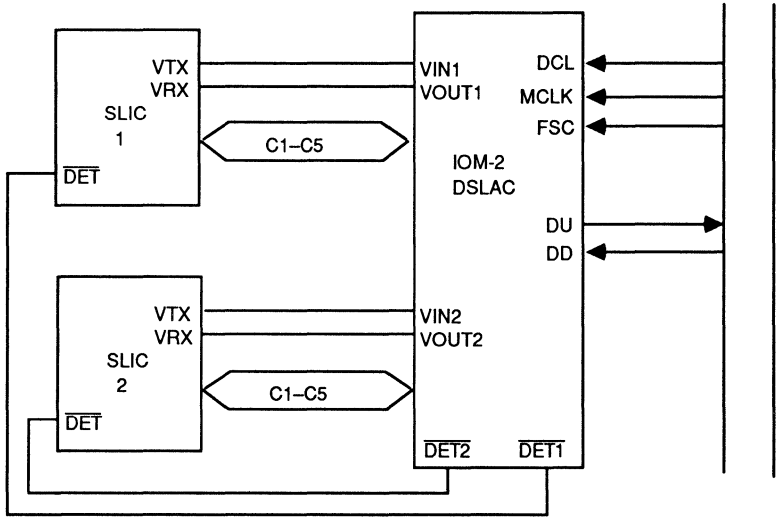


Figure 1. Shows the basic architecture of a Line Card based on the IOM-2-DSLAC™

Publication # 12764 Rev. A Amendment /0
Issue Date: September 1989

GENERAL DESCRIPTION

The IOM-2-DSLAC is designed to be used in Telecommunication Line Cards for both PBX and Central Office telephone exchanges. It converts the analog signal from the subscriber to digital PCM encoded signals for transmission on the IOM-2 highway and converts a PCM encoded signal received from the IOM-2 highway to an analog signal to be sent to the subscriber. When used with two SLICs, the IOM-2-DSLAC provides a complete, software configurable solution to the BORSCHT function.

The IOM-2-DSLAC is a variation of the Am79C02/A DSLAC wherein the Microprocessor Interface (MPI) and the Time Slot Assigner (TSA) have been modified to connect to the ISDN Oriented Modular 2 (IOM-2) interface. Additionally, the I/O lines used to control an external SLIC have been changed from the original DSLAC.

For details about filter transfer functions and audio performance specifications, please refer to the Am79C02/A Preliminary Data Sheet (order # 09875).

INTRODUCTORY DESCRIPTION

The IOM-2-DSLAC performs the CODEC and filter functions associated with the four-wire section of the subscriber line circuitry in a digital switch. These functions involve converting an analog voice signal into digital PCM samples and converting digital PCM samples back into an analog signal. During conversion, digital filters are used to band-limit the voice signals.

User programmable filters are provided to allow the setting of the receive and transmit gains, perform a trans-hybrid balancing function, allow adjustment of the two-wire termination impedance and provide frequency attenuation adjustment of the receive and transmit paths.

The PCM codes can be either 8-bit companded A-law or mu-law.

The independent channels appear as if the IOM-2-DSLAC contains two SLACs. All of the digital filtering is performed in digital signal processors operating from a 2.048 or 4.096 MHz external clock. The A/D, D/A and signal processing is separate for each channel.

The IOM-2-DSLAC is implemented in CMOS. This ensures low power dissapation and the higher speed necessary for implementing all the filters in the digital signal processors.

MANDATORY PERFORMANCE REQUIREMENTS

The IOM-2-DSLAC will be able to perform within the requirements of many differing worldwide specifications. Each country, throughout the world, generally has its own specification. Also there may be more than one specification within a country.

A performance specification for the IOM-2-DSLAC is described in the Am79C02/A Preliminary Data Sheet.

Preliminary

Am79C02/A

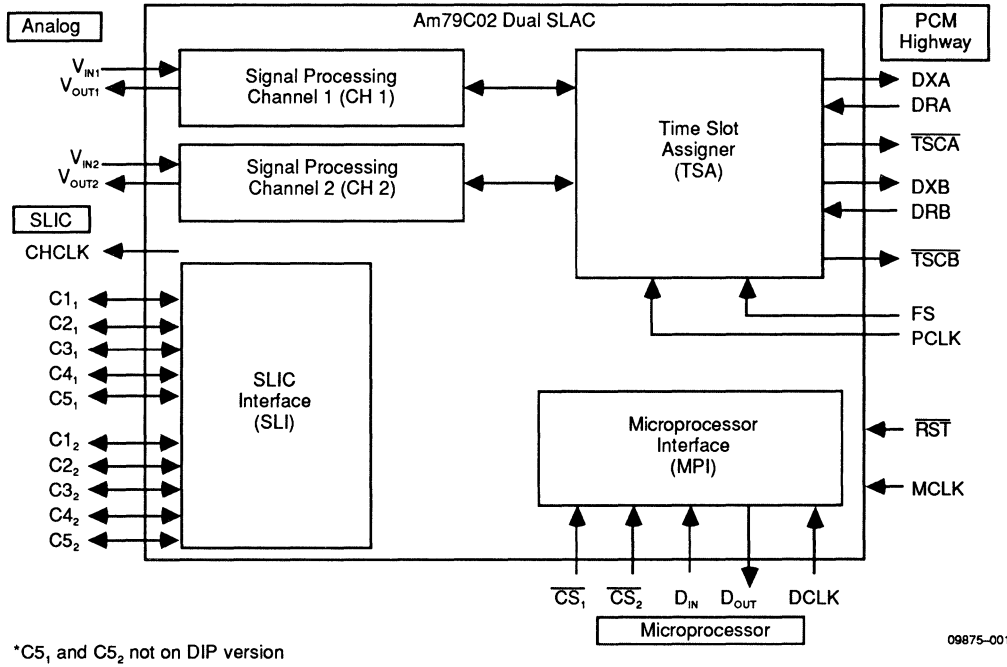
Dual Subscriber Line Audio-Processing Circuit (DSLAC™)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Software programmable:**
  - –SLIC impedance
  - –Trans-hybrid balance
  - –Transmit and Receive gains
  - –Equalization
  - –Digital I/O pins
  - –Time Slot Assigner
- **Adapt and freeze or fixed trans-hybrid balance filter (Am79C02 and Am79C02A)**
- **Continuously adapting trans-hybrid balance filter (Am79C02A only)**
- **A-law or μ-law coding**
- **Dual PCM ports**
  - –Up to 8.192 MHz (128 channels per port) through the PCM interface
- **2.048 MHz or 4.096 MHz master clock**
- **Direct Transformer Drive**
- **Built-in test modes**
- **Low power CMOS**
- **Mixed mode (analog and digital) impedance scaling**
- **Performance characteristics guaranteed over 12 dB gain range**

BLOCK DIAGRAM



*$C5_1$ and $C5_2$ not on DIP version

Publication # 09875 Rev. D Amendment /0
Issue Date: June 1989

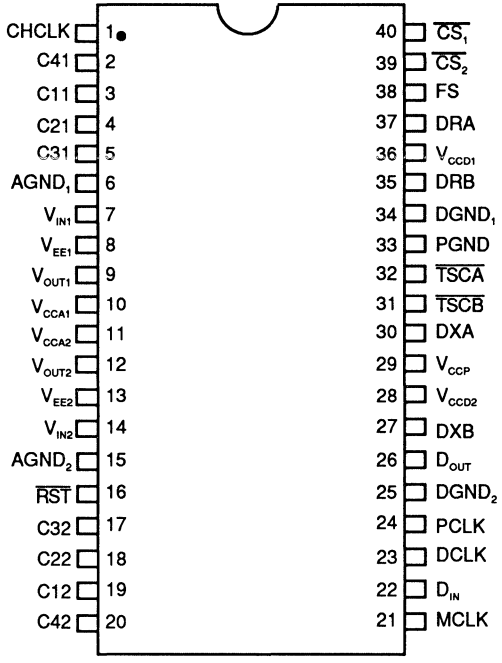
GENERAL DESCRIPTION

The Am79C02/A Dual Subscriber Line Audio-Processing Circuit (DSLAC) integrates the key functions of an analog linecard into one programmable, high-performance dual CODEC-filter device. The DSLAC is based on the proven design of the reliable Am7901A Subscriber Line Audio-Processing Circuit (SLAC™). The advanced architecture of the DSLAC implements two independent channels and employs digital filters to allow software control of transmission, thus providing a cost-effective solution for the four-wire-to-PCM section of a linecard.

Advanced CMOS technology makes the Am79C02/A DSLAC an economical device that has both the functionality and the low power consumption needed by linecard designers to maximize linecard density at minimum cost. When used with two SLICs, the DSLAC provides a complete, software-configurable solution to the BORSCHT function.

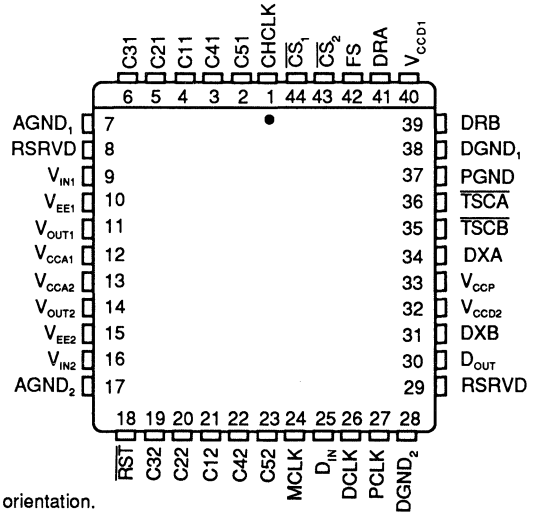
CONNECTION DIAGRAMS
Top View

40-Pin DIP



09875-002C

44-Pin PLCC



Notes: 1. Pin 1 is marked for orientation.
2. RSRVD = No connection

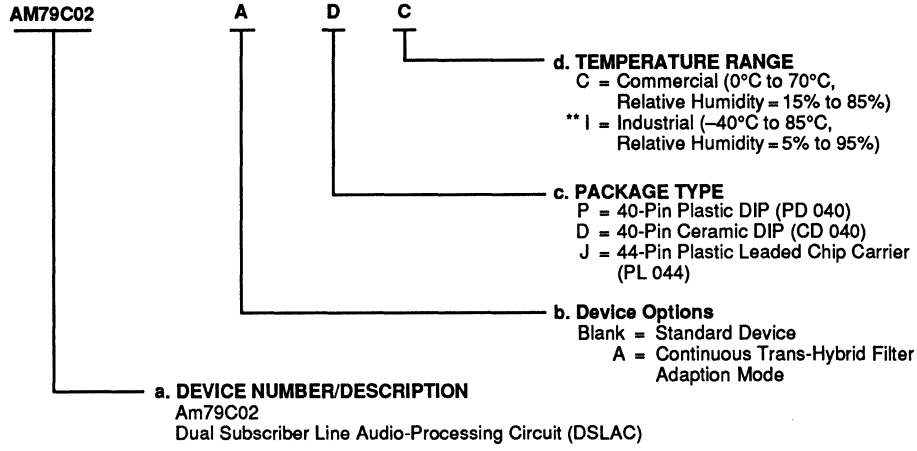
09875-003C

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range



Valid Combinations			
AM79C02	ADC ADI DC DI	APC API PC PI	AJC AJI JC JI

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

**The performance specifications contained in this Data Sheet are valid for the commercial temperature range device only. The specifications for the industrial temperature range device will be released after full characterization.

PIN DESCRIPTION

C11–C51, C12–C52
SLIC Input/Outputs (Input/Output)

The five SLIC control lines per channel are TTL compatible and bidirectional. They can be used to monitor or control the operation of a SLIC or any other device associated with the subscriber line. Lines C11–C51 are associated with Channel 1, and lines C12–C52 are associated with Channel 2. The C51 and C52 lines are only available on the 44-pin PLCC version of the DSLAC.

CHCLK
SLIC Clock (Output)

This output provides a 256-kHz, 50-duty cycle, TTL-compatible clock for use by two SLICs. The CHCLK frequency is synchronous to MCLK but the phase relationship to MCLK is random. CHCLK is capable of driving two TTL inputs.

$\overline{CS_1}$, $\overline{CS_2}$
Chip Selects (Input, Active LOW)

The Chip Select inputs enable the device to read or write control data. $\overline{CS_1}$ is for the Channel 1 microprocessor interface, and $\overline{CS_2}$ is for the Channel 2 microprocessor interface.

DCLK
Data Clock (Input)

The Data Clock input shifts data either into or out of the Microprocessor Interface of the DSLAC. The maximum clock rate is 4.096 MHz.

D_{IN}
Data Input (Input)

Control data is serially written into the DSLAC via the D_{IN} pin with the most significant bit first. The Data Clock determines the data rate. D_{IN} and D_{OUT} may be strapped together to reduce the number of connections to the microprocessor.

D_{OUT}
Data Output (Output)

Control data is serially read out of the DSLAC via the D_{OUT} pin with the most significant bit first. The Data Clock determines the data rate. D_{OUT} is high impedance except when data is being transmitted from the DSLAC under control of $\overline{CS_1}$ or $\overline{CS_2}$. D_{IN} and D_{OUT} may be strapped together to reduce the number of connections to the microprocessor.

DRA, DRB
PCM Inputs (Input)

The Receive PCM data for Channels 1 and 2 is serially received on either the DRA or the DRB port with port selection under user program control. Eight bits are received with the most significant bit first. Data for each channel is received in 8-bit bursts every 125 µs at the PCLK rate.

DXA, DXB
PCM Outputs (Output)

The Transmit PCM data from Channels 1 and 2 is sent serially through either the DXA or DXB port with port selection under user program control. Eight bits are transmitted with the most significant bit first. The output is available every 125 µs and the data is shifted out in 8-bit bursts at the PCLK rate. DXA and DXB are high impedance between bursts and while the device is in the Inactive mode.

FS
Frame Sync (Input)

The Frame Sync pulse is an 8-kHz signal that identifies the beginning of a frame. The DSLAC references individual time slots with respect to this input, which must be synchronized to PCLK.

MCLK
Master Clock (Input)

The Master Clock must be a 2.048-MHz or 4.096-MHz clock input for use by the digital signal processor. MCLK may be asynchronous to PCLK.

PCLK
PCM Clock (Input)

The PCM clock determines the rate at which PCM data is serially shifted into or out of the PCM ports. The maximum clock frequency is 8.192 MHz and the minimum clock frequency is 128 kHz. The PCLK clock may be asynchronous to MCLK.

$\overline{RST}$
Reset (Input, Active LOW)

A TTL LOW signal on this input resets the DSLAC to its default state.

$\overline{TSCA}$, $\overline{TSCB}$
Time Slot Control (Output, Open Drain, Active LOW)

The Time Slot Control outputs are open drain (requiring pull-up resistors) and are normally inactive (high impedance). $\overline{TSCA}$ is active (LOW) when PCM data is present on the DXA output and $\overline{TSCB}$ is active (LOW) when PCM data is present on the DXB output.

V_{IN1}, V_{IN2}
Analog Inputs (Input)

The analog input is applied to the transmit path of the DSLAC. The signal is sampled, digitally processed and encoded for the PCM output. V_{IN1} is the input for Channel 1 and V_{IN2} is the input for Channel 2.

V_{OUT1}, V_{OUT2}
Analog Outputs (Output)

The received PCM data is digitally processed and converted to an analog signal at the V_{OUT} pin. V_{OUT1} is the output from Channel 1 and V_{OUT2} is the output for Channel 2. These outputs can directly drive a transformer SLIC.

4

$AGND_1$	Analog Ground—Channel 1
$AGND_2$	Analog Ground—Channel 2
$DGND_1$	Digital Ground 1
$DGND_2$	Digital Ground 2
PGND	PCM I/O Ground
V_{CCA1}	+5 V Analog Power Supply—Channel 1
V_{CCA2}	+5 V Analog Power Supply—Channel 2
V_{CCD1}	+5 V Digital Power Supply. Internally connected to substrate on the IC
V_{CCD2}	+5 V Digital Power Supply. Internally connected to substrate on the IC
V_{CCP}	+5 V PCM I/O Power Supply. Internally connected to substrate on the IC
V_{EE1}	–5 V Power Supply—Channel 1
V_{EE2}	–5 V Power Supply—Channel 2

The many separate power supply inputs are intended to provide for good power supply decoupling techniques. Note that all of the +5 volt inputs should be connected to the same source, all of the ground inputs should be connected to the same source, and both of the –5 volt inputs should be connected to the same source.

FUNCTIONAL DESCRIPTION

The DSLAC performs the CODEC and filter functions associated with the four-wire section of the subscriber line circuitry in a digital switch. These functions involve converting an analog voice signal into digital PCM samples and converting digital PCM samples back into an analog signal. During conversion, digital filters are used to band-limit the voice signals.

The user-programmable filters set the receive and transmit gain, perform the trans-hybrid balancing function, permit adjustment of the two-wire termination impedance, and provide frequency attenuation adjustment (equalization) of the receive and transmit paths. Adaptive trans-hybrid balancing is also included. All programmable digital filter coefficients can be calculated using the AmSLAC-II software. The PCM codes can be either 8-bit companded A-law or μ-law. The PCM data is read or written to the PCM highway in user-programmable time slots at rates of 128 kHz to 8.192 MHz. The output hold time can be adjusted for compatibility with other devices which can be connected to the PCM highway.

The independent channels allow the DSLAC to function as two SLACs. All of the digital filtering is performed in digital signal processors operating from either a 2.048-MHz or 4.096-MHz external clock. The A/D, D/A, and signal processing is separate for each channel and each channel has its own chip select ($\overline{CS_1}$ and $\overline{CS_2}$) to allow separate programming. The dual channel DSLAC is available in a 40-pin DIP or a 44-pin PLCC, with the PLCC version having one extra SLIC I/O line per channel.

The following documentation describes the operation of a single channel of the DSLAC. The description is valid for either Channels 1 or 2. V_{IN} in this data sheet refers to either V_{IN1} or V_{IN2}, V_{OUT} refers to either V_{OUT1} or V_{OUT2}, and $\overline{CS}$ refers to either $\overline{CS_1}$ or $\overline{CS_2}$.

Operational Modes

Active Mode

Each channel of the DSLAC can operate in either the active (operational) or inactive (standby) mode. In the active mode, the DSLAC is able to transmit and receive PCM and analog information. This is the normal operating mode when a telephone call is in progress. The Activate command, Microprocessor Interface (MPI) Command #5, puts the device into this state. Bringing the DSLAC into the active mode is only possible through the MPI.

Inactive Mode

The DSLAC is forced into the inactive (standby) mode by a hardware or software reset, or is programmed into this mode by the Inactivate command (Command #1). No transmission or reception of PCM data takes place, but the circuits which contain programmed information retain their data. Power is switched off from all non-essential circuitry, though the MPI remains active to receive new commands. The analog output is tied to ground through approximately 3 kohm resistors. Upon initial application of power, the DSLAC is forced into the inactive mode.

Reset State

An active LOW, hardware Reset pin ($\overline{RST}$) is available on the DSLAC which resets the device to the following default state:

1. A-law is selected.
2. B, X, R and Z filters are disabled and AISN gain is zero.
3. Transmit (GX & AX) and receive (GR & AR) gains are set to unity.
4. SLIC input/output direction is set to the input mode.
5. Normal conditions are selected (see Command #4).
6. The B-filter adaptive mode is turned off.
7. Both channels are placed in the Inactive (standby) mode.
8. Transmit and receive time and clock slots are set to zero.
9. DXA/DRA ports are selected for Channel 1.
10. DXB/DRB ports are selected for Channel 2.
11. MCLK is selected to be 4.096 MHz.

Reset states 1 to 7 are identical to those of the software reset (Command #2), but the hardware reset applies to both channels simultaneously. When power is initially applied to the DSLAC or when $\overline{RST}$ is asserted, the following sequence of actions is necessary to ensure correct operation of the DSLAC:

1. Select MCLK frequency (Command # 6).
2. Software reset (Command # 2).
3. Program filter coefficients and all other required parameters.

Upon initial application of power, a minimum of 1 msec is needed before $\overline{CS_1}$ or $\overline{CS_2}$ may go LOW and an MPI command initiated. If the power supply (V_{CCD1} or V_{CCD2}) falls below approximately 2.0 volts, the device is software-reset and will require complete reprogramming with the above sequence. Bit 7 of the SLIC Direction Register will read back as a logical 1 to indicate that a power interruption has been detected. This bit is cleared when a software reset command is sent to the DSLAC. The $\overline{RST}$ pin may be tied to +5 volts if it is not needed in the system.

Signal Processing

Overview of Digital Filters

Several of the blocks in the signal processing section are user programmable. These allow the user to optimize the performance of the DSLAC for the system. Figure 1 shows DSLAC signal processing and indicates the blocks that can be programmed.

The advantages of digital filters are:
–High reliability
–No drift with time or temperature
–Unit-to-unit repeatability
–Superior transmission performance

Two-Wire Impedance Matching

Two feedback paths on the DSLAC modify the effective two-wire input impedance of the SLIC by providing programmable feedback from V_{IN} to V_{OUT}. The Analog Impedance Scaling Network (AISN) is a programmable analog gain of –0.9375 to +0.9375 from V_{IN} to V_{OUT}. The Z filter is a programmable digital filter, also connecting V_{IN} to V_{OUT}.

Distortion Correction and Equalization

The DSLAC contains programmable filters in the receive (R) and transmit (X) directions that may be programmed for line equalization and to correct any attenuation distortion caused by the Z filter.

Trans-Hybrid Balancing

The DSLAC programmable B filter provides trans-hybrid balance. The filter has a single-pole IIR section (B-IIR) and an eight-tap FIR section (B-FIR), both operating at 16 kHz. The DSLAC has an optional adaptive mode for the B filter which may be used to achieve optimum performance. The Echo Path Gain (EPG) and Error Level Threshold (ELT) registers contain values which determine the adaptive mode performance.

Gain Adjustment

The DSLAC transmit path has two programmable gain blocks. Gain block AX is an analog gain of 0 dB or 6.02 dB, located immediately before the A/D converter. Gain block GX is a digital gain that is programmable from 0 dB to 12 dB with a minimum step size of .1 dB if the gain setting is below +10 dB, and a minimum step size of .3 dB for gain settings above +10 dB (0 dBm0 is defined as 1.55 VRMS at V_{IN}). The filters provide a net gain in the range of 0 dB to 18 dB.

The DSLAC receive path has two programmable loss blocks. Loss block GR is a digital loss that is programmable from 0 dB to 12 dB with a worst case step size of 0.1 dB (0 dBm0 is defined as 1.55 VRMS at V_{OUT}). Loss block AR is an analog loss of 0 dB or 6.02 dB, located immediately after the D/A converter. This provides a net loss in the range of 0 dB to 18 dB.

Transmit Signal Processing

In the transmit path, the analog input signal is A/D converted, filtered, companded (A- or μ-law), and made available for output to the PCM highway. The signal processor contains an ALU, RAM, ROM, and control logic to implement the filter sections. The B, X, and GX blocks are user-programmable digital filter sections with coefficients stored in the coefficient RAM while AX is an analog amplifier which can be programmed for 0 dB or 6.02 dB gain. The filters may be made transparent when not required in a system.

The decimator reduces the high input sampling rate to 16 kHz for input to the B, GX, and X filters. The X filter is a 6-tap FIR section which is part of the frequency response correction network. The B filter operates on samples from the receive signal path in order to provide trans-hybrid balancing in the loop. The high-pass filter rejects low frequencies such as 50 or 60 Hz and may be disabled.

Transmit PCM Interface

The transmit PCM interface receives an 8-bit compressed code from the digital A/μ-law compressor. The transmit PCM interface logic (Figure 2) controls the transmission of the data onto the PCM highway through the output port selection circuitry and the time and clock slot control block.

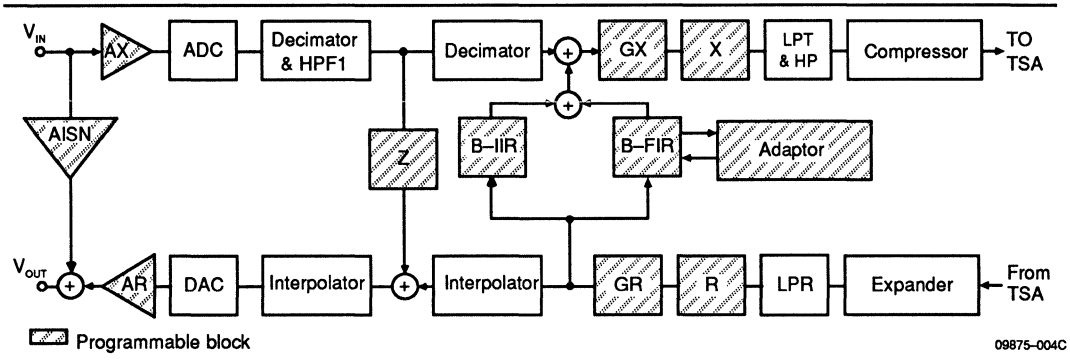


Figure 1. DSLAC Signal Processing

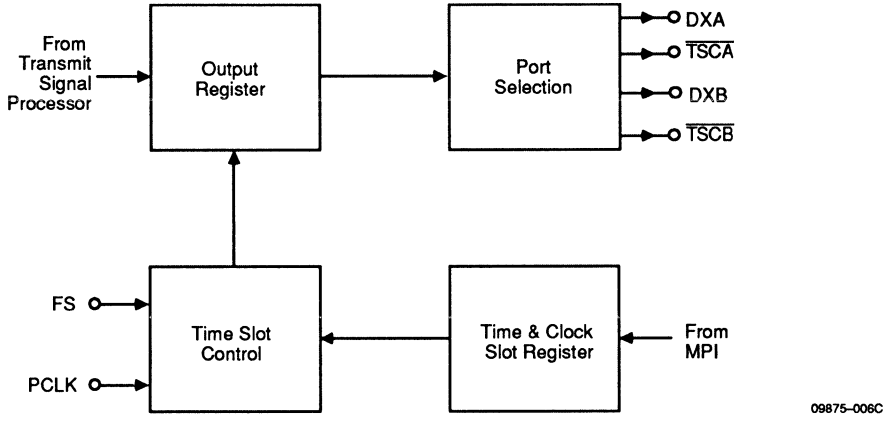


Figure 2. Transmit PCM Interface

The frame sync (FS) pulse identifies the beginning of a transmit frame and all channels (time slots) are referenced to it. The logic contains user-programmable Transmit Time Slot and Transmit Clock Slot registers.

The Time Slot register is 7 bits wide and allows up to 128 8-bit channels (using a PCLK of 8.192 MHz) in each frame. This feature allows any clock frequency between 128 kHz and 8.192 MHz (2 to 128 channels) in a system. The Clock Slot register is 3 bits wide and may be programmed to offset the time slot assignment by 0 to 7 PCLK periods to eliminate any clock skew in the system. The data is transmitted in bytes with the most significant bit first.

The PCM data may be user-programmed for output onto either the DXA or DXB port. Correspondingly, either $\overline{TSCA}$ or $\overline{TSCB}$ is LOW during transmission. An extra delay (PCM delay) in the timing of the DXA and DXB signals may be programmed to allow timing compatibility with other devices on the PCM highway.

Receive Signal Processing

In the receive path, the digital signal is expanded, filtered, converted to analog, and passed to the V_{OUT} pin. The signal processor contains an ALU, RAM, ROM, and Control logic to implement the filter sections. The Z, R, and GR blocks are user-programmable filter sections with their coefficients stored in the coefficient RAM while AR is an analog amplifier which can be programmed for a 0-dB or 6.02-dB loss. The filters may be made transparent when not required in a system.

The low-pass filter band limits the signal. The R filter is a 6-tap FIR section operating at a 16-kHz sampling rate and is part of the frequency response correction network. The analog impedance scaling network (AISN) is a user-programmable gain block providing feedback from V_{IN} to V_{OUT} to emulate different ZSLIC impedances from a single external ZSLIC impedance. The Z filter provides feedback from the transmit signal path to the receive path and is used to modify the effective input impedance to the system. The interpolator increases the sampling rate prior to D/A conversion.

Receive PCM Interface

The receive PCM interface logic (Figure 3) controls the reception of data bytes from the PCM highway, transfers the data to the A/μ-law expansion logic, and then passes the data to the receive path of the signal processor. The frame sync (FS) pulse identifies the beginning of a receive frame, and all channels (time slots) are referenced to it.

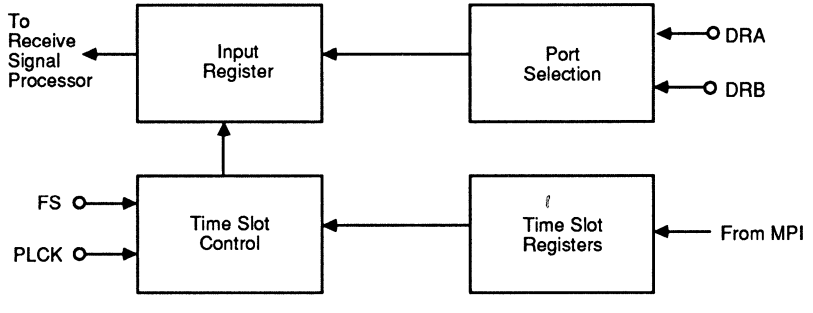


Figure 3. Receive PCM Interface

The logic contains user-programmable Receive Time Slot and Receive Clock Slot registers. The Time Slot register is 7 bits wide and allows up to 128 8-bit channels (using a PCLK of 8.192 MHz) in each frame. This feature allows any clock frequency between 128 kHz and 8.192 MHz (2 to 128 channels) in a system. The Clock Slot register is 3 bits wide and may be programmed to offset the time slot assignment by 0 to 7 PCLK periods to eliminate any clock skews in the system. The PCM data may be user-programmed for input from either the DRA or DRB port.

Analog Impedance Scaling Network (AISN)

The AISN is incorporated in the DSLAC to scale the value of the external Z_{SLIC} impedance. Scaling this external impedance with the AISN (along with the Z filter) allows matching of many different line conditions using a single impedance value. Line cards may be designed for many different specifications without any hardware changes.

The AISN is a programmable gain that is connected across the DSLAC input from V_{IN} to V_{OUT}. The gain can be varied from –0.9375 to +0.9375 in 31 steps of 0.0625. The AISN gain is given by the following equation:

$$h_{AISN} = 0.0625\,[(A \bullet 2^4 + B \bullet 2^3 + C \bullet 2^2 + D \bullet 2^1 + E \bullet 2^0) - 16]$$

where A, B, C, D, and E = 1 or 0.

The AISN gain is used to determine the input impedance of the DSLAC when terminated by Z_{SLIC} as shown in Figure 4.

The DSLAC input impedance is approximately given by:

$$Z_{IN} = \frac{1 - G_{44}\, h_{AISN}}{1 - G_{440}\, h_{AISN}} Z_{SL},$$

where G_{440} (defined as $G_{24}\, G_{42} + G_{44}$) is the echo gain into an open circuit and G_{44} is the echo gain into a short circuit.

There are two special cases to the formula for h_{AISN}. A value of ABCDE = "00000" will specify a gain of 0 (or cut-off), and a value of ABCDE = "10000" is a special case where the AISN circuitry is disabled and the V_{OUT} pad is connected internally to V_{IN} with a gain of 0 dB. This allows a digital-to-digital loopback mode wherein a digital PCM input signal is completely processed through the receive section all the way to the V_{OUT} pin, then connected internally to V_{IN} where it is processed through the transmit section and output as digital PCM data.

Speech Coding

The A/D and D/A conversion follows either the A-law or the μ-law as they are defined in CCITT rec. G.711. A- or μ-law operation is programmed using MPI Command #19. Alternate bit inversion is performed as part of the A-law coding.

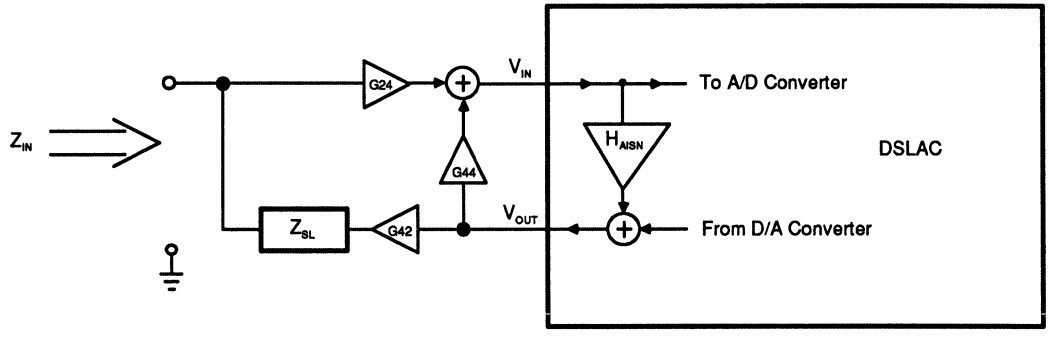


Figure 4. Input Impedance Modification Due to AISN

Command Description and Formats

Microprocessor Interface Description

A microprocessor may be used to program the DSLAC and control its operation using the Microprocessor Interface (MPI). Data programmed previously may be read out for verification. For each channel, commands are provided to assign values to the following parameters:

–Transmit time slot
–Receive time slot
–Transmit clock slot
–Receive clock slot
–Transmit gain
–Receive loss
–B-filter coefficients
–X-filter coefficients
–R-filter coefficients
–Z-filter coefficients
–Adaptive B filter parameters
–AISN coefficient
–Read/Write SLIC Input/Output
–Select A-law or μ-law code
–Select Transmit PCM Port A or B
–Select Receive PCM Port A or B
–Enable/disable B filter
–Enable/disable Z filter
–Enable/disable X filter
–Enable/disable R filter
–Enable/disable GX filter
–Enable/disable GR filter
–Enable/disable AX amplifier
–Enable/disable AR amplifier
–Enable/disable adaptive B filter
–Select test modes
–Select active or inactive (standby) mode

The following description of the MPI is valid for either Channel 1 or 2. Whenever $\overline{CS}$ is specified, it refers to either $\overline{CS_1}$ or $\overline{CS_2}$. If desired, both channels may be programmed simultaneously with identical information by activating $\overline{CS_1}$ and $\overline{CS_2}$ at the same time.

The MPI consists of serial data input (D_{IN}), output (D_{OUT}), data clock (DCLK), and a separate chip select (CS_1 and $\overline{CS_2}$) input for each channel (Figure 5). The serial input consists of 8-bit command words which may be followed with additional bytes of input data or may be followed by the DSLAC sending out bytes of data. All data input and output is MSB (D_7) first and LSB (D_0) last. All data bytes are read or written one at a time, with $\overline{CS}$ going HIGH for at least the minimum off-period before the next byte is read or written.

All commands that require additional input data to the device must have the input data as the next N words written into the device (for example, framed by the next N transitions of $\overline{CS}$). All commands that are followed by output data will cause the device to output data for the next N transitions of $\overline{CS}$ going LOW. The DSLAC will not accept any input commands until all the data has been shifted out. Unused bits in the data bytes are read out as zeros.

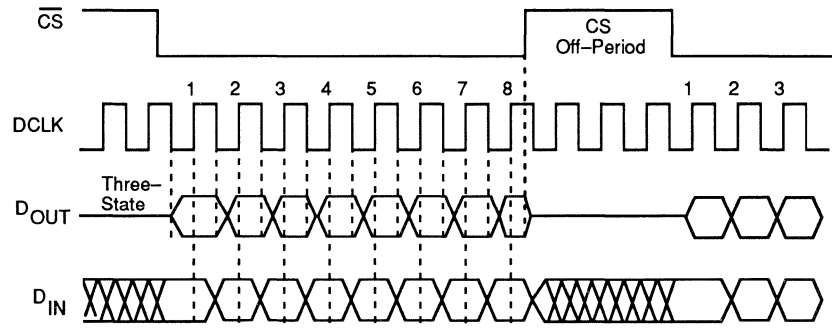


Figure 5. Microprocessor Interface Timing Diagram

An MPI cycle is defined by transitions of $\overline{CS}$ and DCLK. If the CS lines are held in the HIGH state between accesses, the DCLK may run continuously with no change to the internal control data. Using this method, the same DCLK may be run to a number of DSLACs and the individual $\overline{CS}$ lines will select the appropriate device to access. It should be noted that the DCLK can stay in the HIGH state indefinitely with no loss of internal control information regardless of any transitions on the $\overline{CS}$ lines. DCLK can stay in the LOW state indefinitely with no loss of internal control information, provided the $\overline{CS}$ lines remain at a HIGH level.

Summary of MPI Commands**

C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Description
1.	00	0	0	0	0	0	0	0	0	Inactivate (standby mode)
2.	02	0	0	0	0	0	0	1	0	Reset
3.	06	0	0	0	0	0	1	1	0	No Operation
4.	08	0	0	0	0	1	0	0	0	Reset to Normal Conditions
5.	0E	0	0	0	0	1	1	1	0	Activate
6.	1*	0	0	0	1	0	0	*	0	MCLK Selection
7.	40	0	1	0	0	0	0	0	0	Write TX Time Slot & PCM Highway
8.	41	0	1	0	0	0	0	0	1	Read TX Time Slot & PCM Highway
9.	42	0	1	0	0	0	0	1	0	Write RX Time Slot & PCM Highway
10.	43	0	1	0	0	0	0	1	1	Read RX Time Slot & PCM Highway
11.	44	0	1	0	0	0	1	0	0	Write RX & TX Clock Slot Selection
12.	45	0	1	0	0	0	1	0	1	Read RX & TX Clock Slot Selection
13.	50	0	1	0	1	0	0	0	0	Write AISN, PCM delay, Analog gains
14.	51	0	1	0	1	0	0	0	1	Read AISN, PCM delay, Analog gains
15.	52	0	1	0	1	0	0	1	0	Write SLIC Input/Output register
16.	53	0	1	0	1	0	0	1	1	Read SLIC Input/Output register
17.	54	0	1	0	1	0	1	0	0	Write SLIC Input/Output direction
18.	55	0	1	0	1	0	1	0	1	Read SLIC I/O direction and Power Down bit
19.	60	0	1	1	0	0	0	0	0	Write Operating Functions
20.	61	0	1	1	0	0	0	0	1	Read Operating Functions
21.	70	0	1	1	1	0	0	0	0	Write Operating Conditions
22.	71	0	1	1	1	0	0	0	1	Read Operating Conditions
23.	73	0	1	1	1	0	0	1	1	Read Revision Code Number
24.	80	1	0	0	0	0	0	0	0	Write GX Filter Coefficients
25.	81	1	0	0	0	0	0	0	1	Read GX Filter Coefficients
26.	82	1	0	0	0	0	0	1	0	Write GR Filter Coefficients
27.	83	1	0	0	0	0	0	1	1	Read GR Filter Coefficients
28.	84	1	0	0	0	0	1	0	0	Write Z Filter Coefficients
29.	85	1	0	0	0	0	1	0	1	Read Z Filter Coefficients
30.	86	1	0	0	0	0	1	1	0	Write B Filter Coefficients
31.	87	1	0	0	0	0	1	1	1	Read B Filter Coefficients
32.	88	1	0	0	0	1	0	0	0	Write X Filter Coefficients
33.	89	1	0	0	0	1	0	0	1	Read X Filter Coefficients
34.	8A	1	0	0	0	1	0	1	0	Write R Filter Coefficients
35.	8B	1	0	0	0	1	0	1	1	Read R Filter Coefficients
36.	8C	1	0	0	0	1	1	0	0	Write Echo Path Gain
37.	8D	1	0	0	0	1	1	0	1	Read Echo Path Gain
38.	8E	1	0	0	0	1	1	1	0	Write Error Level Threshold
39.	8F	1	0	0	0	1	1	1	1	Read Error Level Threshold

*Code changes with function.
**All codes not listed are reserved by AMD and should not be used.

THE COMMAND STRUCTURE

This section describes in detail each of the MPI commands. Each of the commands is shown along with the format of any additional data bytes that follow. For details of the filter coefficients of the form $C_{xy}m_{bxy}$, please refer to the "Description of Coefficients" section.

1. **Inactivate** (standby mode)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0

During the inactive mode (of one or both channels):
a) all of the programmed information is retained.
b) the Microprocessor Interface (MPI) remains active.
c) the PCM outputs are in high impedance and the PCM inputs are disabled.
d) the analog output is tied to zero volts through an internal resistor (~3 kohm).

2. **Reset**

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	1	0

The reset state of the device is:
a) A-law is selected.
b) B, X, R, and Z filters are disabled and AISN gain is zero.
c) transmit (GX & AX) and Receive (GR & AR) gains are set to unity.
d) all SLIC I/O lines are configured as inputs.
e) normal conditions are selected (see Command #4).
f) the B-filter Adaptive mode is reset.
g) the channel is placed in the inactive (standby) mode.

3. **No Operation**

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	1	0

4. **Reset to Normal Conditions**

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	0	0	0

Reset to Normal Conditions performs the following operations:
a) does not insert 6 dB loss in receive path.
b) receive & transmit paths are not cutoff.
c) high pass filter is enabled.
d) test modes are turned off.
e) PCM delay is inserted.

5. **Activate** (Operational Mode)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	1	1	0

This command places the device in the active mode. No valid PCM data is transmitted until after the second FS pulse is received following the execution of the Activate command.

6. MCLK Selection

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	0	A	0

MCLK may be selected to operate from a 2.048-MHz or 4.096-MHz external clock.
A = 0: 2.048 MHz
A = 1: 4.096 MHz

7. Write, Transmit Time Slot, & PCM Highway Selection

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	0	0	0	0	0	0
Output data:	PCM	TS	TS	TS	TS	TS	TS	TS

PCM = 0: Highway A TS: Time slot number 0 to 127
PCM = 1: Highway B

The Transmit section of both channels must not be set to the same time slot on the same output port simultaneously.

8. Read, Transmit Time Slot, & PCM Highway Selection

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	0	0	0	0	0	1
Output data:	PCM	TS	TS	TS	TS	TS	TS	TS

9. Write Receive Time Slot & PCM Highway Selection

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	0	0	0	0	1	0
Input data:	PCM	TS	TS	TS	TS	TS	TS	TS

PCM = 0: Highway A TS: Time slot number 0 to 127
PCM = 1: Highway B

10. Read, Receive Time Slot, & PCM Highway Selection

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	0	0	0	0	1	1
Output data:	PCM	TS	TS	TS	TS	TS	TS	TS

11. Write, Transmit, & Receive Clock Slot Selection

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	0	0	0	1	0	0
Input data:	—	—	RCS	RCS	RCS	TCS	TCS	TCS

TCS: Transmit Clock Slot number 0 to 7
RCS: Receive Clock Slot number 0 to 7

12. Read, Transmit, & Receive Clock Slot Selection

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	0	0	0	1	0	1
Output data:	—	—	RCS	RCS	RCS	TCS	TCS	TCS

13. Write AISN, PCM Delay, & Analog Gains

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	0	1	0	0	0	0
Input data:	PCD	AX	AR	A	B	C	D	E

PCM Delay: PCD = 0 Delay inserted (SLAC compatible)
PCD = 1 Delay removed (high speed)

Transmit Analog Gain: AX = 0 0 dB gain
AX = 1 6.02 dB gain

Receive Analog Loss: AR = 0 0 dB loss
AR = 1 6.02 dB loss

AISN coefficient: A, B, C, D, E

The Analog Impedance Scaling Network (AISN) gain can be varied from –0.9375 to 0.9375 in multiples of 0.0625. The gain coefficient is decoded using the following equation:

$$h_{AISN} = 0.0625\,[(A \bullet 2^4 + B \bullet 2^3 + C \bullet 2^2 + D \bullet 2^1 + E \bullet 2^0) - 16],$$

where h_{AISN} is the gain of the AISN and A, B, C, D, and E = 0 or 1. A value of ABCDE = "10000" implements a special digital loopback mode, and a value of ABCDE = "00000" indicates a gain of 0 (cutoff).

14. Read AISN, PCM Delay, & Analog Gains

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	0	1	0	0	0	1
Output data:	PCD	AX	AR	A	B	C	D	E

15. Write SLIC Input/Output Register

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	0	1	0	0	1	0
Input data:	—	—	—	C5	C4	C3	C2	C1

C1 through C5 are set to 1 or 0. The data will appear latched on the C1 through C5 SLIC I/O pins, provided they were set in the output mode (see Command #17). The data for any of the pins set to the input mode will be ignored.

16. Read SLIC Input/Output Register

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	0	1	0	0	1	1
Output data:	—	—	—	C5	C4	C3	C2	C1

The logic state of pins C1 through C5 is read regardless of the direction programmed into the Input/Output register.

17. Write SLIC Input/Output Direction

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	0	1	0	1	0	0
Input data:	—	—	—	A	B	C	D	E

Pins C1 through C5 are set to input or output modes individually. The input mode is set when the appropriate data bit is a 0, and the output mode is set when the data bit is a 1. All unused SLIC I/O pins should be programmed as outputs to reduce power consumption.

Data bit A sets pins C51 or C52.
Data bit B sets pins C41 or C42.
Data bit C sets pins C31 or C32.
Data bit D sets pins C21 or C22.
Data bit E sets pins C11 or C12.

18. Read SLIC Input/Output Direction and Power Interrupt Bit

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	0	1	0	1	0	1
Output data:	PI	—	—	A	B	C	D	E

PI = 0 There has not been a power interruption since the last software reset command.

PI = 1 A power interruption has been previously detected requiring the DSLAC to be completely reprogrammed. This bit is cleared by issuing a software reset command.

19. Write Operating Functions

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	1	0	0	0	0	0
Input data:	ABF	A/μ	EGR	EGX	EX	ER	EZ	EB

Adaptive B-Filter: ABF = 0 B filter non-adaptive mode
ABF = 1 B filter adaptive mode

A-law/μ-law: A/μ = 0 A-law coding
A/μ = 1 μ-law coding

GR Filter: EGR = 0 GR filter disabled
EGR = 1 GR filter enabled

GX Filter: EGX = 0 GX filter disabled
EGX = 1 GX filter enabled

X Filter: EX = 0 X filter disabled
EX = 1 X filter enabled

R Filter: ER = 0 R filter disabled
ER = 1 R filter enabled

Z Filter: EZ = 0 Z filter disabled
EZ = 1 Z filter enabled

B Filter: EB = 0 B filter disabled
EB = 1 B filter enabled

Note: The enable adaptive B-filter command is only effective when used with the enable B-filter command.

20. Read Operating Functions

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	1	0	0	0	0	1
Output data:	ABF	A/U	EGR	EGX	EX	ER	EZ	EB

21. Write Operating Conditions

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	1	1	0	0	0	0
Input data:	CTP	CRP	HPF	RG	ALB	TLB	—	—

Cut off Transmit Path:
CTP = 0 Transmit path connected
CTP = 1 Transmit path cut off

Cut off Receive Path:
CRP = 0 Receive path connected
CRP = 1 Receive path cut off

High-Pass Filter:
HPF = 0 High-pass filter enabled
HPF = 1 High-pass filter disabled

Receive Path Gain:
RG = 0 6 dB loss not inserted
RG = 1 6 dB loss inserted

Analog Loopback:
ALB = 0 Analog loopback disabled
ALB = 1 Analog loopback enabled

TSA Loopback:
TLB = 0 TSA loopback disabled
TLB = 1 TSA loopback enabled

22. Read Operating Conditions

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	1	1	0	0	0	1
Output data:	CTP	CRP	HPF	RG	ALB	TLB	—	—

23. Read Revision Code Number

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	0	1	1	1	0	0	1	1
Output data:	#	#	#	#	#	#	#	#

This command returns an eight-bit number describing the revision number of the DSLAC.

24. Write GX Filter Coefficients

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	0	0	0	0
Input data byte 1:	C40		m40		C30		m30	
Input data byte 2:	C20		m20		C10		m10	

The coefficient for the GX filter is defined as:

$$H_{GX} = 1 + \left(C_{10} \bullet 2^{-m10}\{1 + C_{20} \bullet 2^{-m20}[1 + C_{30} \bullet 2^{-m30}(1 + C_{40} \bullet 2^{-m40})]\}\right).$$

25. Read GX Filter Coefficients

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	0	0	0	1
Output data byte 1:	C40		m40		C30		m30	
Output data byte 2:	C20		m20		C10		m10	

26. Write GR Filter Coefficients

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	0	0	1	0
Input data byte 1:	C40		m40		C30		m30	
Input data byte 2:	C20		m20		C10		m10	

The coefficient for the GR filter is defined as:

$$H_{GR} = C_{10} \bullet 2^{-m10}\{1 + C_{20} \bullet 2^{-m20}[1 + C_{30} \bullet 2^{-m30}(1 + C_{40} \bullet 2^{-m40})]\}.$$

27. Read GR Filter Coefficients

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	0	0	1	1
Output data byte 1:	C40		m40		C30		m30	
Output data byte 2:	C20		m20		C10		m10	

28. Write Z Filter Coefficients

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	0	1	0	0
Input data byte 1:	C26		m26		C16		m16	
Input data byte 2:	C27		m27		C17		m17	
Input data byte 3:	C40		m40		C30		m30	
Input data byte 4:	C20		m20		C10		m10	
Input data byte 5:	C41		m41		C31		m31	
Input data byte 6:	C21		m21		C11		m11	
Input data byte 7:	C42		m42		C32		m32	
Input data byte 8:	C22		m22		C12		m12	
Input data byte 9:	C43		m43		C33		m33	
Input data byte 10:	C23		m23		C13		m13	
Input data byte 11:	C44		m44		C34		m34	
Input data byte 12:	C24		m24		C14		m14	
Input data byte 13:	C45		m45		C35		m35	
Input data byte 14:	C25		m25		C15		m15	

The Z-transform equation for the Z filter is defined as:

$$H_Z(z) = Z_0 + Z_1z^{-1} + Z_2z^{-2} + Z_3z^{-3} + Z_4z^{-4} + Z_5z^{-5} + \frac{Z_6}{1 - Z_7z^{-1}}.$$

The coefficients for the FIR Z section are defined as:

$$Z_i = C_{1i} \bullet 2^{-m1i}\{1 + C_{2i} \bullet 2^{-m2i}[1 + C_{3i} \bullet 2^{-m3i}(1 + C_{4i} \bullet 2^{-m4i})]\}$$
for i = 1,2,3,4,5.

The coefficients for the IIR Z section are defined as:

$$Z_i = C_{1i} \bullet 2^{-m1i}(1 + C_{2i} \bullet 2^{-m2i})$$
for i = 6 or 7.

29. Read Z-Filter Coefficients

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	0	1	0	1
Output data byte 1:	C26		m26		C16		m16	
Output data byte 2:	C27		m27		C17		m17	
Output data byte 3:	C40		m40		C30		m30	
Output data byte 4:	C20		m20		C10		m10	
Output data byte 5:	C41		m41		C31		m31	
Output data byte 6:	C21		m21		C11		m11	
Output data byte 7:	C42		m42		C32		m32	
Output data byte 8:	C22		m22		C12		m12	
Output data byte 9:	C43		m43		C33		m33	
Output data byte 10:	C23		m23		C13		m13	
Output data byte 11:	C44		m44		C34		m34	
Output data byte 12:	C24		m24		C14		m14	
Output data byte 13:	C45		m45		C35		m35	
Output data byte 14:	C25		m25		C15		m15	

30. Write B-Filter Coefficients

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	0	1	1	0
Input data byte 1:	C30		m30		C20		m20	
Input data byte 2:	C10		m10		C31		m31	
Input data byte 3:	C21		m21		C11		m11	
Input data byte 4:	C32		m32		C22		m22	
Input data byte 5:	C12		m12		C33		m33	
Input data byte 6:	C23		m23		C13		m13	
Input data byte 7:	C34		m34		C24		m24	
Input data byte 8:	C14		m14		C35		m35	
Input data byte 9:	C25		m25		C15		m15	
Input data byte 10:	C36		m36		C26		m26	
Input data byte 11:	C16		m16		C37		m37	
Input data byte 12:	C27		m27		C17		m17	
Input data byte 13:	C48		m48		C38		m38	
Input data byte 14:	C28		m28		C18		m18	

The z = transform equation for the B filter is defined as:

$$H_B(z) = B_0 + B_1z^{-1} + B_2z^{-2} + B_3z^{-3} + B_4z^{-4} + B_5z^{-5} + B_6z^{-6} + \frac{B_7z^{-7}}{1 - B_8z^{-1}}.$$

The coefficients for the FIR B section and the gain of the IIR B section are defined as:

$$B_i = C_{1i} \bullet 2^{-m1i}[1 + C_{2i} \bullet 2^{-m2i}(1 + C_{3i} \bullet 2^{-m3i})].$$

The feedback coefficient of the IIR B section is defined as:

$$B_8 = C_{18} \bullet 2^{-m18}\ \{1 + C_{28} \bullet 2^{-m28}[1 + C_{38} \bullet 2^{-m38}(1 + C_{48} \bullet 2^{-m48})]\}.$$

31. Read B-Filter Coefficients

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	0	1	1	1
Output data byte 1:	C30		m30		C20		m20	
Output data byte 2:	C10		m10		C31		m31	
Output data byte 3:	C21		m21		C11		m11	
Output data byte 4:	C32		m32		C22		m22	
Output data byte 5:	C12		m12		C33		m33	
Output data byte 6:	C23		m23		C13		m13	
Output data byte 7:	C34		m34		C24		m24	
Output data byte 8:	C14		m14		C35		m35	
Output data byte 9:	C25		m25		C15		m15	
Output data byte 10:	C36		m36		C26		m26	
Output data byte 11:	C16		m16		C37		m37	
Output data byte 12:	C27		m27		C17		m17	
Output data byte 13:	C48		m48		C38		m38	
Output data byte 14:	C28		m28		C18		m18	

32. Write X-Filter Coefficients

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	1	0	0	0
Input data byte 1:	C40		m40		C30		m30	
Input data byte 2:	C20		m20		C10		m10	
Input data byte 3:	C41		m41		C31		m31	
Input data byte 4:	C21		m21		C11		m11	
Input data byte 5:	C42		m42		C32		m32	
Input data byte 6:	C22		m22		C12		m12	
Input data byte 7:	C43		m43		C33		m33	
Input data byte 8:	C23		m23		C13		m13	
Input data byte 9:	C44		m44		C34		m34	
Input data byte 10:	C24		m24		C14		m14	
Input data byte 11:	C45		m45		C35		m35	
Input data byte 12:	C25		m25		C15		m15	

The z-transform equation for the X filter is defined as:

$$H_X(z) = X_0 + X_1 z^{-1} + X_2 z^{-2} + X_3 z^{-3} + X_4 z^{-4} + X_5 z^{-5}.$$

The coefficients for the X filter are defined as:

$$X_i = C_{1i} \bullet 2^{-m1i}\{1 + C_{2i} \bullet 2^{-m2i}[1 + C_{3i} \bullet 2^{-m3i}(1 + C_{4i} \bullet 2^{-m4i})]\}.$$

33. Read X-Filter Coefficients

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	1	0	0	1
Output data byte 1:	C40		m40		C30		m30	
Output data byte 2:	C20		m20		C10		m10	
Output data byte 3:	C41		m41		C31		m31	
Output data byte 4:	C21		m21		C11		m11	
Output data byte 5:	C42		m42		C32		m32	
Output data byte 6:	C22		m22		C12		m12	
Output data byte 7:	C43		m43		C33		m33	
Output data byte 8:	C23		m23		C13		m13	
Output data byte 9:	C44		m44		C34		m34	
Output data byte 10:	C24		m24		C14		m14	
Output data byte 11:	C45		m45		C35		m35	
Output data byte 12:	C25		m25		C15		m15	

34. Write R-Filter Coefficients

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	1	0	1	0
Input data byte 1:	C40		m40		C30		m30	
Input data byte 2:	C20		m20		C10		m10	
Input data byte 3:	C41		m41		C31		m31	
Input data byte 4:	C21		m21		C11		m11	
Input data byte 5:	C42		m42		C32		m32	
Input data byte 6:	C22		m22		C12		m12	
Input data byte 7:	C43		m43		C33		m33	
Input data byte 8:	C23		m23		C13		m13	
Input data byte 9:	C44		m44		C34		m34	
Input data byte 10:	C24		m24		C14		m14	
Input data byte 11:	C45		m45		C35		m35	
Input data byte 12:	C25		m25		C15		m15	

The z-transform equation for the R filter is defined as:

$$H_R(z) = R_0 + R_1z^{-1} + R_2z^{-2} + R_3z^{-3} + R_4z^{-4} + R_5z^{-5}.$$

The coefficients for the R filter are defined as:

$$R_i = C_{1i} \bullet 2^{-m1i}\{1 + C_{2i} \bullet 2^{-m2i}[1 + C_{3i} \bullet 2^{-m3i}(1 + C_{4i} \bullet 2^{-m4i})]\}.$$

35. Read R-Filter Coefficients

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	1	0	1	1
Output data byte 1:	C40		m40		C30		m30	
Output data byte 2:	C20		m20		C10		m10	
Output data byte 3:	C41		m41		C31		m31	
Output data byte 4:	C21		m21		C11		m11	
Output data byte 5:	C42		m42		C32		m32	
Output data byte 6:	C22		m22		C12		m12	
Output data byte 7:	C43		m43		C33		m33	
Output data byte 8:	C23		m23		C13		m13	
Output data byte 9:	C44		m44		C34		m34	
Output data byte 10:	C24		m24		C14		m14	
Output data byte 11:	C45		m45		C35		m35	
Output data byte 12:	C25		m25		C15		m15	

36. Write Echo Path Gain

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	1	1	0	0
Input data byte 1:	C80		m80		C70		m70	
Input data byte 2:	C60		m60		C50		m50	
Input data byte 3:	C40		m40		C30		m30	
Input data byte 4:	C20		m20		C10		m10	

The equation for the Echo Path Gain is defined as:

$$EPG = 8 \bullet C_{10} \bullet 2^{-m10} \left(1 + C_{50} \bullet 2^{-m50}\{1 + C_{60} \bullet 2^{-m60}[1 + C_{70} \bullet 2^{-m70}(1 + C_{80} \bullet 2^{-m80})]\}\right),$$

C_{20}, M_{20}, C_{30}, M_{30}, C_{40}, and M_{40} must be zero.

37. Read Echo Path Gain

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	1	1	0	1
Output data byte 1:	C80		m80		C70		m70	
Output data byte 2:	C60		m60		C50		m50	
Output data byte 3:	C40		m40		C30		m30	
Output data byte 4:	C20		m20		C10		m10	

38. Write Error Level Threshold

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	1	1	1	0
Input data byte 1:	C20		m20		C10		m10	

The equation for the Echo Path Gain is defined as:

$$ELT = C_{10} \bullet 2^{-m10}(1 + C_{20} \bullet 2^{-m20}).$$

39. Read Error Level Threshold

	D7	D6	D5	D4	D3	D2	D1	D0
Command:	1	0	0	0	1	1	1	1
Output data byte 1:	C20		m20		C10		m10	

Programmable Filters

General Description of CSD Coefficients

The filter functions are performed by a series of multiplications and accumulations. A multiplication is accomplished by repeatedly shifting the multiplicand and summing the result with the previous value at that summation node. The method used in the DSLAC is known as Canonic Signed Digit (CSD) multiplication and splits each coefficient into a series of CSD coefficients.

Each programmable FIR filter section has the following general transfer function:

$$HF(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + \ldots + h_n z^{-n} \qquad \text{eq. (1)}$$

where the number of taps in the filter = n + 1.

The transfer function for IIR part of Z and B filters is:

$$HI(z) = \frac{1}{1 - h_{(n+1)} z^{-1}} \cdot \qquad \text{eq. (2)}$$

The values of the user-defined coefficients (h_i) are assigned via the MPI. Each of the coefficients (h_i) is defined in the following general equation:

$$h_i = B_1 2^{-M_1} + B_2 2^{-M_2} + \ldots + B_N 2^{-M_N}, \qquad \text{eq. (3)}$$

where:

the number of shifts = $M_i \leq M_{i+1}$
sign = $B_i = \pm 1$
N = Number of CSD coefficients.

The value of h_i in (3) represents a decimal number which is broken down into a sum of successive values of:

± 1.0 multiplied by 2^{-0}, or 2^{-1}, or $2^{-2} \ldots 2^{-7} \ldots$

or

± 1.0 multiplied by 1, or 1/2, or 1/4 ... 1/128....

The limit on the negative powers of 2 is determined by the length of the registers in the ALU.

The coefficient h_i in Equation 3 can be considered to be a value made up of N binary 1s in a binary register where the leftmost part represents whole numbers, the rightmost part represents decimal fractions, and a decimal point separates them. The first binary 1 is shifted M_1 bits to the right of the decimal point, the second binary 1 is shifted M_2 bits to the right of the decimal point, the third binary 1 is shifted M_3 bits to the right of the decimal point, and so on.

Note that when M_1 is 0, the resulting value is a binary 1 in front of the decimal point, that is, no shift. If M_2 is also 0, the result is another binary 1 in front of the decimal point, giving a total value of binary "10" in front of the decimal point (i.e., a decimal value of 2.0). The value of N, therefore, determines the range of values the coefficient h_i can take; for example, if N = 3 the maximum and minimum values are ±3, and if N = 4 the values are between ±4.

Detailed Description of DSLAC Coefficients

The CSD coding scheme in the DSLAC uses a value called m_i, where m_1 represents the distance shifted right of the decimal point for the first binary 1, m_2 represents the distance shifted to the right of the *previous* binary 1, and m_3 represents the number of shifts to the right of the second binary 1. Note that the range of values determined by N is unchanged. Equation 3 is now modified (in the case of N = 4) to:

$$h_i = B_1 2^{-M1} + B_2 2^{-M2} + B_3 2^{-M3} + B_4 2^{-M4} \qquad \text{eq. (4)}$$

$$h_i = C_1 2^{-m1} + C_1 C_2 2^{-(m1+m2)} + C_1 C_2 C_3 2^{-(m1+m2+m3)} + C_1 C_2 C_3 C_4 2^{-(m1+m2+m3+m4)} \qquad \text{eq. (5)}$$

$$h_i = C_1 2^{-m1} \bullet \{1 + C_2 2^{-m2} \bullet [1 + C_3 2^{-m3} \bullet (1 + C_4 2^{-m4})]\} \qquad \text{eq. (6)}$$

where:

$M_1 = m_1$	and	$B_1 = C_1$
$M_2 = m_1 + m_2$		$B_2 = C_1 \bullet C_2$
$M_3 = m_1 + m_2 + m_3$		$B_3 = C_1 \bullet C_2 \bullet C_3$
$M_4 = m_1 + m_2 + m_3 + m_4$		$B_4 = C_1 \bullet C_2 \bullet C_3 \bullet C_4$

In the DSLAC, a coefficient h_i consists of N CSD coefficients, each being made up of 4 bits and formatted as $C_{xy}m_{xy}$, where C_{xy} is one bit (MSB) and m_{xy} is three bits. Each CSD coefficient is broken down as follows:

C_{xy} is the sign bit (0 = positive, 1 = negative).

m_{xy} is the 3-bit shift code. It is encoded as a binary number as follows:

000:	0 shifts
001:	1 shifts
010:	2 shifts
011:	3 shifts
100:	4 shifts
101:	5 shifts
110:	6 shifts
111:	7 shifts

y is the coefficient number (the "i" in h_i).

x is the position of this CSD coefficient within the h_i coefficient. It represents the relative position of the binary 1 represented by this CSD coefficient within the hi coefficient. The most significant binary 1 is represented by x = 1. The next most significant binary 1 is represented by x = 2, and so on.

Thus, $C_{13}m_{13}$ represents the sign and the relative shift position for the first (most significant) binary 1 in the 4th (h_3) coefficient.

The number of CSD coefficients, N, is limited to 4 in the GR, GX, R, X, Z, and the IIR part of the B filter, and 3 for the FIR part of the B filter. Note also that the GX filter coefficient equation is slightly different from that of the other filters:

$$h_{iGX} = 1 + h_i \qquad \text{eq. (7)}$$

Please refer to the section detailing the commands for complete details on the programming of the coefficients.

Adaptive B-Filter Overview

The DSLAC B filter is designed to work with pre-programmed coefficients or with coefficients determined by an adaptive algorithm. The adaptive algorithm can be operated in a mode where it continuously adapts (Am79C02A only) or where it adapts for a short period and then holds its value.

Operation with pre-programmed coefficients requires only the use of MPI Command #30 to feed in the coefficients. The adaptive mode uses some pre-programmed coefficients (may be any legal coefficients or zero coefficients) and generates new ones using an algorithm which, by a series of iterations, minimizes the receive signal that is echoed in the transmit signal (due to mismatches in the SLIC, hybrid, and line). Adaptation only applies to the FIR part of the filter.

In the continuous adaptation mode, the algorithm is switched on (via MPI Command #19) after a call is connected and remains on until the call ends. In this way, the B filter is continually being optimized to the received signal.

In the adapt and freeze mode, the algorithm is used only when a line is brought into service and the DSLAC is activated. The algorithm is switched on and is allowed to converge with the received signal, which is a band-limited white noise signal generated in the exchange for this purpose. The noise signal need only be injected for less than a second to yield converged coefficients. The adaptive mode is then switched off (via Command #19).

The converged coefficients may be read out of the DSLAC (using MPI Command #31) and stored for future reference. The DSLAC is now optimized for general input signals.

Adaptive Filter Programming

The purpose of the B filter is to cancel the received signal that leaks across the hybrid into the transmit path. The B filter transfer function must match (as closely as possible) the transfer function of the echo path.

There are two programmable registers associated with the adaptive B filtering. The Echo Path Gain (EPG) is a programmable value that predicts the amount of the receive signal leaking across the hybrid to the transmit path. The EPG is used as part of an algorithm which stops the adaptive filter from iterating in the presence of signals from the subscriber line (near-end talker).

The Error Level Threshold (ELT) is a programmable value that determines the trans-hybrid loss the adaptive filter will attempt to meet. The adaptive algorithm will continue to iterate until it meets the loss requirement specified by the ELT. Both the EPG and ELT values are generated by the AmSLAC-II software program. Please refer to the AmSLAC-II Technical Manual (order no. 10249A).

User Test Modes

The DSLAC supports testing by providing both digital and analog loopback paths as shown in Figure 6. In the TSA Loopback Mode, the DR input is connected to the DX output in the Time Slot Assigner circuitry. The TSA Loopback Mode is programmed via Command #21.

A different type of digital loopback is provided when the AISN register is programmed with a value of "10000." In this case, the AISN circuitry is disabled and the V_{OUT} pad is connected internally to V_{IN}. This allows the D/A and A/D converters to be included in the digital loopback test.

This mode is programmed via Command #13. Note that the signal which is connected internally from V_{OUT} to V_{IN} is also present on the V_{OUT} pin.

The V_{IN} input can be connected to the V_{OUT} output through the Z filter for analog loopback. The response of the line to low frequencies can be tested by disabling the high-pass filter. Additionally, the receive and transmit paths may be cut off.

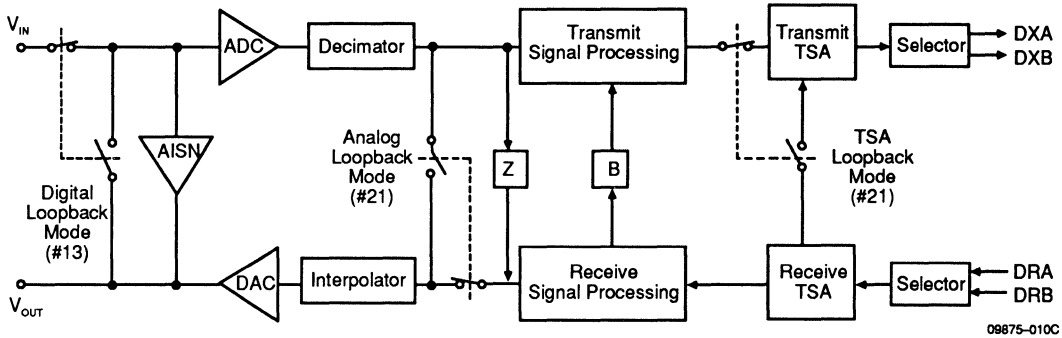


Figure 6. Test Mode Operation

APPLICATIONS

The DSLAC performs a programmable CODEC-filter function for two telephone lines. It interfaces to the telephone lines through either a transformer or an electronic SLIC such as the Am795X series devices. The DSLAC provides latched digital I/O to control and monitor two SLICs and has a 256-kHz clock output to operate the switched mode regulator in an Am795X. When several line conditions must be matched, the physical SLIC can be constant, and its characteristics (such as apparent impedance, attenuation, and hybrid balance) can be altered by programming each channel's coefficients to suit the line. For a transformer-based SLIC, the DSLAC can drive the transformer without a buffer.

Connection to a dual, PCM highway backplane is through a simple buffer chip. Several DSLACs can be bussed together into one bus interface buffer. An intelligent bus interface chip is not required because each DSLAC provides its own buffer control. The DSLAC can be controlled through the Microprocessor Interface, either by a microprocessor on the linecard or by a central processor.

Figures 7 and 8 illustrate typical Am79C02 DSLAC applications. Figure 7 shows the basic system architecture. Figure 8 illustrates the significant details of the interface to an Am795X-based SLIC and to a transformer-based SLIC.

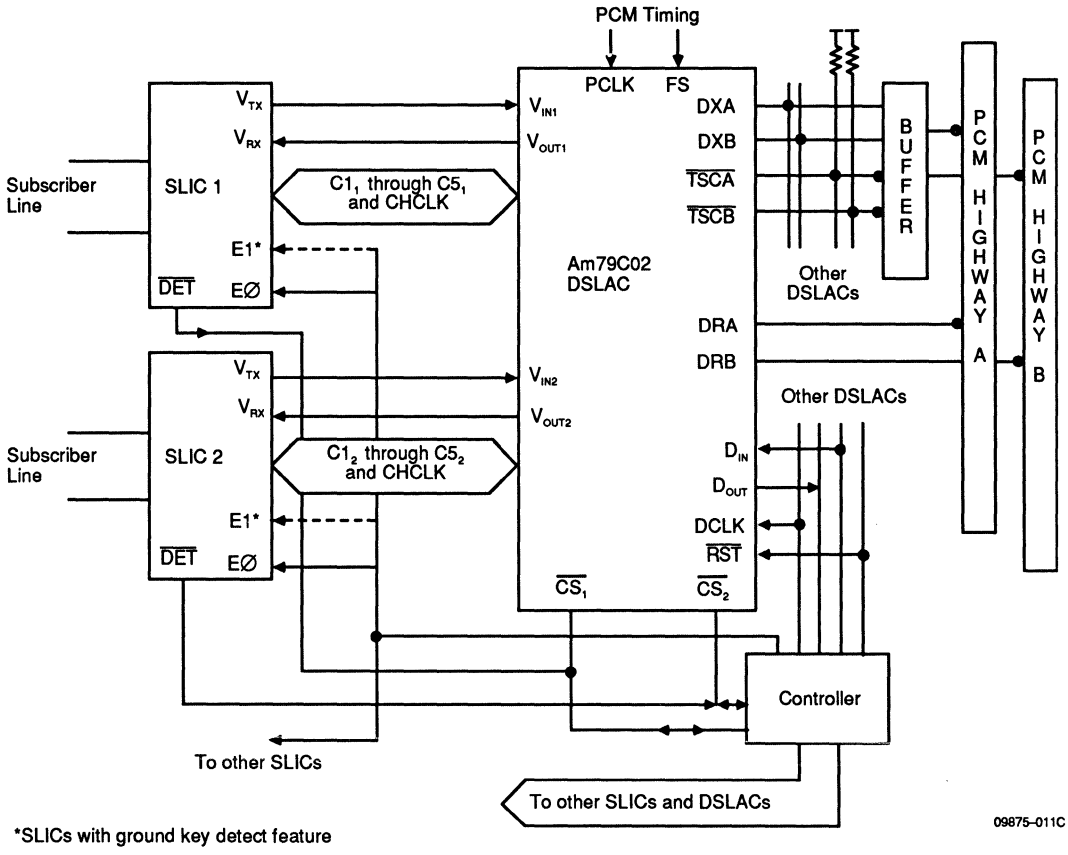


Figure 7. Basic System Architecture

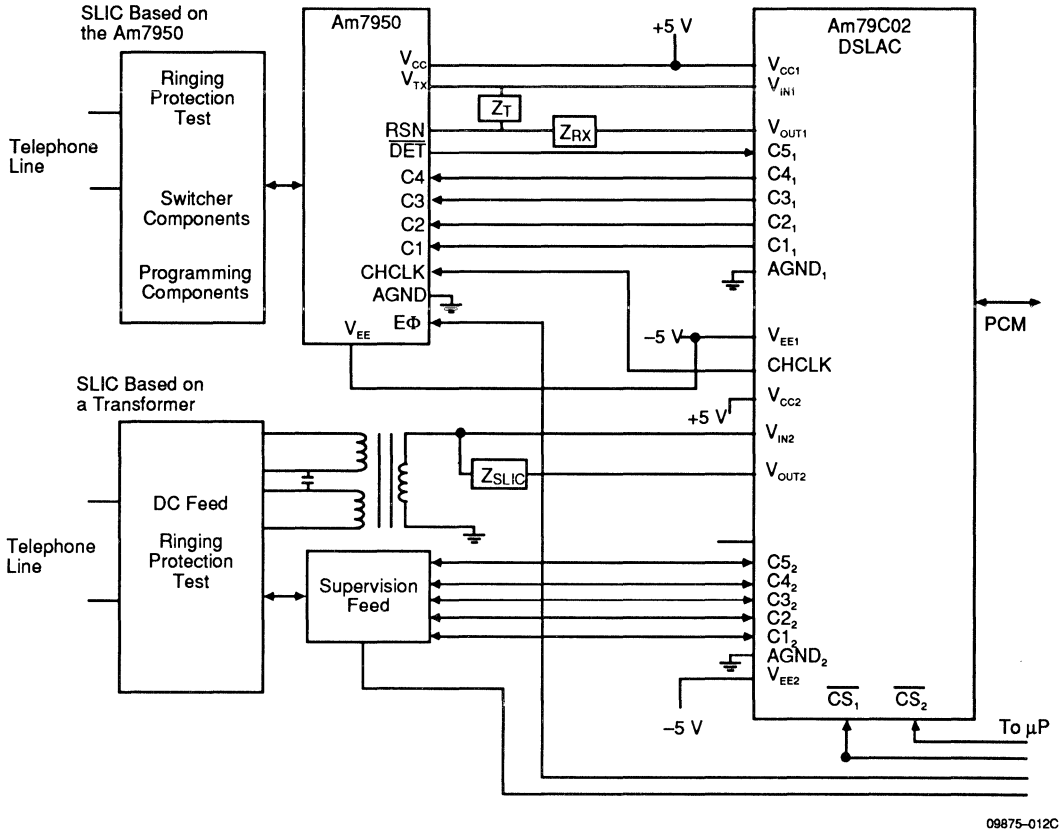


Figure 8. Typical SLIC Connections

Controlling the SLIC

SLIC Chopper Clock

The CHCLK output pin on the DSLAC drives the CHCLK inputs for Am795X series SLICs. The CHCLK output is a 256-kHz TTL-compatible signal that can drive two SLICs. It is only active when one or both channels are activated; otherwise it is held HIGH internally.

SLIC Input/Output

The DSLAC has 5 TTL-compatible I/O pins (C1 to C5) for each channel. On the 40-pin DSLAC, only C1 through C4 are available. On the 44-pin version, C5 (one for each channel) is also available and can be used for another function (for example, to control metering signal injection). The outputs are programmed using Command #15 and the status is read back using Command #16. The direction of the pins (input or output) is specified by programming the SLIC I/O direction register (Command #17).

Calculating Filter Coefficients with AmSLAC-II

AmSLAC-II is a software program which models the DSLAC, the line conditions, the SLIC, and the line card components to obtain the coefficients of the programmable filters of the DSLAC and some of the transmission performance plots.

The following parameters relating to the desired line conditions and the components/circuits used in the line card are to be provided as input to the program:

1. **Line Impedance.** The line impedance or the balance impedance of the line which is usually specified by the local PTT.
2. **Desired Impedance**. This is the desired terminating impedance at the exchange. This impedance is also specified by the local PTT.
3. **SLIC Impedance**. This is the actual terminating impedance at the exchange.
4. **GR Filter Attenuation**. This is the desired attenuation for the GR filter.
5. **GX Filter Gain**. This is the desired gain of the GX filter.
6. **Receive Buffer Transfer Function**. It is quite common to use an amplifier and/or filter between the SLIC and the SLAC in the design of the line card. The

transfer function of this amplifier/filter is called the Receive Buffer Transfer Function.

7. **Transmit Buffer Transfer Function**. Same as the Receive Buffer Transfer Function but for the Transmit path.
8. **Fuse Resistance and Coupling**. This is the value of the Fuse Resistance and the Coupling capacitor used in the line card.
9. **Two-Wire Return Loss Template**. The Two-Wire Return Loss Template is usually specified by the local PTT.
10. **Four-Wire Return Loss Template**. The Four-Wire Return Loss Template is usually specified by the local PTT.

The output from the AmSLAC-II program includes the coefficients of the GR, GX, Z, R, X, B, and EPG filters as well as transmission performance plots of (1) two-wire return loss, (2) receive and transmit path frequency response, and (3) four-wire return loss.

The software supports the use of the AMD Am795X series SLICs or a transformer SLIC, or allows entry of the transfer functions describing the behavior of any type of SLIC (hybrid).

Systems for Customer Evaluation

The DSLAC Low Noise Evaluation Board is designed to demonstrate the high performance capabilities of the DSLAC. The board is used to evaluate the DSLAC available in a 40-pin DIP package.

The SLAC/DSLAC Computer Interface Board provides a friendly, computer-driven interface to control up to two DSLAC Low Noise Evaluation Boards or SLAC Low Noise Boards. The Computer Interface Board allows an IBM®-compatible PC-XT™ or PC-AT™ to control a SLAC, DSLAC, and a SLIC via its serial port. The board is designed to operate with the DSLACIF software program which runs on the PC. A block diagram of a typical lab setup is shown in Figure 9.

The Computer Interface Board can also interface to a Hewlett-Packard 3779 series PMA or a Wandel and Goltermann (W&G) PCM-4. These PCM Channel Measurement Sets are used to measure the quality of signal transmission through the DSLAC.

An RS-232C serial port on the SLAC/DSLAC Computer Interface Board is designed to plug directly into a serial port on the back of a PC. The DSLACIF program which controls the Computer Interface Board will operate on an IBM PC-XT, PC-AT, or compatible computer containing at least 1 serial port and having at least 512 KB of memory. The program is capable of running from a floppy disk (360 KB) or from a hard disk. The DSLACIF program is completely menu driven and an extensive on-line HELP facility is available.

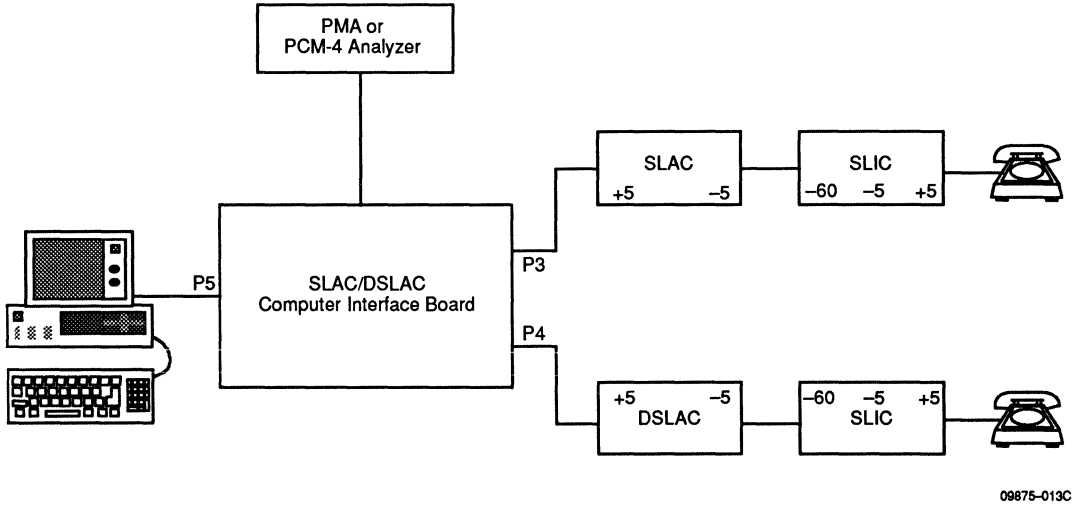


Figure 9. Evaluation System Block Diagram

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	$-60 \le T_A \le +125°C$
Ambient Operating Temperature	$-40 \le T_A \le +85°C$
Ambient relative humidity (noncondensing)	5% to 100%
V_{CCA1} with respect to AGND	−0.4 V to +6.0 V
V_{CCA2} with respect to AGND	−0.4 V to +6.0 V
V_{CCD1} with respect to $DGND_1$	−0.4 V to +6.0 V
V_{CCD2} with respect to $DGND_2$	−0.4 V to +6.0 V
V_{CCP} with respect to PGND	−0.4 V to +6.0 V
V_{EE1} with respect to AGND	+0.4 V to −6.0 V
V_{EE2} with respect to AGND	+0.4 V to −6.0 V
V_{IN} with respect to V_{CCA} (V_{EE} = −5 V)	+0.4 V to −10.0 V
V_{IN} with respect to V_{EE} (V_{CCA} = +5 V)	−0.4 V to +10.0 V
Any other pin with respect to $DGND_1$	−0.4 V to V_{CC}

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Analog Supply V_{CCA1}, V_{CCA2}	+5.0 V ±5%
Digital Supply V_{CCD1}, V_{CCD2}, V_{CCP}	+5.0 V ±5%
Analog Supply V_{EE1}, V_{EE2}	−5.0 V ±5%
$DGND_1$, $DGND_2$, PGND	0 V
$AGND_1$, $AGND_2$	±50 mV
Ambient Temperature	$0 \le T_A \le +70°C$
Ambient Relative Humidity	15% to 85%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise noted

Typical values are for T_A = 25°C and nominal supply voltages. Minimum and maximum specifications are over the temperature and supply voltage ranges shown in "Operating Ranges."

Parameter Symbol	Parameter Descriptions	Min.	Typ.	Max.	Unit
V_{IL}	Input LOW Voltage	−0.5		0.8	V
V_{IH}	Input HIGH Voltage	2.0		V_{CC}	V
I_{IL}	Input Leakage Current			±10	µA
V_{OL}	Output LOW Voltage (I_{OL} = −2 mA)			0.4	V
V_{OH}	Output HIGH Voltage (I_{OH} = 400 µA)	2.4			V
V_{OLTSC}	Output LOW Voltage on $\overline{TSCA}$, $\overline{TSCB}$ (I_{OL} = 14 mA)			0.4	V
I_{OL}	Output Leakage Current (H_i-Z State)			±10	µA
V_{IR}	Analog input Voltage Range (AX = 0 dB) (AX = 6.02 dB)			±3.1466 ±1.5733	V V
V_{IOS} I_{IL} (V_{IN})	Offset Voltage allowed on V_{IN} Input Leakage Current on V_{IN}			±35 ±10	mV µA
Z_{OUT}	V_{OUT} output impedance		1	10	ohms
I_{OUT}	V_{OUT} output current (f < 3400 Hz) (Note 1)			±6.3	mA
V_{OR}	V_{OUT} Voltage Range (AR = 0 dB) (AR = −6.02 dB)			±3.1466 ±1.5733	V V
V_{OOS}	V_{OUT} Offset Voltage (AISN off)			±15	mV
V_{OOSA}	V_{OUT} Offset Voltage due to AISN			±15	mV
LIN_{AISN}	Linearity of AISN circuitry (input = 0 dBm0)			±1/4	LSB
PD	Power Dissipation both channels active 1 channel active both channels inactive (Note 2)		180 120 12	300 160	mW mW mW
I_{CC}	Total +5 V current, both channels active 1 channel active both channels inactive (Note 2)		24 18 2.5		mA mA mA
I_{EE}	Total −5 V current, both channels active 1 channel active both channels inactive (Note 2)		10 5 0.05		mA mA mA
C_I	Input Capacitance (Digital)		15		pF
C_O	Output Capacitance (Digital)		15		pF
PSSR	Power Supply Rejection Ratio (1.02 kHz, 100 mV_{RMS}, either supply or path, GX = GR = 0dB)	40			dB

Notes: 1. When the DSLAC is in the inactive mode, the analog output will present a 0 V output level through an ~3K resistor.
2. Power Dissipation in the inactive mode is measured with all digital inputs at $V_{ih} = V_{CC}$ and $V_{il} = V_{SS}$ and with no load connected to V_{OUT1} or V_{OUT2}.

Transmission Characteristics

When the gain in the receive path is set at 0 dB, an 813-Hz PCM sine wave input with level 0 dBm0 will correspond to a nominal RMS voltage of 1.55 volts for μ-law and 1.555 volts for A-law at the analog output. When the gain in the transmit path is set at 0 dB, an 813-Hz sine wave signal with a nominal RMS voltage of 1.55 volts for μ-law and 1.555 volts for A-law will correspond to a level of 0 dBm0 at the digital output.

When relative levels (dBm0) are used in any of the following transmission specifications, the specification holds for any setting of the AX + GX gain from 0 to 12 dB and the AR + GR loss from 0 to 12 dB. Performance specification for settings of the AX + GX gain from 12 to 18 dB and the AR + GR loss from 12 to 18 dB will be determined as the device is characterized.

These performance specifications are valid for the commercial temperature range device only. The specifications for the industrial temperature range device will be released after full characterization.

Gain Stability

For a 0 dBm0 813-Hz (A-law) or 1014-Hz (μ-law) sine wave signal, the gains in the transmit and in the receive path (with B = 0, Z = 0 & X = R = 1) will not deviate from their ideal value by more than ±0.2 dB.

The variation of the digital to digital loop gain (when the analog input and output ports are connected together) or the analog to analog loop gain (when the digital input and output ports are connected together) will be within ±0.2 dB. The above specifications apply with reference to aging, temperature, and supply voltage variations within the specifications of the "Operating Ranges".

Attenuation Distortion

The attenuation of the signal in either path is nominally independent of the frequency. The deviations from nominal attenuation will stay within the limits shown. The reference frequency is 813 Hz and the signal level is 0 dBm0. The deviation is less than ±0.125 dB for 300 Hz < f < 3000 Hz.

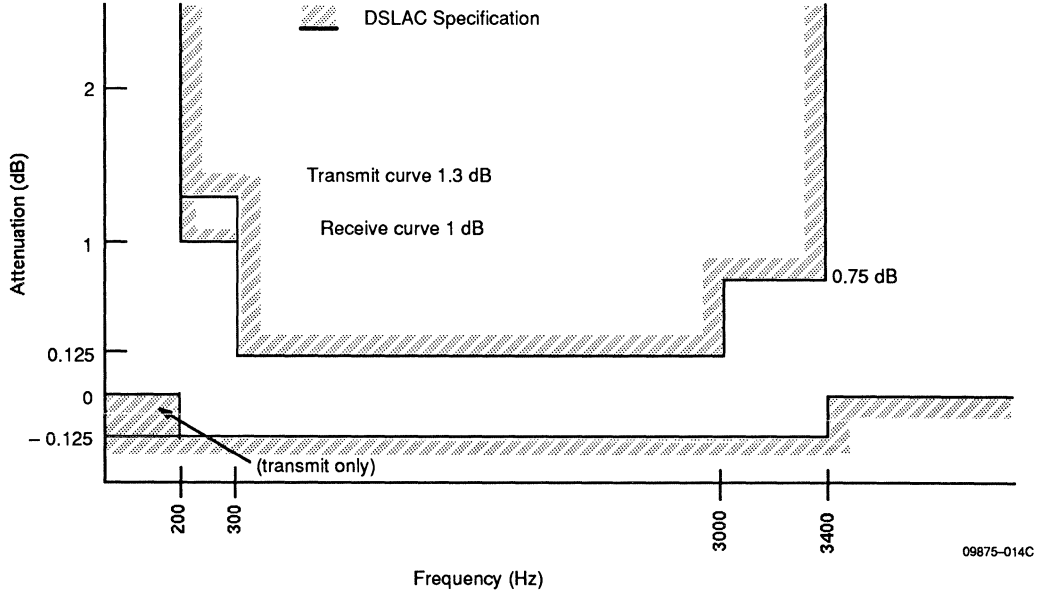


Figure 10. Attenuation Distortion (Single Ended)

Group Delay Distortion

For either transmission path, the group delay distortion is within the limits shown. The minimum value of the group delay is taken as the reference. The signal level should be 0 dBm0.

Group Delay

The Group Delay spec is defined as the sum of the minimum values of the group delays for the transmit and the receive paths when the transmit and receive time slots are identical and the B, X, R, and Z filters are disabled. For PCLK frequencies of greater than 1.536 MHz, the group delay is less than 630 µs. For PCLK frequencies of less than 1.025 MHz, the group delay is less than 695 µs. (At PCLK frequencies between these two values, the group delay may vary from one cycle to the next.)

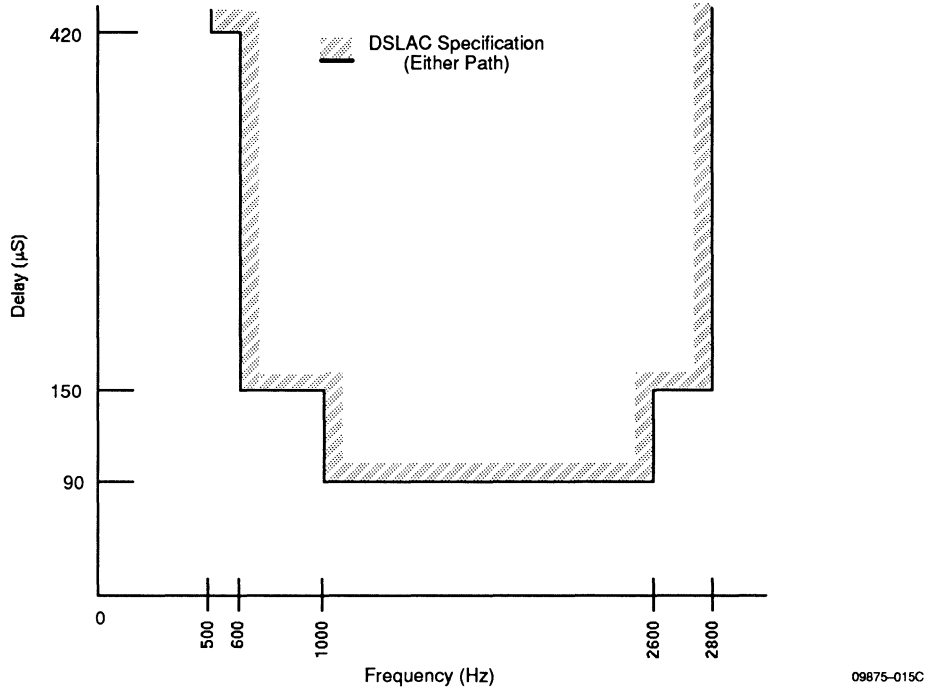


Figure 11. Group Delay Distortion

Discrimination Against Out-of-Band Input Signals

When an out-of-band sine wave signal with frequency f and level A is applied to the analog input, the level of any frequency component below 4 kHz at the digital output, caused by the out-of-band signal, is at least the specified dB level below the level of a signal at the same output originating from an 813-Hz sine wave signal with a level of A dBm0 also applied to the analog input. The minimum specifications are:

Frequency of out-of-band signal	Amplitude of out-of-band signal	Level below A
16.6 Hz < f < 60 Hz	–25 dBm0 < A < 0 dBm0	20 dB
60 Hz < f < 100 Hz	–25 dBm0 < A < 0 dBm0	10 dB
3400 Hz < f < 4600 Hz	–25 dBm0 < A < 0 dBm0	see Figure 12
4600 Hz < f < 72 kHz	–25 dBm0 < A < 0 dBm0	32 dB
4600 Hz < f < 100 kHz	A = 0 dBm0	35 dB

The attenuation of the waveform below amplitude A between 3400 Hz and 4600 Hz is given by the formula:

$$\text{Attenuation (dB)} = 14 - 14 \sin \frac{\pi(4000-f)}{1200}$$

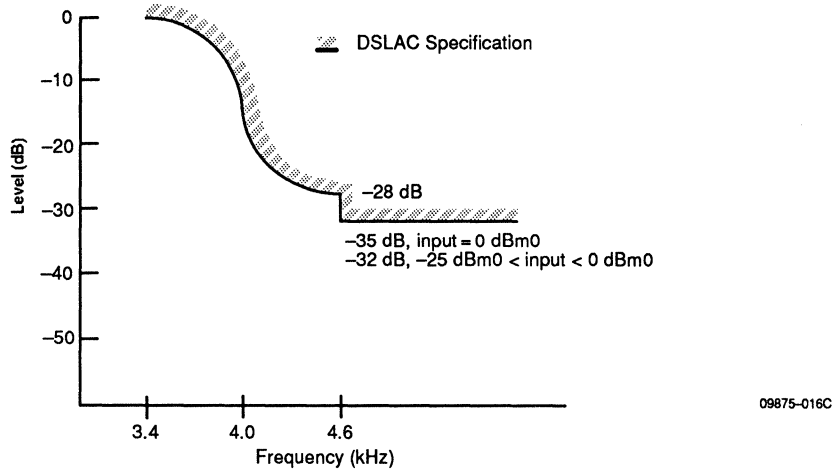


Figure 12. Discrimination Against Out-of-Band Signals

Spurious Out-of-Band Signals at the Analog Output

With PCM code words representing a sine wave signal in the range of 300 Hz to 3400 Hz at a level of 0 dBm0 applied to the digital input, the level of the spurious out-of-band signals between 4.6 kHz and 24 kHz at the analog output is less than –32 dBm0. The level of spurious out-of-band signals between 24 kHz and 90 kHz is given by:

$(23 - 40 \bullet \log_{10} f)$,

where f is the frequency in kHz.

Between 90 kHz and 1 MHz, the signal at the analog output is less than –55 dBV. With code words representing any sine wave signal in the range 3.4 kHz to 4.0 kHz at a level of 0 dBm0 applied to the digital input, the level of the signals at the analog output are below the limits in Figure 13. The amplitude of the Spurious Out-of-Band signals between 3400 Hz and 4600 Hz is given by the formula:

$$A = -14 - 14 \sin \frac{\pi (f - 4000)}{1200}$$

Harmonic Distortion

The output signal level, at any single frequency in the range of 300 Hz to 3400 Hz, other than that due to an applied 0 dBm0 sine wave signal with frequency f in the same frequency range, is less than –46 dBm0. With f swept between 0 to 300 Hz and 3400 to 12 kHz, any generated output signals other than f are less than –28 dBm0. This specification is valid for either transmission path.

Intermodulation Distortion

Two sine wave signals of different frequencies f_1 and f_2 (not harmonically related) in the range 300 Hz to 3400 Hz and of equal levels in the range –4 dBm0 to –21 dBm0 do not produce any

$2 \bullet f_1 - f_2$

products having a level greater than –42 dB relative to the level of the two input signals.

A sine wave signal in the frequency band 300 Hz to 3400 Hz with input level –9 dBm0 and a 50-Hz signal with input level –23 dBm0, will not give any intermodulation products exceeding a level of –56 dBm0. These specifications are valid for either transmission path.

Idle Channel Noise

When the signal at the analog input is zero and the digital output (DXA or DXB) is connected to the digital input (DRA or DRB), the maximum levels of the noise measured at the analog output are:

Weighted noise:	–68 dBm0p
Unweighted noise (300–3400 Hz):	–55 dBm0

When the signal at the analog input is zero, the maximum level of the noise measured at the digital output does not exceed –68 dBm0p (A-law) or 19 dBrnc0 (μ-law). When PCM code words representing zero volts are applied to the digital input, the maximum level of the noise measured at the analog output does not exceed –78 dBm0p (A-law) or 12 dBrnc0 (μ-law). No single frequency component in the range above 3800 Hz may exceed a level of –55 dBm0.

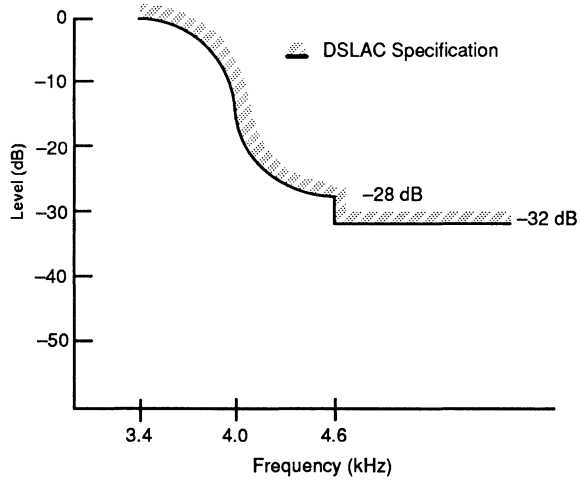


Figure 13. Spurious Out-of-Band Signals

Crosstalk

Transmit to Receive crosstalk within a channel. The crosstalk level at the analog output due to a 0 dBm0 sine wave signal in the frequency range 300 Hz to 3400 Hz, applied to the analog input, is less than –75 dBm0.

Receive to Transmit crosstalk within a channel. The crosstalk level at the digital output due to a 0 dBm0 sine wave signal in the frequency range 300 Hz to 3400 Hz, applied to the digital input, is less than –75 dBm0.

Transmit to Transmit crosstalk between channels. With a 0 dBm0 sine wave signal in the frequency range 300 Hz to 3400 Hz applied to the analog input of one channel, the level at the digital output of the other channel does not exceed –76 dBm0.

Transmit to Receive crosstalk between channels. The crosstalk level at the analog output of one channel due to a 0 dBm0 sine wave signal in the frequency range 300 Hz to 3400 Hz, applied to the analog input of the other channel, will be less than –78 dBm0.

Receive to Transmit crosstalk between channels. The crosstalk level at the digital output of one channel due to a 0 dBm0 sine wave signal in the frequency range 300 Hz to 3400 Hz, applied to the digital input of the other channel, will be less than –76 dBm0.

Receive to Receive crosstalk between channels. The crosstalk level at the analog output of one channel due to a 0dBm0 sine wave signal in the frequency range 300Hz to 3400 Hz, applied to the digital input of the other channel, will be less than –78 dBm0.

Variation of Gain with Input Level

The gain deviation relative to the gain at –10 dBm0 is within the limits shown for either transmission path when the input signal is a noise signal (for example, CCITT Rec. O.131).

The gain deviation relative to the gain at –10 dBm0 is within the limits shown for either transmission path when the input is a sine wave signal in the frequency range 700 Hz to 1100 Hz (excluding submultiples of 8 kHz).

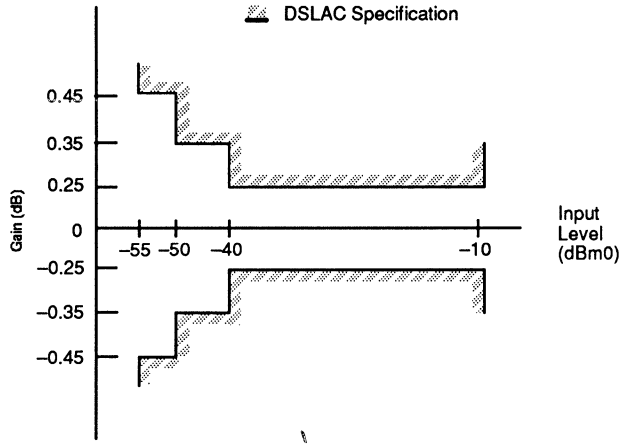


Figure 14. Gain Tracking with Noise Input

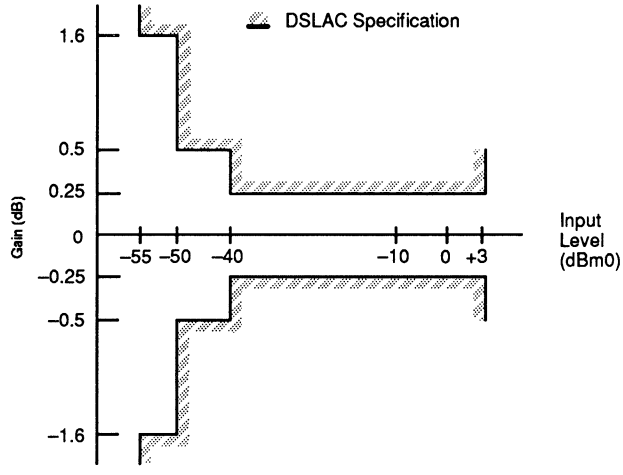


Figure 15. Gain Tracking with Tone Input

Total Distortion, Including Quantizing Distortion

The signal-to-total distortion ratio will exceed the limits shown for the receive path when the input signal is a noise signal (for example, CCITT Rec. O.131). The transmit path specification is 1 dB less than that shown for the receive path.

The signal-to-total distortion will exceed the limits shown for either transmission path when the input is a sine wave signal in the frequency range 700 Hz to 1100 Hz (excluding submultiples of 8 kHz).

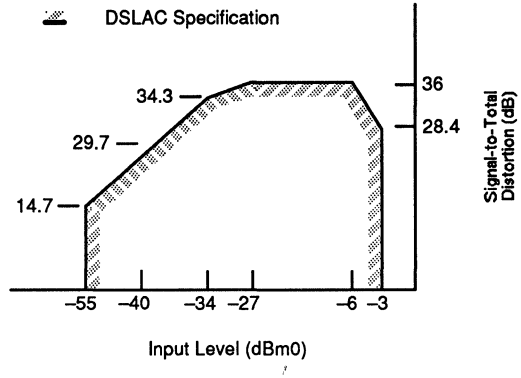


Figure 16. Total Distortion with Noise Input (Receive Path)

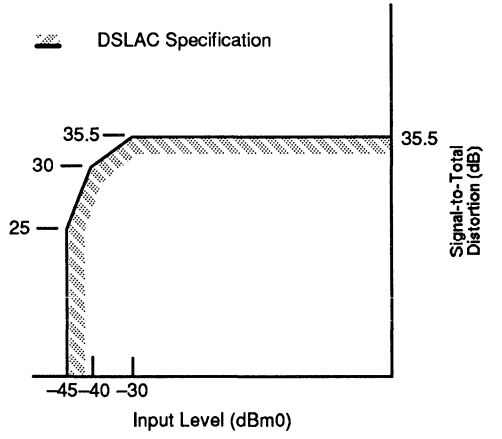


Figure 17. Total Distortion with Tone Input (Both Paths)

Overload Compression

Figure 18 shows the acceptable region of operation for input signal levels above the reference input power (0 dBm0).

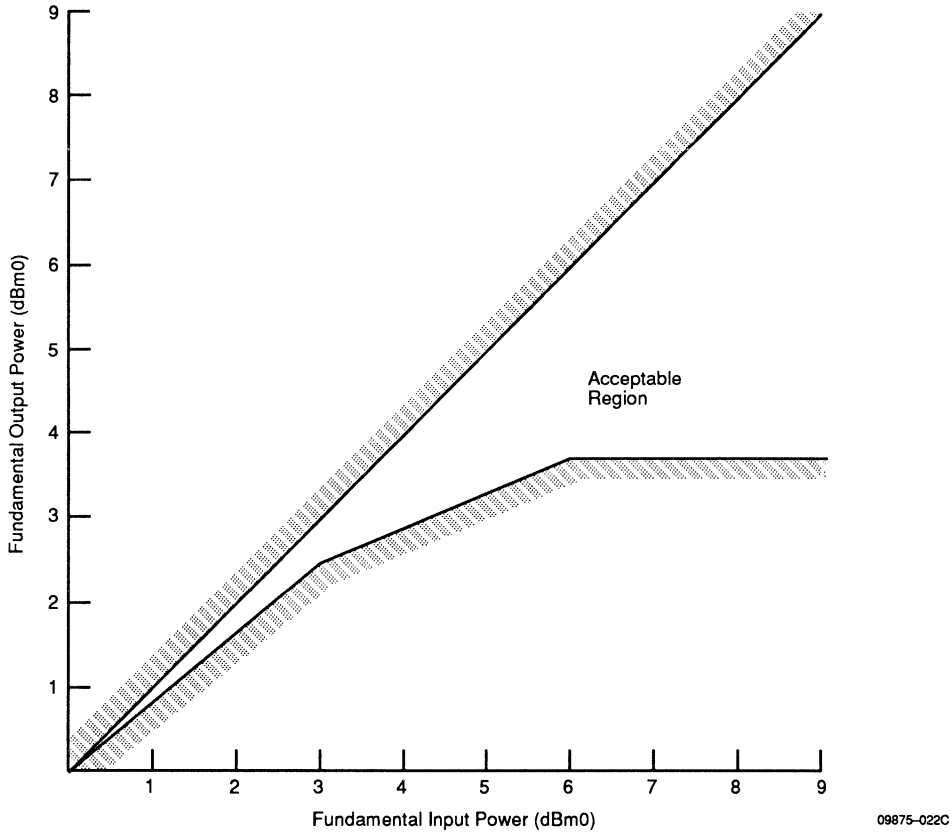


Figure 18. Overload Compression

SWITCHING CHARACTERISTICS over operating range unless otherwise noted

Microprocessor Interface

Min. and Max. values are valid for all digital outputs with a 150 pF load, except C1 to C5 with a 30 pF load. Pullup resistors of 360 ohms are attached to $\overline{TSCA}$ and $\overline{TSCB}$.

No.	Symbol	Parameter	Min.	Typ.	Max.	Units
1	t_{DCY}	Data Clock Period	244			ns
2	t_{DCH}	Data Clock HIGH Pulse Width (Note 1)	97			ns
3	t_{DCL}	Data Clock LOW Pulse Width (Note 1)	97			ns
4	t_{DCR}	Rise Time of Clock			25	ns
5	t_{DCF}	Fall Time of Clock			25	ns
6	t_{ICSS}	Chip Select Setup Time, Input Mode	70		t_{DCY}	ns
7	t_{ICSH}	Chip Select Hold Time, Input Mode	0		t_{DCH} –20	ns
8	t_{ICSL}	Chip Select Pulse Width, Input Mode		$8t_{DCY}$		ns
9	t_{ICSO}	Chip Select off Time, Input Mode (Note 7)		5		µs
10	t_{IDS}	Input Data Setup Time	30			ns
11	t_{IDH}	Input Data Hold Time	30			ns
12	t_{OLH}	SLIC Output Latch Valid	20		1000	ns
13	t_{OCSS}	Chip Select Setup Time, Output Mode	70		t_{DCY}	ns
14	t_{OCSH}	Chip Select Hold Time, Output Mode	0		t_{DCH} –20	ns
15	t_{OCSL}	Chip Select Pulse Width, Output Mode		$8t_{DCY}$		ns
16	t_{OCSO}	Chip Select Off Time, Output Mode (Note 7)		5		µs
17	t_{ODD}	Output Data Turn On Delay (Note 5)			50	ns
18	t_{ODH}	Output Data Hold Time	10			ns
19	t_{ODOF}	Output Data Turn off Delay			50	ns
20	t_{ODC}	Output Data Valid	10		50	ns

PCM Interface

No.	Symbol	Parameter	Min.	Typ.	Max.	Units
21	t_{PCY}	PCM Clock Period (Note 2)	0.122		7.8125	µs
22	t_{PCH}	PCM Clock HIGH Pulse Width	48		3890	ns
23	t_{PCL}	PCM Clock LOW Pulse Width	48		3890	ns
24	t_{PCF}	Fall Time of Clock			15	ns
25	t_{PCR}	Rise Time of Clock			15	ns
26	t_{FSS}	FS Setup Time	25		t_{PCY}–50	ns
27	t_{FSH}	FS Hold Time	50			ns
28	t_{TSD}	Delay to $\overline{TSC}$ Valid (with Programmable Delay) (Note 3)	5 30		80 80	ns ns
29	t_{TSO}	Delay to $\overline{TSC}$ off (with Programmable Delay) (Note 6)	5 30		80 80	ns ns
30	t_{DXD}	PCM Data Output Delay (with Programmable Delay) (Note 4)	3 30		70 150	ns ns
31	t_{DXH}	PCM Data Output Hold Time (with Programmable Delay) (Note 4)	5 30		70 150	ns ns
32	t_{DXZ}	PCM Data Output Delay to High-Z (with Programmable Delay) (Note 4)	5 30		70 150	ns ns
33	t_{DRS}	PCM Data Input Setup Time	25		70	ns
34	t_{DRH}	PCM Data Input Hold Time	5		150	ns

Master Clock

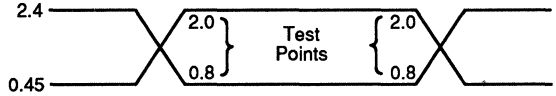
For 2.048 MHz ± 100 ppm or 4.096 MHz ± 100 ppm operation:

No.	Symbol	Parameter	Min.	Typ.	Max.	Units
35	t_{MCY}	Master Clock Period (2.048 MHz)	488.23	488.28	488.33	ns
		Master Clock Period (4.096 MHz)	244.11	244.14	244.17	ns
36	t_{MCR}	Rise Time of Clock			15	ns
37	t_{MCF}	Fall Time of Clock			15	ns
38	t_{MCH}	MCLK HIGH Pulse Width (2.048 MHz)	200			ns
		MCLK HIGH Pulse Width (4.096 MHz)	80			ns
39	t_{MCL}	MCLK LOW Pulse Width (2.048 MHz)	200			ns
		MCLK LOW Pulse Width (4.096 MHz)	80			ns

Notes
1. DCLK may be stopped in the HIGH or LOW state indefinitely without loss of information. If DCLK is stopped in the HIGH state, $\overline{CS}$ can subsequently make any number of transitions without activating the Microprocessor Interface logic.
2. The maximum allowed PCM clock frequency is 8.192 MHz. The actual PCM clock rate is dependent on the number of channels allocated within a frame. The minimum clock frequency is 128 kHz. A PCLK of 1.544 MHz may easily be used for standard U.S. transmission systems.
3. TSC is delayed from FS by a typical value of $N \cdot t_{PCY}$, where N is the value stored in the time/clock-slot register.
4. There is a special conflict detection circuitry which will prevent high power dissipation from occurring when the DXA or DXB pins of two DSLACs are tied together and one DSLAC starts to transmit before the other has gone into a high impedance state.
5. The first data bit is enabled on the falling edge of $\overline{CS}$ or on the falling edge of DCLK, whichever occurs last.
6. t_{TSO} is defined as the time at which the output achieves the open circuit condition.
7. The DSLAC requires 40 cycles of the 8-MHz internal clock (5 µs) between SIO operations. If the MPI is being accessed while the MCLK input is not active, a Chip Select Off time of 20 µs is required.

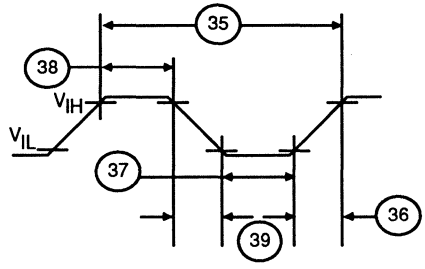
SWITCHING WAVEFORMS

Input and Output Waveforms for AC Tests



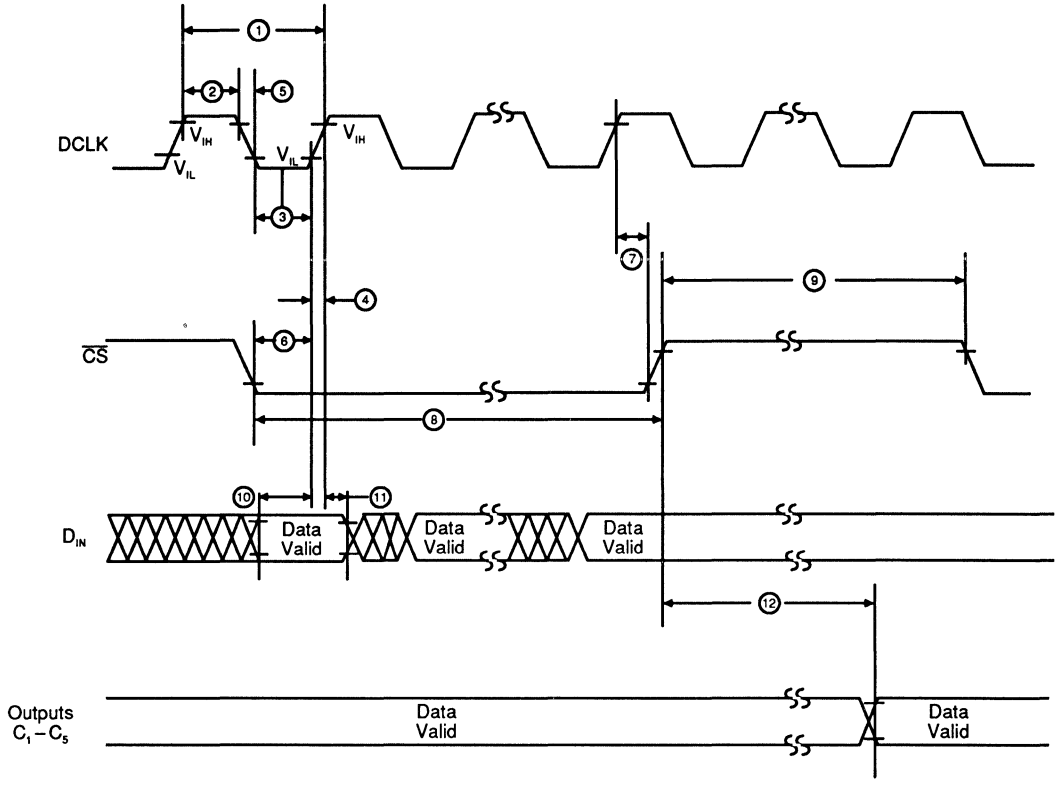
09875-023C

Master Clock Timing



09875-024C

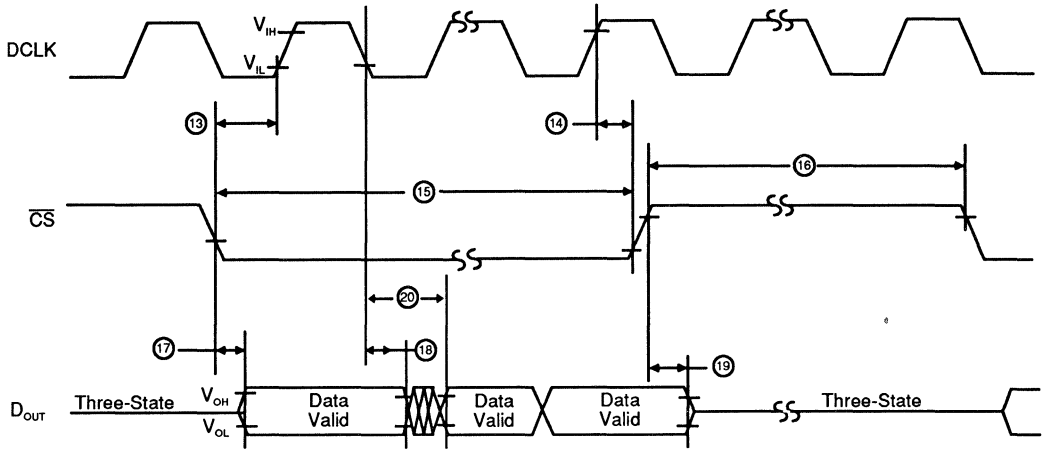
Microprocessor Interface (Input Mode)



09875-025C

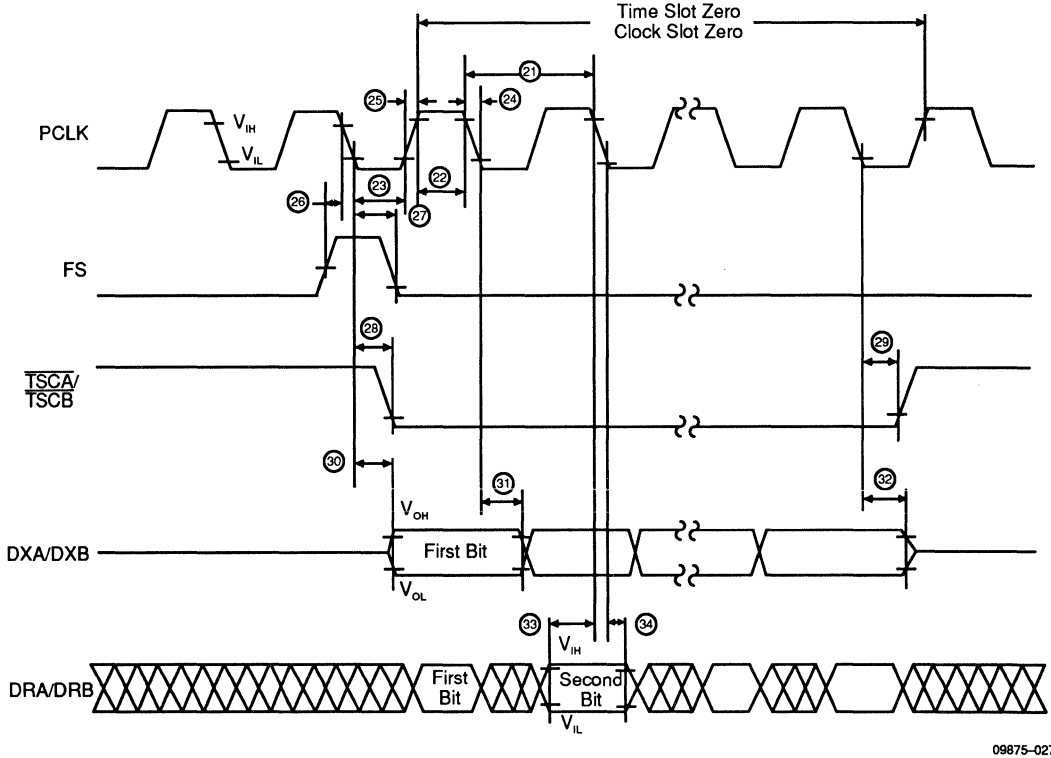
4

Microprocessor Interface (Output Mode)



09875-026C

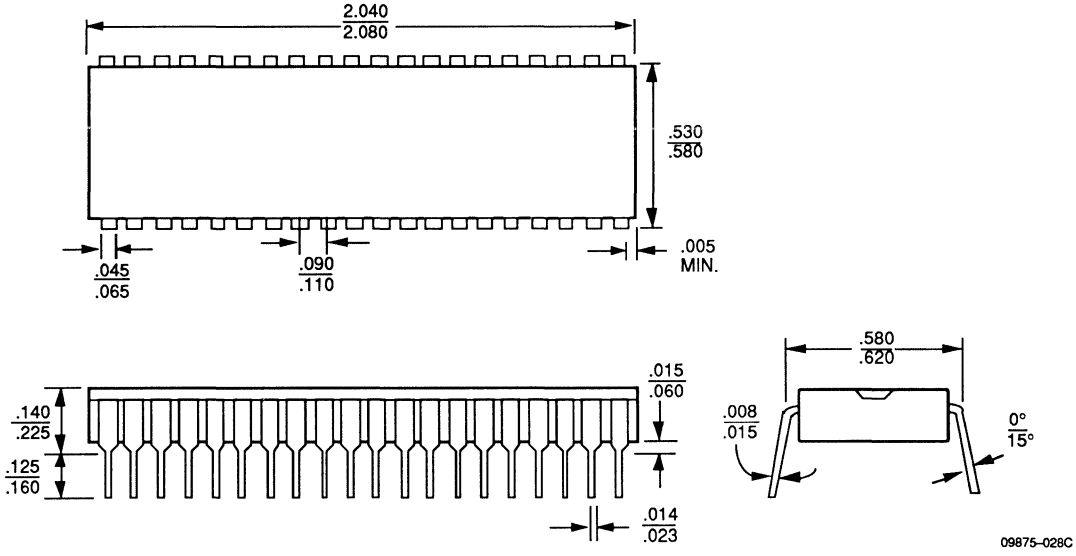
PCM Highway Timing



09875-027C

PHYSICAL DIMENSIONS

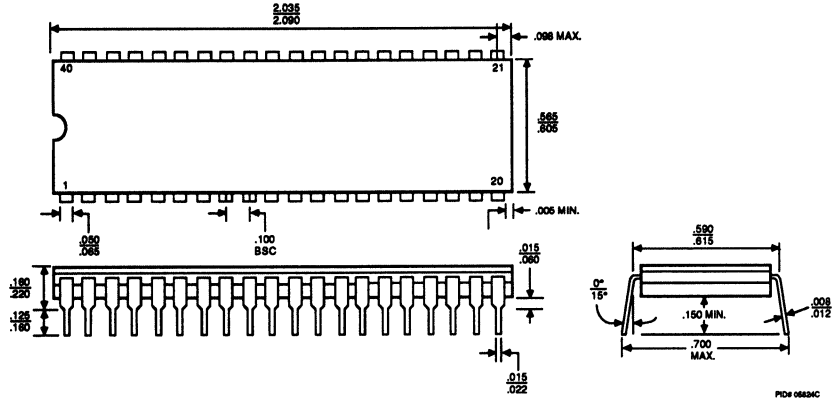
PD 040



4

PHYSICAL DIMENSIONS

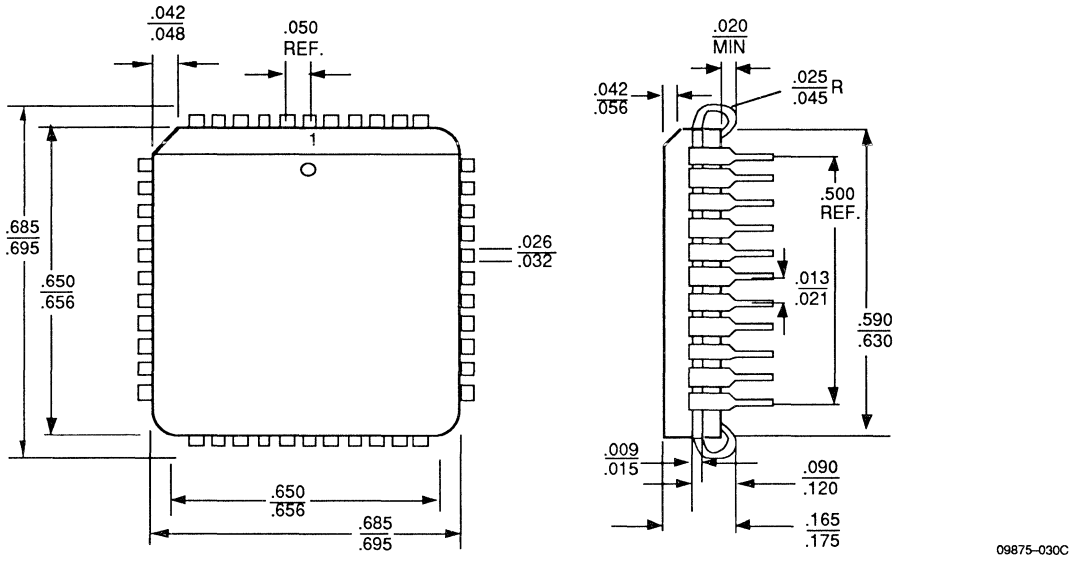
CD 040



09875-029C

PHYSICAL DIMENSIONS

PL 044



4

Advanced
Micro
Devices

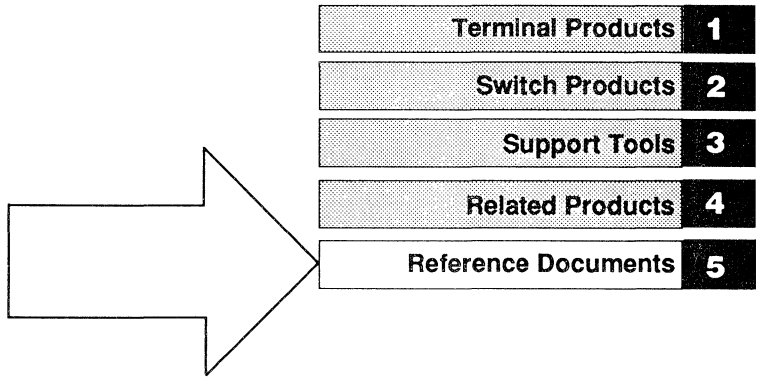


Table of Contents

Advanced
Micro
Devices

ISDN Systems Engineering Application Notes

5

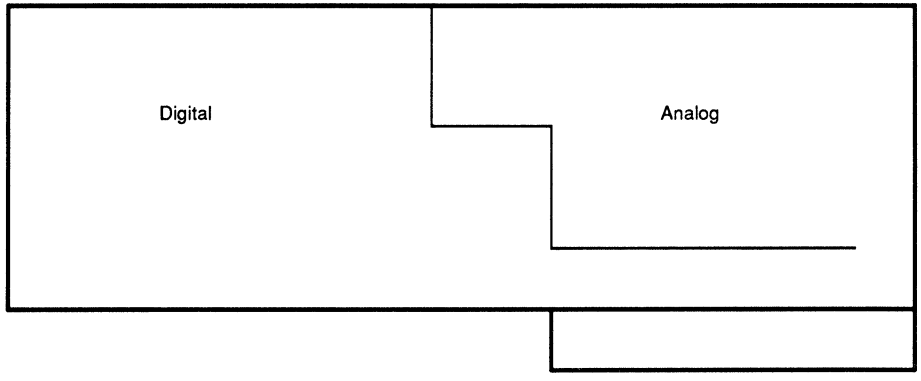
Publication #	Rev.	Amendment	Issue Date:
12557	A	/0	7/89

Am79C30A DSC Layout Hints

The Am79C30A integrates high-speed digital logic and sensitive audio circuitry into a single device. Although audio transmission tests are demanding, observance of basic guidelines of component selection and layout will yield excellent performance.

SEPARATE ANALOG AND DIGITAL AREAS

In the example below, a card containing digital circuitry as well as a Digital Subscriber Controller (DSC) has been designed for use in a PC. Four Printed Circuit Board (PCB) layers are used: two for signals and the other two for power and ground. The Vcc and Vss planes of the PCB include cuts which separate the board into two areas that are joined near the edge connector. The DSC and all analog circuits are placed on the right side of the power-plane cuts, as close as possible to the S interface and telephone connectors on the card edge. Since DSC analog and digital functions are manufactured on a single substrate, it is necessary to use a common power supply for the analog and digital Vcc pins of the DSC PLCC to avoid the possibility of excessive current flow through the DSC. The power-plane cuts illustrated in the diagram below reduce the coupling of digital switching noise into the supply path of the DSC and analog components.



UTILIZE EFFECTIVE DECOUPLING PRACTICES

Capacitor selection—For an application such as the one on the previous page, it is recommended to use 0.1-µF high-frequency ceramic capacitors on each IC, and in parallel with 10.0-µF tantalum capacitors near the edge connector. It is important to check for special decoupling requirements for individual ICs, since some functions (for example, dynamic RAM controllers) require bulk decoupling near the package to handle sudden switching loads. For the DSC, it is necessary to include bulk decoupling capacitors if the loudspeaker outputs drive a heavy current load.

Capacitor placement—Individual decoupling capacitors should be placed as close as possible to the supply pins of the IC, with minimum lead length. This is because at higher frequencies the inductances of the capacitor and IC leads are not negligible, and sudden surges of supply current will result in noise spikes on the supply pins.

Decouple all supply pins—On devices such as the DSC in a PLCC package, it is helpful to provide decoupling for each supply pin. Analog Vcc should be decoupled to Analog Vss, and Digital Vcc to Digital Vss, using high-frequency 0.1-µF ceramic capacitors.

Publication #	Rev.	Amendment	Issue Date:
12607	A	/0	7/89

PLAN AN EFFECTIVE BOARD LAYOUT

Care in routing—In an analog environment, it is particularly important to keep signal traces as short as possible, and to route analog traces away from high-frequency signals such as clock lines. Traces that are connected to ground may be used to help isolate key signals. The longer a trace becomes, the greater will be its tendency to act like an antenna or transmission line. It is well worth the time to plan the location of components to minimize the length of critical traces. For example, if a low-level signal to be amplified enters a board through a connector, it is preferable to place the amplifier as close as possible to the connector itself. Although the noise floor may be low, any noise picked up would be amplified along with the signal.

Care in placement—The overall noise performance of a board is also improved by taking precautions in the placement of components. For all devices with crystal oscillators, it is recommended to place the crystal and load capacitors as close as possible to the crystal pins of the device. For the DSC, it is also helpful to place the RC network for the Main Audio Processor (MAP) as close as possible to the DSC CAP1 and CAP2 pins to reduce the coupling of noise into the device.

CCITT I.430 Tutorial

CCITT Recommendation I.430 defines the Layer 1 characteristics of the user-network interfaces at the ISDN S and T reference points. The Am79C30A Digital Subscriber Controller is fully compliant with I.430, and it is instructive to examine I.430 in detail to understand its implications in both IC and system design.

INTRODUCTION: THE ROLE OF I.430

It is important to note that although CCITT Recommendation I.430 is an international specification, the requirements for network connection vary from country to country and are still under development. It is unlikely that a single comprehensive international specification will evolve in the near future; a more likely scenario is that individual countries will at first adopt their own extensions of I.430. For example, in the United States the American National Standards Institute (ANSI) is developing a standard that basically conforms with I.430, but also "specifies individual departures from CCITT I.430 that reflect the domestic environment of the United States." These departures are numerous, including for example differences in power feeding, hazardous voltage protection, and use of the multiframing bits. These differences are in general extensions to or clarification of I.430, as opposed to being incompatibilities. The creation of a single world standard is of necessity a slow process, and it is inevitable that there will be such temporary differences as we converge towards a true global network.

An additional factor in the ISDN environment is certification with individual switch vendors. In the world of analog networks, a telephone or modem device would be tested for compliancy with the specifications of the governing telecommunications authorities (for example, FCC Part 68 in the United States) whereupon the device could be connected to the analog network. In the ISDN environment, switch vendors have independently developed products while the specifications evolved, resulting in the need to undergo compliancy tests with each individual switch vendor for the physical, link, and signaling layers. Fortunately, these specifications are generally equivalent to or slightly relaxed from I.430 at the physical layer, with the majority of differences occurring in signaling. An example of such a specification is AT&T 5D5-900-301.

CCITT Recommendation I.430 is thus seen to be a basic document for the development of equipment for the ISDN S and T reference points, but for the foreseeable future designers will need to look to telecommunications regulatory agencies and ISDN switch vendors for additional requirements.

GENERAL OVERVIEW

Sections 1 and 2 of I.430 present a general description of the reference configurations applicable to I.430, and an overview of the services to be provided to Layer 2 and the management entity (ME). Layer 2 refers to the next higher layer of the OSI software model, which is responsible for certain physical link functions. The ME services all OSI layers with a wide range of system functions, including event notification, error handling, maintenance loops, memory management, and initialization. It is important to note that there is currently no finalized international standard for the functions of the ME or its interfaces to the various OSI layers. Work is underway on a series of CCITT Recommendations (Q.940, 941, 942) in an attempt to standardize ME functions and interfaces; however, at the current time, designers are implementing unique and potentially incompatible versions. Of particular interest in Section 2 are the primitives to be passed between Layer 1 and other entities. For instance, Layer 2 or the ME. I.430 specifically states that these primitives represent the logical exchange of information and control, and are not intended to constrain the implementation of entities or interfaces. No assumption is made about the definition of the other layers or entities to which I.430 is to be connected. For instance, I.430 defines activation/deactivation primitives to be passed between Layers 1 and 2, yet the Q.921 Layer 2 specification makes no mention of what to do with these primitives. The point to be understood here is that I.430 does not specify a software interface, either to the management entity or to Layer 2.

MODES AND CONFIGURATIONS

Sections 3, 4, and 8 of I.430, together with Annex A, describe the modes of operation and wiring configurations of the user-network interface. In "point-to-point" operation a single TE and a single NT are interconnected, whereas in point-to-multipoint operation multiple TEs may be simultaneously connected to a single NT in either a "short passive" or "extended passive" bus configuration. The timing recovery requirements for the NT side vary in the different wiring configurations, and I.430 defines four classes of NT that operate on some or all of the three wiring configurations. NTs may be designed

Publication #	Rev.	Amendment	Issue Date:
12608	A	/0	7/89

for short passive bus operation only, for both point-to-point and passive bus, for extended passive bus only, or for point-to-point only.

In the short passive bus wiring configuration, up to eight TEs may be connected at random points along the cable. If the NT device uses fixed timing, it samples data at regular fixed intervals even though pulses arrive from the various terminals with different transmission delays. This implies that the maximum time difference between pulses arriving at the NT must be less than a bit period. The round-trip delay for NTs configured for passive bus only operation may vary from 10 to 14 microseconds, the lower value of 10 microseconds being derived from the 2-bit delay through the TE plus the maximum negative phase deviation. With fixed timing, it is the round-trip delay that limits the maximum cable length of the short passive bus configuration and not the attenuation of the cable itself. This delay corresponds to a maximum distance on the order of 100–200 meters for the short passive bus configuration.

The simplest configuration is point-to-point, where a TE is present at one end of the cable and an NT at the other. In this case, the maximum transmission distance is determined by cable attenuation and receiver performance, not round-trip delay. The accepted target distance for point-to-point operation is 1 kilometer. Adaptive NT timing is required, which uses a phase-lock loop to compensate for the round-trip delay. NTs that are configured for point-to-point operation only must accommodate round-trip delays, which may vary from 10 to 42 microseconds. NTs that are configured for either point-to-point or short passive bus operation must allow delays ranging from 10 to 42 microseconds for point-to-point, and 10 to 13 microseconds for short passive bus.

In the extended passive bus configuration, the maximum cable distance is increased over the short passive bus by grouping the terminals together at the far end of the cable, thereby restricting the range of the differential round-trip delay between any two terminals. In this case the NT again uses adaptive timing. Although the calculations in Annex 4 indicate up to four TEs in this configuration, it is left up to individual regulatory authorities to define detailed configurations. The distance objective of the extended passive bus configuration is 1 kilometer, but will vary greatly with the wiring of the TEs. NTs configured for extended passive bus operation only must accommodate round-trip delays ranging from 10 to 42 microseconds.

Note that in all three configurations there is only one terminating resistor at each end of the cable, regardless of the number of terminals connected. The presence of extra terminating resistors will artificially degrade distance performance.

The final diagram of Annex A shows a star configuration NT1, in which point-to-multipoint operation is realized using point-to-point wiring. In this case, the NT1 must do extra buffering and processing of the D-channel echo bits to handle contention by all the TEs for the single D channel on the network side of the NT1.

An interesting point in Section 4 is that the correct polarity of wiring must be maintained in the TE to NT direction for point-to-multipoint configurations. This is because all D-channel MARKs must appear as LOW MARKs to ensure proper operation of the D-channel priority mechanism.

FUNCTIONAL CHARACTERISTICS

Section 5 of I.430 summarizes the basic functions of an I.430-compatible interface.

Two B Channels each provide a 64-kb/s full-duplex channel for user information. No restriction on the content or usage of these channels is defined, other than the requirement that they do not contain circuit switching information for the network.

Bit timing recovery from the data stream is necessary since the four-wire interface does not provide any separate data clocks.

Octet timing provides the TE and NT with an 8-kHz frame-sync signal.

Frame alignment allows the TE and NT to correctly identify and recover the time division multiplexed overhead bits and data channels.

The D channel provides a 16-kb/s full-duplex channel for user data packets and network signaling packets.

Power feeding is included as an I.430 function, but implementations and requirements vary greatly depending upon the regulatory environment.

Deactivation and activation are procedures defined to direct the TE and NT into or out of low-power idle states.

Connection and disconnection of a TE at the interface is defined as the appearance and disappearance respectively of power to the TE. These events are significant in the higher-level procedures that assign a unique TEI (Terminal Endpoint Identifier) number to each TE during system initialization.

The formats of the 48-bit frames are then defined, which are different for each direction of transmission. Figure 3 of I.430 is appended to the end of this application note for reference.

FRAME STRUCTURE AND ALIGNMENT

In the I.430 pseudo-ternary line code, a binary ONE (SPACE) is represented by no line signal, whereas a binary ZERO (MARK) is represented by either a positive or negative pulse. Transmitted binary ZEROs must alternate in polarity to maintain "DC balance" on the line,

with the exception of intentional "code violations" in the frame where successive ZEROs do not alternate in polarity.

Both types of frames begin with a framing bit F, which is always a positive MARK, followed by a balance bit L, which is always a negative MARK. The last MARK of the previous frame is guaranteed to be positive, meaning that the F bit is a code violation. The first MARK following the frame balance bit is required to be a negative MARK, which produces a second code violation. In the NT to TE direction, the auxiliary framing bit F_A or the N bit guarantee the required violation within 14 bits or less from the F bit if the data bits between L and F_A do not. If the multiframing mode (to be discussed in detail shortly) is enabled, the NT will transmit a SPACE in the F_A position and N will be a MARK. If multiframing is not enabled, the NT will send a MARK in the F_A position and N will be a SPACE.

In the TE to NT direction, the F_A bit will guarantee a second code violation within 13 bits or less from the F bit if multiframing is not enabled. If multiframing is enabled, only 4 out of 5 frames are guaranteed to satisfy the 13 or less criterion, but the framing procedures are tolerant of this missing violation.

In the NT to TE direction, frame synchronization is achieved when 3 consecutive frames with valid pairs of code violations satisfying the 14-bit criterion are detected. Loss of synchronization occurs when a period of time equivalent to 2 consecutive frames elapses without detection of violations that satisfy the 14-bit criterion, whereupon the TE must cease transmission immediately.

In the TE to NT direction, frame synchronization is achieved when 3 consecutive frames with valid pairs of code violations satisfying the 13-bit criterion are detected. If all F_A bits are ZERO, loss of synchronization occurs when a period of time equivalent to 2 consecutive frames elapses without detection of violations that satisfy the 13-bit criterion. Otherwise, 3 frames are required to indicate loss of synchronization. For both types of frames, the next 8 bits following the F/L pair constitute the first octet of data for the B1 channel. In the NT to TE direction, these bits are followed by an E bit and a D bit. The E bit is the echo channel for the D bits; upon receiving a D bit, the NT echoes it back to the TE(s) in the next E-bit position so the TEs may compare it to their own transmission for collision detection purposes. The first D bit is followed by the A bit, which is set when the interface is activated and synchronized. The A bit is followed by the auxiliary framing pair F_A and N, which are in turn followed by the first octet of data for the B2 channel. The second occurrence of an E and D bit is next, followed by the multiframing bit M. The remainder of the frame consists of the second octet of B1 data, the third E and D bit pair, the "S" bit (for further study), the second octet of B2 data, the fourth and final E and D bit pair, and the frame balance bit L. Note that the final frame balance bit L serves the dual purpose of DC balancing the entire frame as well as guaranteeing that the last MARK of the frame is a positive MARK.

The frame structure in the TE to NT direction is considerably different. The frame begins with an F/L pair as before, followed by the first octet of B1 data. However, the D bits are preceded by balance bits L so that the D bit may be restricted to either SPACE (no line signal) or negative MARK. This prevents the collision of negative MARKs with positive MARKs in the multipoint situation, which could result in a composite waveform resembling SPACE. Since the spacing condition is interpreted as line IDLE, the efficiency of the D-channel access procedures would be reduced. A key feature to note is that the L bits force the first MARK of each data octet to be negative MARK. This simplifies correct DC balancing during multipoint operation when the B1 and B2 data sources come from different TEs; if the TEs did not know which polarity of MARK to begin with, they could introduce erroneous code violations.

The first occurrence of a D bit is followed by a balance bit L, the F_A bit, and another L bit. The L bit preceding the F_A bit guarantees that the F_A bit can be a negative MARK, and the next L bit guarantees that the first MARK of the following B2 data octet can be a negative MARK. The rest of the frame follows a similar pattern for the remaining D and B bits. As before, the final L bit DC balances the frame and ensures that the F bit of the next frame will be a code violation.

D-CHANNEL PROCEDURES

The D-channel procedures allow multiple TEs to share the D channel in a multipoint configuration, or single TEs to operate in a point-to-point configuration.

When a TE has no Layer 2 frames to transmit, it sends binary ONEs on the D channel, which results in no line signal. When the NT has no Layer 2 frames to transmit, it may send binary ONEs or repeat HDLC flags. An LAPD flag is the octet "01111110," and defines the beginning or ending of an LAPD packet. A flag that closes a packet may also define the beginning of a new packet. To prevent the occurrence of the flag pattern in the actual data, the transmitter must include a "zero insertion" function that automatically inserts a ZERO after each sequence of five contiguous logical ONEs. Similarly, the receiver must include a "zero deletion" function that automatically deletes any ZEROs that follow five contiguous logical ONEs. Any occurrence of six logical ONEs in the received data will therefore indicate a flag. As discussed in the previous section, the NT will echo a received D bit in the next available E-bit position towards the TEs. The TEs will monitor the E-bit position, both to gain access to the D channel as well as to check for collision with other TEs. A TE that intends to transmit data must monitor the E bit, counting the number of consecutive binary ONEs. Once a count threshold corresponding to the priority of

the TE is reached, it may begin to transmit data. The priority level may be fixed for a particular TE, or it may be programmed after system powerup. The priority count is incremented following successful transmission of a packet to allow other TEs equal access to the channel. Typical priority counts range from eight and nine for signaling packets to ten and eleven for data packets. Once a TE begins D-channel transmission, it must monitor the E bit and compare it with its last transmitted D bit. If they match, transmission continues, otherwise a collision or line error has occurred and the TE must cease transmission and return to counting binary ONEs.

ACTIVATION/DEACTIVATION SEQUENCES

The I.430 specification defines a set of F states, numbered from F1 through F8, to indicate the state of the TE during activation and deactivation procedures. Similarly, a set of G states numbered from G1 through G4 are used to indicate the state of the NT. Section 6.2 of I.430 is included for reference at the end of this application note and defines the precise meaning of each F and G state. Also included is Table 2 of I.430, which describes the format of the "INFO" signals exchanged by TE and NT during the initialization sequence. Tables 3 and 4 of I.430 are appended to this note, and define state tables which indicate the state transitions to be made based upon events such as reception of particular INFO signals.

The meaning and sequence of states are clearly defined in the specification, but a few points require special note. Attention is drawn to the difference between the inactive and deactivated TE states. In the inactive state there is no power, and therefore nothing is happening in the TE. Consequently, the Am79C30A only provides indication of states F2 through F8. In the deactivated state, the TE has been placed in a low power state, but will monitor the line for an indication to activate or may request activation itself if requested to do so by higher TE software layers. Note also that a TE may request activation, but may not request deactivation. A TE distinguishes INFO2 from INFO4 by the state of the "A," or activation bit.

The "handshake" sequence of the F and G states is relatively forgiving, since the D-channel protocol must be satisfied before any actual data can be transmitted. For instance, in multipoint operation it is possible to have overlapping but unsynchronized INFO1 signals from TEs to NT. A single INFO1 signal consists of a continuous pattern of positive MARK, negative MARK, and six SPACEs. The composite signal may not resemble an individual INFO1, but the NT will respond with an INFO2 pattern and the individual TEs will then synchronize.

MULTIFRAMING

Multiframing is a mechanism that provides for the transmission of a Q bit in the TE to NT direction every fifth frame. The information is intended for the NT only; there is no requirement for the NT to transmit any information in the upstream direction to the exchange termination. It is intended to provide an additional low-speed channel from the TE to the NT. At the current time, the procedures and usage of the multiframe capabilities are not widely implemented. Section 7 of the ANSI document does, however, define some uses of the multiframing capability for maintenance functions.

If multiframing is supported, the Q-bit positions are identified by the setting of the F_A bit to a binary ONE in the NT to TE direction every fifth frame. This allows the TEs to synchronize their Q transmissions, preventing collision of F_A bits from one TE with the Q bits from another. However, there is no mechanism for Q-bit collision detection recovery between TEs. An additional feature is the setting of the M bit in the NT to TE direction every 20 frames, creating a multiframe structure that supports the transmission of 4-bit characters from TE to NT. A single S bit is transmitted each frame in the NT to TE direction.

The fact that systems have been designed while the standards were evolving can lead to situations where equipment may not respond exactly as expected. The case in point is when an NT transmits a logical "1" in the F_A position to an Am79C30A during the activation sequence. Although the NT should not consistently send "1" in this position, cases have been observed where this condition has indeed occurred. If multiframing is disabled in the Am79C30A, it will echo the received F_A bit back to the NT. If the B1 channel is disabled, or sends $FF, the TE and NT will fail to synchronize since the NT will not see a second code violation within the first 13 bits of the frame. A simple workaround for this situation is to enable multiframing (set Am79C30A MF register bit 0 to "1") and force the F_A response to be logical "0" (set Am79C30A MFQB register bit 4 to "0"). This guarantees that the F_A bit in the TE to NT direction will provide a code violation if the B1 channel data does not.

The Am79C30A provides full support for all multiframing functions. If multiframing is enabled, the DSC will monitor the received F_A and M bits to establish multiframe synchronization, buffer received S bits, and buffer Q bits for transmission synchronized to the received multiframe.

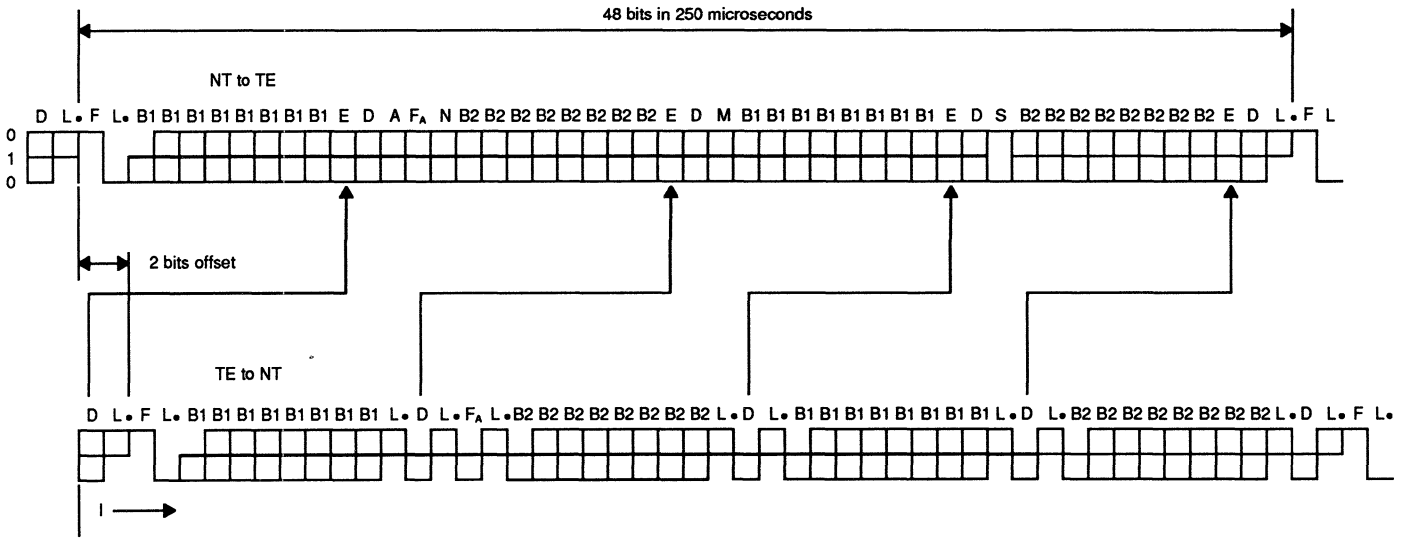
ELECTRICAL CHARACTERISTICS

Recommendation I.430 describes in detail the electrical testing requirements of the S and T interfaces. Areas covered include the allowable phase jitter and deviation allowed from TE input to output, the allowable jitter output of the NT, and the waveforms and test conditions under which these measurements are made. The NT and TE input and output impedance requirements are strictly specified, as is the shape of the output waveforms and performance objectives in the presence of noise. A

detailed treatment of electrical compliance is given in a companion application note on S and T Interface Circuitry.

POWER FEEDING ISSUES

A key section of I.430 deals with the requirements for power feeding and consumption. Tight restrictions on power consumption are defined for telephones that are designated to operate in "restricted," or emergency power modes. For normal operation the power budget is more lenient, but a designer who wishes to add features to the basic telephone will need to plan power consumption carefully. These power targets create considerable challenges for IC and systems designers alike, and have a great impact on terminal design. Power issues are discussed in detail in companion notes on S and T Interface Power Issues.



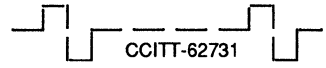
F = framing bit
L = DC balancing bit
D = D-Channel bit
E = D-echo-channel bit
F_A = Auxiliary framing bit (see section 6.3)

N = bit set to a binary value $N = \overline{F}_A$ (NT to TE) (see section 6.3)
B1 = bit within B Channel 1
B2 = bit within B Channel 2
A = bit used for activation
S = the use of this bit is for further study
M = Mulitiframing bit

Notes:
1. Dots demarcate those parts of the frame that are independently DC-balanced.
2. The F_A bit in the direction TE to NT is used as a Q-bit in every fifth frame if the Q-Channel capability is applied (see Section 6.3.3).
3. The nominal 2-bit offset is as seen from the TE (I_A in Figure 2/I.430). The corresponding offset at the NT may be greater due to delay in the interface cable and varies by configuration.

Figure 3/I.430. Frame Structure at Reference Points S and T

Table 2/I.430. Definition of INFO Signals (note 1)

Signals from NT to TE		Signals from TE to NT	
INFO 0	No signal	INFO 0	No signal
		INFO 1 (note 2)	A continuous signal with the following pattern: Positive ZERO, negative ZERO, six ONEs.  Nominal bit rate = 192 kbit/s
INFO 2 (note 3)	Frame with all bits of B, D, and D echo channels set to binary ZERO. Bit A set to binary ZERO. N and L bits set according to the normal coding rules.		
INFO 4 (note 3)	Frames with operational data on B, D, and D echo channels. Bit A set to binary ONE.	INFO 3	Synchronized frames with operational data on B and D channels.

Notes: 1. For configurations where the wiring polarity may be reversed (see Section 4.3), signals may be received with the polarity of the binary ZEROs inverted. All NT and TE receivers should be designed to tolerate wiring polarity reversals.

2. TEs that do not need the capability to initiate activation of a deactivated I.430 interface (that is, TEs required to handle only incoming calls) need not have the capability to send INFO 1. In all other respects, these TEs shall be in accordance with Section 6.2. It should be noted that in the point-to-multipoint configuration, more than one TE transmitting simultaneously will produce a bit pattern as received by the NT and different from that described above, that is, two or more overlapping (asynchronous) instances of INFO 1.

3. During the transmission of INFO 2 or INFO 4, the F_A bits and the M bits from the NT may provide the Q-bit pattern designation as described in Section 6.3.3.

Table 3/I.430. Activation/deactivation Layer 1 Finite State Matrix for TEs Powered from Power Source 1 or 2

State Name / Number / Event / INFO	Inactive F1 INFO 0	Sensing F2 INFO 0	Deacti-vated F3 INFO 0	Awaiting signal F4 INFO 1	Identi-fying input F5 INFO 0	Synchro-nized F6 INFO 3	Activated F7 INFO 3	Lost framing F8 INFO 0
Power on and detection of Power 2 (notes 2 and 3)	F1	—	—	—	—	—	—	—
Loss of power (note 2)	—	F1	MPH-II(d) F1	MPH-II(d) MPH-DI PH-DI F1	MPH-II(d) MPH-DI PH-DI F1	MPH-II(d) MPH-DI PH-DI F1	MPH-II(d) MPH-DI PH-DI F1	MPH-II(c) MPH-DI PH-DI F1
Disappearance of Power S (note 3)	—	F1	MPH-II(d) F1	MPH-II(d) MPH-DI PH-DI F1	MPH-II(d) MPH-DI PH-DI F1	MPH-II(d) MPH-DI PH-DI F1	MPH-II(d) MPH-DI PH-DI F1	MPH-II(c) MPH-DI PH-DI F1
PH Act Req	/	|	ST.T3 F4	|	|	—	|	—
Expiry T3	/	/	—	MPH-DI PH-DI F3	MPH-DI PH-DI F3	MPH-DI PH-DI F3	—	—
Rec. INFO 0	/	MPH-II(c) F3	—	—	—	MPH-DI PH-DI F3	MPH-DI PH-DI F3	MPH-DI PH-DI MPH-EI1 F3
Rec. any signal (note 1)	/	—	—	F5	—	/	/	—
Rec. INFO 2	/	MPH-II(c) F6	F6	/	F6	—	MPH-EI1 F6	MPH-EI1 F6
Rec. INFO 4	/	MPH-II(c) PH-AI MPH-AI F7	PH-AI MPH-AI F7	/	PH-AI MPH-AI F7	PH-AI MPH-AI MPH-EI2 F7	—	PH-AI MPH-AI MPH-EI1 F7
Lost framing	/	/	/	/	/	MPH-EI1 F8	MPH-EI1 F8	—

Notations

—	No change, no action
/	Impossible situation
|	Impossible by the definition of the Layer 1 service
a, b; F*n*	Issue primitives "a" and "b" and then go to state "F*n*"
PH-AI	Primitive PH-Activate Indication
PH-DI	Primitive PH-Deactivate Indication
MPH-AI	Primitive MPH-Activate Indication
MPH-DI	Primitive MPH-Deactivate Indication
MPH-EI1	Primitive MPH-Error Indication reporting error
MPH-EI2	Primitive MPF-Error Indication reporting recovery from error
MPH-II(c)	Primitive MPH-Information Indication (connected)
MPH-II(d)	Primitive MPH-Information Indication (disconnected)
ST.T3	Start timer T3
Power S	Power Source 1 or Power Source 2

Table 4/I.430. Activation/deactivation Layer 1 Finite State Matrix for NTs

State Name / Number / Event / INFO	Inactive G1 INFO 0	Pending activation G2 INFO 2	Active G3 INFO 4	Pending deactivation G4 INFO 0
Power on and detection of power S	F2			
PH-Act Req	Start timer T1 G2	—	|	Start timer T1 G2
MPH-Deact Req	|	Start timer T2 PH-DI; G4	Start timer T2 PH-DI; G4	|
Expiry T1 (note 1)	—	Start timer T2 PH-DI; G4	/	—
Expiry T2 (note 2)	—	—	—	G1
Receiving INFO 0	—	—	MPH-DI; MPH-EI G2 (note 3)	G1
Receiving INFO 1	Start timer T1 G2	—	/	—
Receiving INFO 3	/	Stop timer T1 PH-AI; MPH-AI G3 (note 4)	—	—
Lost framing	/	/	MPH-DI; MPH-EI G2 (note 3)	—

Notations

—	No state change
/	Impossible by the definition of peer-to-peer physical layer procedures or system internal reasons
|	Impossible by the definition of the physical layer service
a, b; G*n*	Issue primitives "a" and "b" and then go to state "G*n*"
PH-AI	Primitive PH-Activate Indication
PH-DI	Primitive PH-Deactivate Indication
MPH-AI	Primitive MPH-Activate Indication
MPH-DI	Primitive MPH-Deactivate Indication
MPH-EI	Primitive MPH-Error Indication

Primitives are signals in a conceptual queue and will be cleared on recognition, while the INFO signals are continuous signals that are available all the time.

Notes: 1. Timer 1 (T1) is a supervisory timer that has to take into account the overall time to activate. This time includes the time it takes to activate both the ET-NT and the NT-ET portion of the customer access. ET is the exchange termination.

2. Timer 2 (T2) prevents unintentional reactivation. Its value is 25 ms ≤ value ≤ 100 ms. This implies that a TE has to recognize INFO 0 and to react on it within 25 ms. If the NT is able to unambiguously recognize INFO 1, then the value of timer 2 may be 0.

6.2 Activation/deactivation

6.2.1 Definitions

6.2.1.1 TE states

6.2.1.1.1 STATE F1 (inactive): In this inactive state the TE is not transmitting. In the case of locally powered TEs that cannot detect the appearance/disappearance of Power Source 1 or 2, this state is entered when local power is not present. For TEs that can detect Power Source 1 or Power Source 2, this state is entered whenever loss of power (required to support all TEI functions) is detected, or when the absence of power from Source 1 or 2, whichever power source is used for determining the connection status, is detected.

6.2.1.1.2 STATE F2 (sensing): This state is entered after the TE has been powered on but has not determined the type of signal (if any) that the TE is receiving.

6.2.1.1.3 STATE F3 (deactivated): This is the deactivated state of the physical protocol. Neither the NT nor the TE is transmitting.

6.2.1.1.4 STATE F4 (awaiting signal): When the TE is requested to initiate activation by means of a PH-Activate Request primitive, it transmits a signal (INFO 1) and waits for a response from the NT.

6.2.1.1.5 STATE F5 (identifying input): At the first receipt of any signal from the NT, the TE ceases to

transmit INFO 1 and awaits identification of signal INFO 2 or INFO 4.

6.2.1.1.6 STATE F6 (synchronized): When the TE receives an activation signal (INFO 2) from the NT, it responds with a signal (INFO 3) and waits for normal frames (INFO 4) from the NT.

6.2.1.1.7 STATE F7 (activated): This is the normal active state with the protocol activated in both directions. Both the NT and the TE are transmitting normal frames.

6.2.1.1.8 STATE F8 (lost framing): This is the condition when the TE has lost frame synchronization and is awaiting re-synchronization by receipt of INFO 2 or INFO 4, or deactivation by receipt of INFO 0.

6.2.1.2 NT states

6.2.1.2.1 STATE G1 (inactive): In this deactivated state, the NT is not transmitting.

6.2.1.2.2 STATE G2 (pending activation): In this partially active state, the NT sends INFO 2 while waiting for INFO 3. This state will be entered on request by higher layers by means of a PH-Activate Request primitive, or on the receipt of INFO 0 or lost framing while in the active state (G3). Then the choice to eventually deactivate is up to higher layers within the NT.

6.2.1.2.3 STATE G3 (active): This is the normal active state where the NT and TE are active with INFO 4 and INFO 3, respectively. A deactivation may be initiated by the NT system management by means of an MPH-Deactivate Request primitive, or the NT may be in the active state all the time, under non-fault conditions.

6.2.1.2.4 STATE G4 (pending deactivation): When the NT wishes to deactivate, it may wait for a timer to expire before returning to the deactivated state.

CCITT G.714 CODEC Testing Tutorial

The incorporation of ISDN voice capabilities into terminal and computer equipment presents many engineers with new terminology and testing ideas. A brief overview of CCITT Recommendation G.714 is helpful in gaining familiarity with the concepts of codec performance testing.

INTRODUCTION TO G.714

There does not yet exist a single comprehensive CCITT recommendation for testing codec and audio performance in the ISDN environment, and many systems vendors are using selected portions of CCITT G.714 as a basis for evaluating components. In fact, the transmission characteristics of the Am79C30A datasheet are based upon G.714. As will be seen in the following sections, only portions of G.714 are applicable to ISDN ICs, since G.714 was defined to specify performance of A/D and D/A conversions in a four-wire central office trunk environment. In such mixed analog and digital systems, a voice signal may pass through multiple A/D and D/A conversions; for instance, a signal may spend part of its time on analog frequency division multiplexed microwave channels and part of its time on digital T1 channels. Because of the wide variation in analog phone line characteristics and the cumulative effects of multiple A/D and D/A conversions, G.714 is a fairly stringent specification. Introduced in 1984 as an extension to the 1972 CCITT G.712 recommendation, G.714 specifies performance for send and receive sides (half channels) separately whereas G.712 specifies the performance of PCM connections from analog port to analog port (full channel). The intention is that any combination of transmitter and receiver that meet G.714 will be guaranteed to meet G.712 when cascaded, avoiding the possibility of offsetting errors in sender and receiver. At the current time, G.714 is applied to the ISDN environment due to the lack of a more relevant specification; it is expected that evolving CCITT recommendations will specify the requirements of the ISDN environment as a combination of codec performance and handset acoustics.

The introductory section of G.714 defines two terms that are used in the recommendation. A "standard send side" is a hypothetical ideal A/D converter preceded by an ideal low-pass filter. Similarly, a "standard receive side" is a hypothetical ideal D/A converter followed by an ideal low-pass filter. Typically, these devices are simulated by digital transmission test sets. Furthermore, where a nominal reference frequency of 1000 Hz is mentioned, the actual frequency chosen should be in the range of 1004 to 1020 Hz.

SECTION 2: ADJUSTMENT OF RELATIVE LEVELS

Section 2 of G.714 specifies the adjustment of absolute gain and load capacity. To measure the absolute gain of the transmitter, a 0-dBm0 analog signal at a nominal frequency of 1000 Hz is applied to the input, and the PCM output level must be 0 ± 0.3 dBm0. The 0-dBm0 level is approximately 3 dB below the peak PCM code and is intended to represent the normal peak signal level. The extra 3 dB is to allow some headroom to prevent clipping under normal operating conditions. It is important to note that "dBm0" is a relative measure; the exact rms voltage corresponding to the 0-dBm0 level varies from one system to another. In an analog transmission system environment, the 0-dBm0 point is chosen to represent a desired relation between the PCM codes on the codec digital interface and the corresponding analog levels on the system trunk interface. In an ISDN environment, the digital and analog conversions take place in a telephone environment and not a trunk environment. The absolute gain measurement is often applied as a component specification of part-to-part variation, but a true system specification for the ISDN environment would provide a specific relation between PCM codes on the codec digital interface and the corresponding acoustic levels on the telephone handset. It is crucial to carefully consider the scope and applicability of system specifications when applying them as component specifications.

The load capacity for the transmitter is checked by applying a sine wave of nominal frequency, 1000 Hz, to the input, and increasing the input level until the first occurrence of both the positive and negative full-scale PCM values. A value of 0.3 dB is added to this input level to compensate for the last PCM step, and the resultant value should be within ±0.3 dB of the theoretical full scale.

The receive side absolute gain is checked by applying a sequence of PCM codes corresponding to a 0-dBm0 sine wave, and verifying that the resultant analog output level is 0 ±0.3 dBm0.

Publication #	Rev.	Amendment	Issue Date:
12609	A	/0	7/89

SECTIONS 3, 4, 5, AND 6

Section 3 of G.714 describes level measurement stability requirements with respect to power supply and temperature variations, but does not specify the actual permitted variations. Careful selection of both active and discrete components should guarantee compliance with this system specification.

Sections 4 and 5 of G.714 specify the nominal four-wire input and output impedance, return loss, and longitudinal balance requirements of analog lines—parameters that are not directly relevant to the ISDN environment. Similarly, Section 6 deals with the signal levels that should be present at the interface between a four-wire trunk and frequency division multiplexed equipment, which is obviously not directly relevant to the ISDN application.

SECTIONS 7 AND 8: ATTENUATION AND GROUP DELAY

Section 7 of G.714 specifies a template for gain versus frequency performance, which is illustrated in Figure 2/G.714 at the end of this application note. It may be seen that this template is relatively flat between 300 and 2400 Hz, the usable audio range. Increased attenuation is allowed below 200 Hz to help filter 50- and 60-Hz power-supply harmonics, and the requirements are relaxed above 2400 Hz to allow a low-pass filter roll-off to be implemented in a practical fashion. The test is performed at a preferred power level of –10 dBm0, although 0 dBm0 is considered acceptable. The nominal 0-dB reference point is at 1000 Hz.

Section 8 of G.714 specifies a template for group delay distortion, and also specifies that the absolute group delay for transmitter and receiver should not exceed 360 and 240 microseconds, respectively, when measured at the frequency that has minimum group delay. The template allows for increased group delay at the upper and lower limits of the voiceband to allow out-of-band attenuation requirements to be met with simpler filters. As in the previous test, the preferred input power level is –10 dBm0, but 0 dBm0 is an acceptable alternative. The group delay distortion template is illustrated in Figure 3/G.714 on page 16.

SECTIONS 9–13: NOISE MEASUREMENT

Section 9 of G.714 specifies that with the input of the send side terminated with a nominal 600-ohm impedance, the measured idle channel noise should not exceed –66 dBm0p. Similarly, Section 10 states that the noise contributed by the receiver should not exceed –75 dBm0p when its input is driven by either PCM 0 (μ-law) or PCM 1 (A-law). A value of –75 dBm0p means that the noise should be attenuated 75 dB relative to the 0-dBm0 point, and adjusted by a psophometric noise weighting curve. The psophometric curve is widely used in Europe to combine the filtering effects of telephone handsets and subjective human hearing response. In North America, a different curve called "C-message weighting" is used for the same purpose.

Sections 11 and 12 of G.714 define limits for the rejection and prevention of out-of-band noise. Section 11 requires that any sine wave of frequency greater than 4.6 kHz at the input of the send side should not result in any in-band signal greater than 25 dB below the level of the test signal. Furthermore, "under the most adverse practical network conditions" of out-of-band noise, the PCM channel should not contribute more than 100 pWOp of additional noise in the band 0–4 kHz at the channel output. This requirement is intended as a guideline and is not applicable as a component specification since the test conditions are not explicit. The unit pWOp refers to picoWatts of power, referenced to the 0-dBm point and adjusted by a psophometric noise filter. For the receive side, Section 12 indicates that a 0-dBm0 digitally simulated sine wave in the range of 300 to 3400 Hz should not result in any spurious out-of-band image signals greater than –25 dBm0.

In addition to the total noise power specifications, Section 13 requires that no single frequency from the send or receive side should exceed –50 dBm0.

SECTIONS 14 AND 15: STD AND GAIN TRACKING

For signal-to-distortion testing, Section 14 of G.714 recommends two alternative methods that are not exactly equivalent: sine wave testing (Method 2) and noise-based testing (Method 1). It is stated that Method 1 "gives fairly smooth curves, not very dependent upon input signal level. The sine wave method can be more sensitive in identifying possible localized codec imperfections. Thus the two methods respond to practical codec impairments in slightly different ways." The G.714 specification allows regulatory administrations to choose between either or both of these methods. For sine wave testing, the test signal used is a nominal 420-, 820-, or 1020-Hz waveform. For noise-based testing, a particular pseudorandom sequence of tones is used. There are two different templates for send and receive in noise-based testing, whereas a single template applies for sine wave testing.

A similar situation exists for the measurement of gain variation with respect to input level, commonly referred to as gain tracking. A choice is given between noise-based testing or sine wave testing, with different templates for each method.

The various templates for STD and gain tracking are attached in G714 Figures 4a, 4b, 5, 6, and 7.

SECTIONS 16–18: CROSSTALK AND SIGNALING

Since G.714 was written to specify four-wire analog trunks, a variety of crosstalk tests are defined in Sections 16 and 17 to limit the crosstalk between send and receive channels, and between individual channels of a multiplex group. These measurements do not directly apply to the ISDN environment. Moreover, in a telephone application, the designer will deliberately mix a large sidetone signal into the receive path from the transmit path.

Section 18 of G.714 deals with interference from signaling, which is not directly relevant to the ISDN environment.

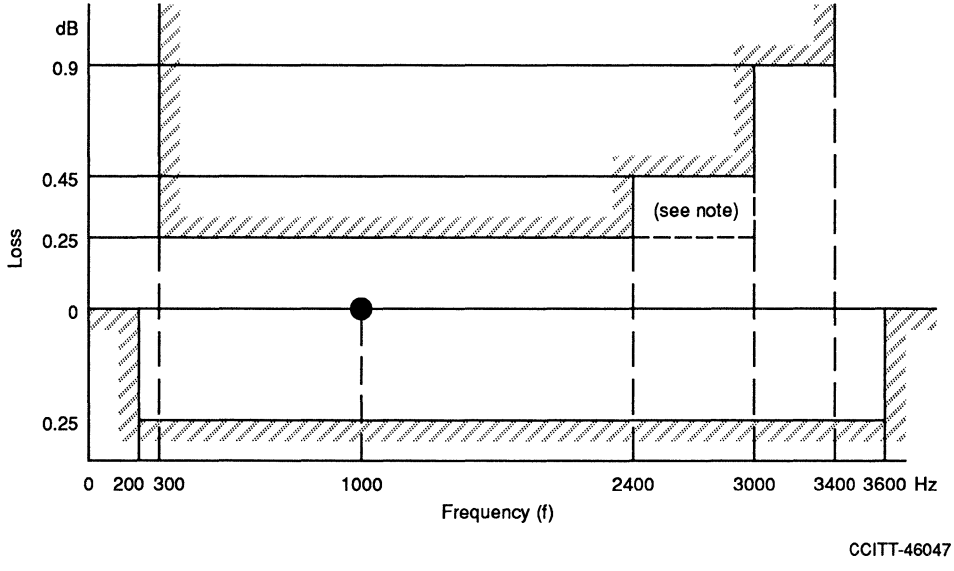


Figure 2/G.714. Attenuation/Frequency Distortion

Note: In some applications in which several PCM channels may be connected in tandem, it may be necessary to extend the +0.25-dB limit from 2400 Hz to 3000 Hz.

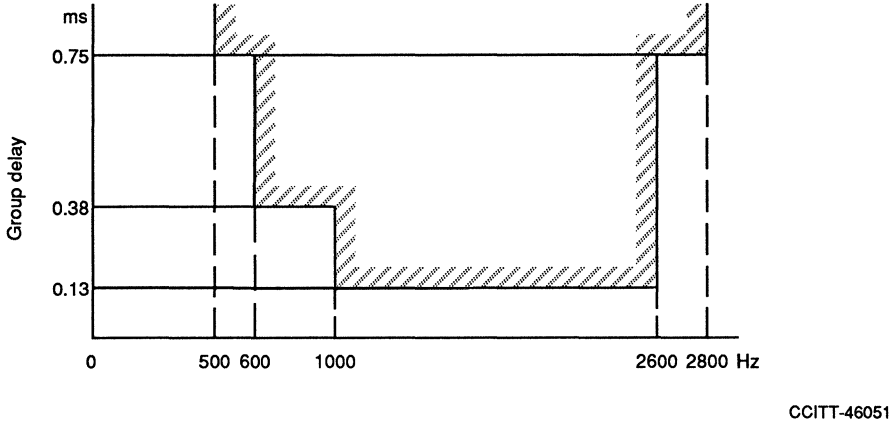


Figure 3/G.714. Group Delay Distortion with Frequency

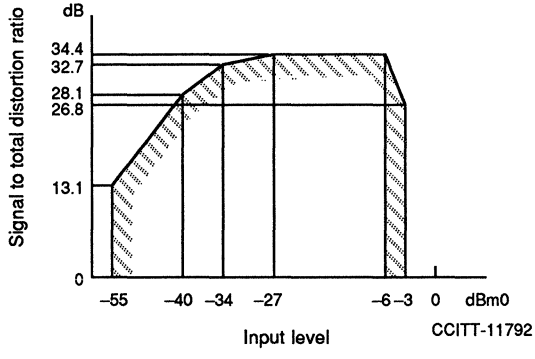


Figure 4a/G.714. Signal-to-Total Distortion Ratio as a function of Input Level (Method 1), Send Side

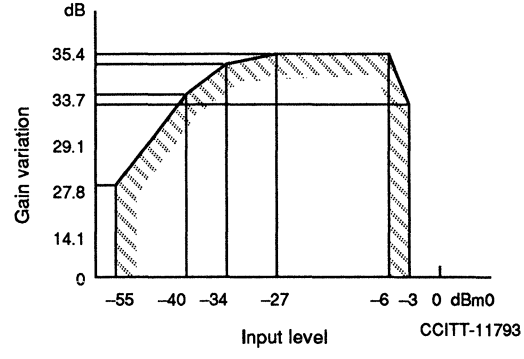


Figure 4b/G.714. Signal-to-Noise Distortion as a function of Input Level (Method 1), Receive Side

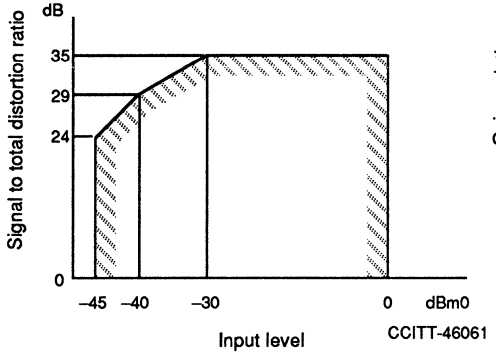


Figure 5/G.714. Signal-to-Total Distortion as a Function of the Input Level (Method 2), Send and Receive Side

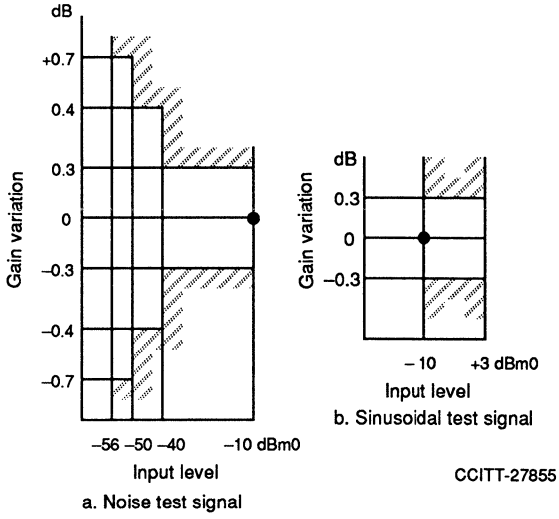


Figure 6/G.714. Variation of Gain with Input Level (Method 1)

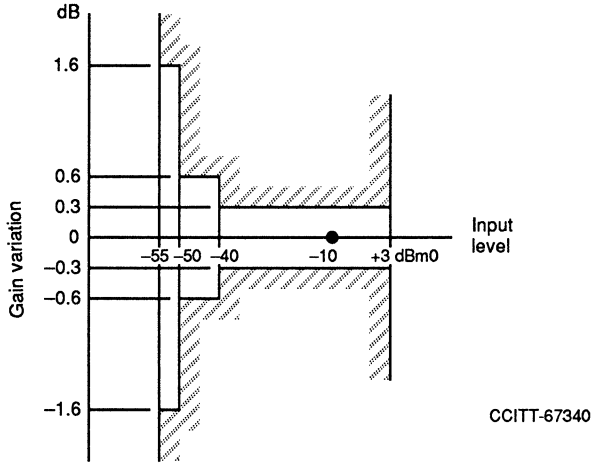


Figure 7/G.714. Variation of Gain with Input Level (Method 2)

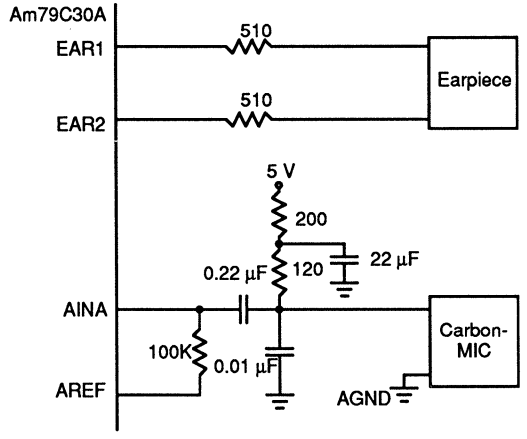
Am79C30A Carbon Handset Interface

The carbon-type microphone was for many years the standard choice for telephone design. Although it is being supplanted by electret and dynamic microphones, the carbon microphone still offers advantages of user acceptance, mechanical reliability, and ease of interface.

The accompanying schematic illustrates that a carbon handset may be interfaced to the Am79C30A DSC using only a few discrete components. The receive path from the DSC EAR1 and EAR2 outputs to the handset speaker is particularly simple, consisting of a pair of 510-ohm resistors. A pair of resistors was chosen instead of a single 1.1-kohm resistor to provide current limiting protection on both EAR outputs in case the user connects something other than a handset to the connector. The 510-ohm series resistors were selected to provide a subjective comfortable listening level with the DSC gains set at 0 dB. This design was implemented using an ITT 6544-0M2 handset, and it is important to note that the series resistor values may vary depending upon the handset manufacturer. Some further adjustment of the series resistance may be needed for compliance with applicable sound pressure level tests, depending upon the application. Note that the total resistance seen between the EAR1 and EAR2 outputs must be greater than 540 ohms, which is the maximum drive capability of the EAR outputs. The LS outputs are provided for lower impedance loads, such as loudspeakers, and are capable of driving a 40-ohm load. The GER filter of the DSC may be used for volume control, but it is desirable to center the desired volume level using the external resistance to make optimum use of the dynamic range of the digital signal processing.

In the transmit path, the gain of the carbon microphone is determined by the series combination of the 120-ohm and 200-ohm resistors, which set the bias current provided to the microphone. As was the case for the earpiece, the gain was set to provide a subjective comfortable volume level, and may require adjustment for a different manufacturer or if the application requires specific acoustic testing.

In all microphone circuits for the Am79C30A DSC, it is critical to ensure that the external circuitry does not contribute noise in the audio band. This is particularly true in personal computer applications, since the power supplies tend to be very noisy. The 200-ohm resistor and the 22-microfarad capacitor form a low-pass filter for the +5-V power supply, which reduces supply noise that would otherwise appear at the AINA input. If the power supply was quiet, the RC network would not be required and the 120-ohm and 200-ohm resistors could be replaced by a single 320-ohm resistor. The .01-microfarad capacitor is intended to filter out high-frequency noise from the microphone and is optional. The 0.22-microfarad capacitor provides DC blocking so that the 100-kohm resistor may bias the AINA input from the analog reference AREF.

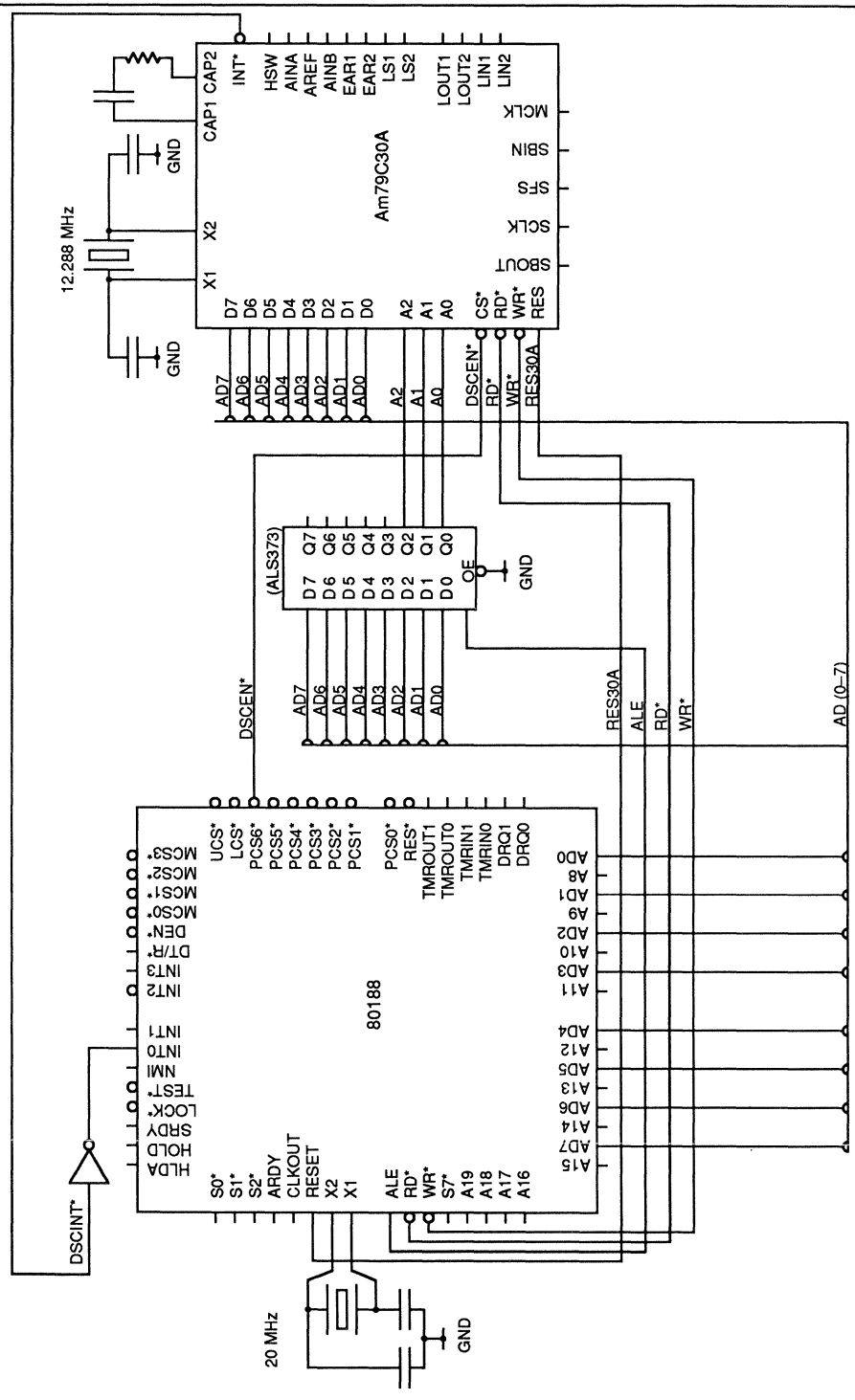


Publication #	Rev.	Amendment	Issue Date:
12610	A	/0	7/89

Am79C30A 80188 Microprocessor Interface

The 80188 is an excellent choice of microprocessor to use in conjunction with the Am79C30A DSC, since it integrates several useful peripherals and provides sufficient processing power to easily handle D-channel signaling and data processing. Spare processor bandwidth may be used to implement data processing capability for a B channel or to add other system features.

The accompanying schematic illustrates the key components that are required to interface the 80188 microprocessor to the Am79C30A DSC. The Am79C30A presents an iAPX-type interface, thus the DSC connects virtually directly to the microprocessor system. The 8-bit data bus is directly connected between the DSC and the microprocessor; the CMOS design of the DSC and the high drive capability of the 80188 microprocessor make extra bus buffering unnecessary for most systems. The RD* and WR* control signals are also directly connected between the microprocessor and the DSC. A '373-type transparent latch strobed by the processor's ALE signal is used to demultiplex the low-order address bits, and the 3 least significant output bits of this latch are connected to the DSC. Generally, 80188-based systems require this demultiplexing latch to interface to other peripherals and memory. The interrupt output of the DSC is active LOW, which necessitates an inverter before connection to an 80188 interrupt input. The DSC reset is provided by the reset output of the processor. The DSC chip select is provided by a peripheral chip select line from the 80188 microprocessor. The RD* and WR* minimum LOW time for the DSC is 200 ns, which requires the peripheral chip select line to be programmed for insertion of one processor wait state for 10-MHz operation. The DSC crystal in this application diagram is Saronix part number NYP-122-20, used in conjunction with two 30-pF load capacitors. If 10-MHz processor operation is not required, the DSC MCLK output may be used as the clock source for the processor to provide a wide range of frequencies under software control.


Am79C30A
CAP2
CAP1
INT*
HSW
AINA
AREF
AINB
EAR1
EAR2
LS1
LS2
LOUT1
LOUT2
LIN1
LIN2
MCLK
SBIN
SFS
SCLK
SBOUT
X2
X1
12.288 MHz
GND
D7
D6
D5
D4
D3
D2
D1
D0
A2
A1
A0
CS*
RD*
WR*
RES
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
DSCEN*
RES30A
(ALS373)
Q7
Q6
Q5
Q4
Q3
Q2
Q1
Q0
OE
ALE
AD (0–7)
80188
UCS*
LCS*
PCS6*
PCS5*
PCS4*
PCS3*
PCS2*
PCS1*
PCS0*
TMROUT1
TMROUT0
TMRIN1
TMRIN0
DRQ1
DRQ0
MCS3*
MCS2*
MCS1*
MCS0*
DEN*
DT/R*
INT3
INT2
INT1
INT0
NMI
TEST*
LOCK*
SRDY
HOLD
HLDA
S0*
S1*
S2*
ARDY
CLKOUT
RESET
S7*
A19
A18
A17
A16
DSCINT*
20 MHz

ISDN Reference Model Tutorial

The ISDN reference model provides a topographical map of the network from the user to the telephone company's switch. This map identifies equipment types and "reference points" throughout the network.

BACKGROUND

The basic structure of the Integrated Services Digital Network (ISDN) is specified in the CCITT I.411 recommendation. The network architecture is rooted in the existing analog telephone network, in that the digital portions of the present telephone network are based on a 64-kb/s channel, where voice is digitized as 8000 8-bit words per second. In general ISDN retains this basic 64K rate, but allows it to be used for either voice or data.

ISDN service is divided into two classes, basic rate and primary rate. The basic rate service provides two 64-kb/s channels for either voice or data—referred to as bearer, or B, channels, and a 16-kb/s signaling/data channel, labeled the D channel, for call control and low speed, up to 9600-b/s packet data. This "2B plus D" capability is the standard service provided to the user, that is, a telephone jack on the office wall provides the basic rate interface. Primary rate service provides a combination of 23 B channels and one 64-kb/s D channel (in Europe, primary rate service is 30 B channels plus one D channel because the European inter-office trunk network is based on a 2.048-megabit-per-second data rate instead of the 1.544-megabit-per-second rate used in North America). This "23B plus D" service is used primarily to connect PABXs to central offices and mainframe (or mini) computers. The general idea is that the B channels from all of the basic rate interfaces are gathered by the switch, PABX, or central office (Centrex service) and (1) routed within the switch to other local basic rate interfaces, (2) routed to distant basic rate interfaces via other switches and primary rate inter-switch trunk lines, or (3) concentrated and routed to a local or remote computing facility via primary rate lines. There are other options, such as providing primary rate service to the user's desk, but the options listed above are the predominant configurations.

The bandwidth of the primary rate interface can be partitioned in ways other than 23B plus D. For instance, one D channel can support its associated 23 B channels plus 24 B channels from each of an additional three primary rate interfaces. Alternately, the bandwidth of the primary rate channel can be partitioned into four 384-kb/s "H0" channels, or one 1.536-megabit-per-second "H11" channel (1.92-megabit-per-second "H12" in Europe). H0 channels can be used in combination with 64-kb/s B channels on the same primary rate line.

ISDN LANDSCAPE

Independent of whether basic or primary rate service is provided, the network topography from the desk top to the switch is the same. Figure 1 shows this topography, identifying specific classes of equipment that make up the network. In addition, certain "reference points" are defined that represent various interfaces. At each of these reference points the CCITT has established, or is in the process of establishing, standards for both hardware and software.

EQUIPMENT CLASSIFICATION

The equipment that makes up the network is classified based on function and location within the network. Starting from the network and moving toward the user we have the Line Termination (LT), which is located in the telephone company's switch, often at the central office. The LT performs Layer 1 functions for B and D channels, plus D-channel Layer 2 and 3 functions. Directly downstream of the LT is the Network Termination (NT). There are two types of NTs: NT1 and NT2. The NT1 performs such functions as line-length extension (repeaters) and two- to four-wire conversion (U to S interface). A key characteristic of NT1 devices is that they only deal with Layer 1 of the OSI seven-layer model. NT2s are intelligent, and actively participate in the call routing/control process. PABXs and line concentrators are examples of NT2 devices (a PABX actually contains both an NT1 and an NT2, with the T reference point being contained internal to the switch). Additionally, NT2 devices can be connected to multiple types of ISDN lines simultaneously. One important aspect of NT devices is that they often form the boundary between equipment owned by the customer and equipment owned by the telephone company. Farther downstream is the Terminal Equipment (TE). TEs represent computers, telephones, data terminals, and so on that are directly compatible with the ISDN. The last class of equipment is the Terminal Adapter (TA). TAs are the "box modems" of the ISDN

Publication #	Rev.	Amendment	Issue Date:
12612	A	/0	7/89

world. They provide for the connection of non-ISDN-compatible equipment to the network, that is, existing equipment.

REFERENCE POINTS

R Reference Point—The R reference point establishes the boundary between non-ISDN-compatible equipment and the network. Terminal Adapters (TA) are used to convert the communication protocol used by the non-ISDN-compatible terminal to the desired basic rate or primary rate protocol. It should be mentioned at this point that the network does not specify the data protocol used on either the B or H channels; all the network sees is a stream of bits. As a practical matter, however, standard protocols such as V.110, V.120, and X.25 will be used.

S Reference Point—The S reference point provides the connection between the NT2 equipment and the terminal (TE) or terminal adapter (TA). If no NT2 is present, there is no S reference point. In this case the TE or TA is connected directly to an NT1 device, and the interface is designated as a T reference point. Both primary rate and basic rate services can be provided at the S reference point.

It should be noted that it is common to refer to the four-wire basic rate service specified by CCITT recommendation I.430 as S interface service. While it is true that the S reference point is most often implemented in this way, a two-wire basic rate interface or primary rate interface can also be used at the S reference point. Additionally, the four-wire interface can be used to provide the U reference point. To keep the ISDN "alphabet soup" straight, it is important to remember that the various reference points identify the connection points between equipment classes and not the specific implementation or protocol of the interconnection.

T Reference Point—In addition to the connection of TAs and TEs to NT1 equipment, the T reference point provides the connection between an NT1 and an NT2. In a PABX, for example, the line circuit that connects to the network in the upstream direction (U reference point) provides the NT1 function, and the line circuit that connects to the TE or TA equipment (S reference point) provides the NT2 function. The T reference point in this case is internal to the PABX.

U Reference Point—The U reference point connects the LT to the NT1. Normally, a two-wire basic rate interface or a primary rate line is used, but the four-wire basic rate interface can also be used.

NETWORK TIMING

System Clock—The entire digital portion of the telephone network is synchronized to a master clock. This clock is passed from toll offices to central offices to PABXs down to the terminal equipment. This upstream to downstream flow of clocking information is required to maintain constant data rates throughout the network. It is the responsibility of any downstream device to recover the network clock from the data coming from the upstream switch. Data that is transmitted farther downstream is transmitted synchronously with respect to this recovered clock.

Slave/Slave Applications—A clock synchronization problem can arise on the trunk side of NT2 devices (side connected to the upstream device). The Layer 1 transceiver on the trunk-side line card recovers the network clock from the data on the line from the central office (this is how the PABX synchronizes to the network clock). The line card is said to be a "slave" to the network. The line card is also connected to the PABX's internal data highway (PCM Highway), which provides its own clock. Thus, the line card is also a "slave" to the PCM Highway. A synchronization problem exists since these two clocks are not necessarily locked in phase and frequency. To resolve this conflict, data buffering must be provided on the line card to absorb the mismatch.

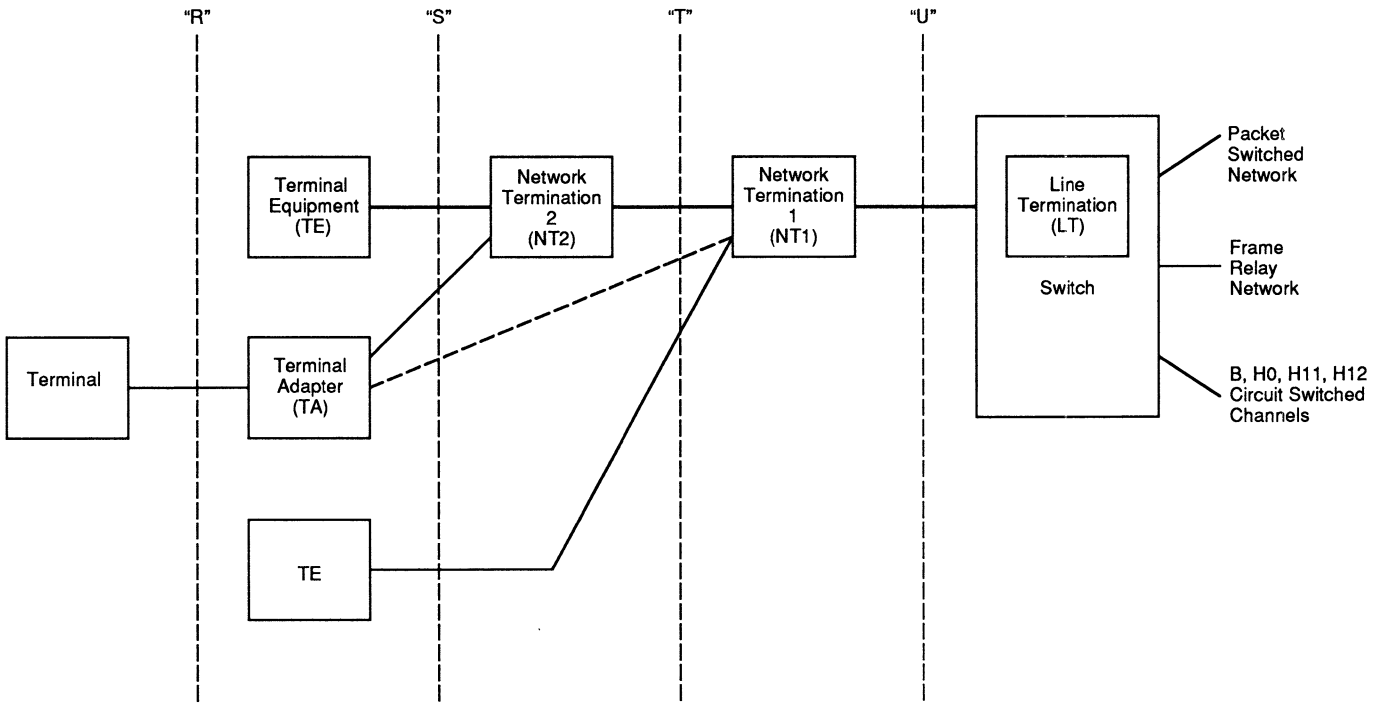


Figure 1. ISDN Network Topography

Am79C30A/32A DSC FIFO Handling

The Am79C30A Digital Subscriber Controller contains 2 data FIFOs and 3 status FIFOs to improve the throughput of packet transmission and reception. While the basic operation of the FIFOs is straightforward, there are some "tunable" parameters that can be programmed to increase performance.

GENERAL FIFO OPERATION

The Am79C30A Digital Subscriber Controller (DSC) has two 8-byte data FIFOs: one for the transmission of data and one for the reception of data. In addition to the data FIFOs there are three 2-byte-deep status FIFOs; the D-Channel Receive Byte Count Register (DRCR), the Address Status Register (ASR), and the D-Channel Error Register (DER). These 2-byte status FIFOs make it possible to receive two back-to-back data packets. The data FIFOs are very useful in the transmission and reception of LAPD packet data via the ISDN S interface. In this scenario the minimum and maximum packet sizes are defined, but the actual length of the packets may vary between these two extremes. During data transmission, the packet data is initially stored in local RAM and the processor is used to move the data from memory into the DSC transmit FIFO. When packet reception is in progress, the processor is used to move data from the DSC receive FIFO into local memory. Except for the initial 8 bytes loaded into the transmit FIFO, all FIFO servicing is handled from an interrupt service routine. Due to the importance of receiving and transmitting data, FIFO handling is given top priority in the interrupt service routine.

To be able to transmit and receive data packets longer than 8 bytes, some mechanism must be available to efficiently fill and empty the FIFOs. The solution provided by the Am79C30A DSC is programmable thresholds. When the number of bytes in the transmit FIFO is reduced to a preprogrammed level as a result of data transmission, an interrupt is generated to alert the processor to the fact that the FIFO needs to be filled. Similarly, when the number of bytes in the receive FIFO grows to a preprogrammed level as a result of data reception, the processor is alerted to the fact that the FIFO needs to be emptied.

The values that may be programmed into the transmit FIFO threshold are 1, 2, 4, or 8 empty bytes. The values that may be used for the receive FIFO threshold are 1, 2, 4, or 8 bytes. A good way to handle data transmission through the FIFO is as follows. At the start of a packet transmission, the transmit FIFO is initially filled with 8 bytes of data, and transmission is started by writing the Transmit Byte Count Register. One point to be emphasized here is that if the Transmit FIFO is initially loaded with a number of bytes less than or equal to the Transmit Threshold, a Transmit Threshold interrupt will be generated when the Transmit Byte Count is written. In all other cases, the threshold interrupts occur only when the number of bytes contained in the FIFO changes to the level programmed in the threshold register. In the case where the transmit FIFO threshold is programmed to 4 empty bytes, a Transmit Threshold interrupt is generated after 4 bytes have been transmitted, indicating that there are now 4 empty bytes in the transmit FIFO. The DSC transmitter does not stop transmission, so the processor must begin filling the FIFO within four byte-transmission times (approximately 2 milliseconds) or underrun will occur. When the processor is refilling the FIFO, it performs write operations until bit 4 of the D-Channel Status Register 2 (Transmit Buffer Available) indicates that the FIFO is completely full. At this point, the interrupt service routine is exited, and the processor is free to perform other functions. See the sample source code for the Transmit FIFO Handler later in this technical note.

In the situation where the receive FIFO threshold is 4 bytes, an interrupt will be generated to the processor after 4 bytes have been received. To prevent overrun, the processor has approximately four byte-reception times (2 milliseconds) to begin reading information out of the FIFO. The processor should continually read the FIFO until bit 1 of the D-Channel Status Register 2 indicates that the FIFO is empty, or bit 0 of the DSR2 indicates that the last byte of the received packet has been read. The processor is then free to perform other functions.

The transmit and receive threshold values are "tunable" variables and depend upon the specific application. For example, experience has shown that threshold values of 4 are the most conservative choices. These values allow the processor a great deal of time to respond to the interrupt while also allowing the processor ample time to perform other tasks between interrupts. A transmit threshold of 4 empty bytes and a receive threshold of 4 bytes is a good starting point for handling the data FIFOs. These values may then be tuned as system performance dictates. When transmitting packets significantly larger than 8 bytes, however, the best system

Publication #	Rev.	Amendment	Issue Date:
12613	A	/0	7/89

performance may be achieved by programming the transmit FIFO threshold with 8 empty bytes and the receive FIFO threshold with 8 bytes. This situation results in a minimum amount of total time spent in the interrupt service routine while maximizing the amount of work done while actually in the interrupt service routine. The trade-off is that the system must be capable of responding to the threshold interrupt quickly enough (375 microseconds for transmit and 425 microseconds for receive) to prevent FIFO overrun and underrun.

DLC INTERRUPTS

Numerous interrupts are associated with the operation of the Data Link Controller (DLC), but not all of them need be enabled in every application. It is normally sufficient to enable interrupts for End of Valid Transmit Packet, End of Receive Packet, Transmit Threshold, Receive Threshold, and any D-Channel Error conditions that are pertinent to the application. This set of interrupts will allow the DSC to monitor all the events that arise during the course of packet data transmission and reception.

End of Receive Packet Interrupt

When an End of Receive Packet (EORP) interrupt occurs, it will appear in the Interrupt Register as a D-Channel Status Register 1 (DSR1) interrupt. An EORP interrupt signifies that a closing flag has been received. Note, however, that the receive FIFO may still contain data from this received packet, and the number of bytes in the FIFO may not be enough to trigger the threshold interrupt. The D-Channel Status Register 2 (DSR2) bit 1 will indicate whether data to be read is still in the Receive FIFO. Any data remaining in the Receive FIFO must be read out 1 byte at a time until DSR2 bit 0 is set, indicating that the last byte of the receive packet has been read. Note that the Last Byte of Receive Packet status goes active *after* the byte has been read.

Reading the last byte of the receive packet causes the D-Channel Error Register (DER) to propagate to the top of the 2-byte DER FIFO, which in turn updates the D-Channel Status Registers and the Interrupt Register. It is important to read DER, DSR1, DSR2, and the Address Status Register (no required order) before reading the D-Channel Receive Byte Count Register (DRCR). Reading the MSB of the DRCR causes the next receive byte count and associated Address Status Register byte to propagate to the output of their respective 2-byte FIFOs, causing the D-Channel Status Registers and Interrupt Register to be updated. At this point, the D-Channel Error Register should be checked to determine if any receive error conditions, such as Overflow or Overrun, exist.

End of Transmit Packet Interrupt

When a packet is to be transmitted, the Transmit FIFO is loaded with a number of bytes, normally 8 for packets larger than 8 bytes. The D-Channel Transmit Byte Count Register (DTCR) is then written, because writing the MSB of the DTCR causes transmission to begin. If the FIFO is not preloaded, the software has to respond very quickly when the DTCR is written to prevent underrun. When the MSB of the DTCR is written, the transmitter will first ensure that it has access to the D channel, and then it will transmit an opening flag. If no data is in the FIFO after the opening flag has been transmitted, an underrun will occur and be followed by a transmit abort. The FIFO then remains reset until the error condition is handled.

The packet transmission will be handled by the interrupt service routine via the Transmit Threshold interrupt until the last byte of the packet has been transmitted. When the closing flag is transmitted, an End of Valid Transmit Packet interrupt will occur to alert the application software that the transmitter is now available for new transmissions. Care must be taken to keep track of how many bytes have been written to the Transmit FIFO, because the D-Channel Transmit Byte Count Register contains the number of bytes that are actually left to be transmitted, including the bytes in the FIFO. This means that software should keep track of how many bytes have been written to the FIFO, and it should stop transferring to the FIFO when the total number of bytes have been written. See the Interrupt Service Routine Transmit FIFO Handler sample code for an example of how this might be handled. After writing this last byte, the processor will exit the interrupt service routine and perform other tasks until it receives the End of Valid Transmit Packet Interrupt. No FIFO cleanup is required when the End of Valid Transmit Packet occurs. The application usually will use this interrupt to inform higher level software that the transmitter is now available to send another packet.

MISCELLANEOUS

Simultaneous Receive Threshold and End of Receive Packet Interrupts

When receiving a packet, it is possible for a Receive Threshold interrupt and an End of Receive Packet interrupt to occur at the same time. For example, if the Receive Threshold was programmed for 4 bytes and the last 4 bytes of a packet are received, the Receive Threshold and the End of Receive Packet interrupts will occur simultaneously. The interrupt service routine will have to handle both of these interrupts. Assuming the Receive Threshold is at a higher priority in the interrupt

service routine than the End of Receive Packet, the Receive Threshold will get serviced first. The receive threshold service routine will be responsible for reading the remaining bytes out of the receive FIFO. After the last byte of the receive packet is read, a flag should be set by the application software indicating all bytes of the current packet have been read. (See the *rcvstatus* software register in the Receive FIFO Handler source code example.) When the End of Receive Packet interrupt is serviced, the rcvstatus software register will be checked to determine if any bytes from the current packet are still in the FIFO. When this flag bit is set in the rcvstatus register, the interrupt service routine knows not to read any more bytes. The Receive Byte Available bit in DSR2 cannot be used for this purpose, because there may be more bytes in the FIFO that belong to the next packet.

Terminated Data Packets

Packet transmission and reception may be terminated in many ways, and this situation must be handled differently depending upon whether transmission or reception was terminated. For example, a packet being transmitted may be intentionally aborted by the application software, or the abort may be the result of an error condition, such as underrun, which terminates the packet transmission. To intentionally abort the transmission, a binary 1 must be written into bit 7 of the Initialization Register. Be sure to write a binary 0 into bit 7 any time after writing the 1, or the Am79C30A will continually issue abort characters and will be unable to begin transmission of the next packet. The transmit data FIFO and all of its internal pointers are reset when a transmit abort occurs. The transmit FIFO will remain in this reset condition until DSR1 and DER are read or the D-Channel Transmit Byte Count Register is written to begin a new transmission.

The case of the receive abort is a bit more complicated because the receive FIFO is capable of holding more than one packet at a time. A perfectly good packet may be in the receive FIFO and then an error condition can occur during reception of the next packet. The erroneous packet reception will be terminated, but the initial packet should remain intact. The current packet reception can be intentionally aborted by writing a binary 1 to bit 6 of the Initialization Register. After 60 microseconds to allow for the abort, a binary 0 must be written into bit 6 or the Am79C30A will continue to abort incoming data. Any time a receive error or intentional abort occurs, the bytes that are already in the receive FIFO will remain there. The only way to remove this data is to read the bytes out one at a time or place the Am79C30A in idle mode. Placing the part in idle mode is normally much too drastic a measure to take, because it essentially halts all functions of the part except LIU activation detection. This means that any voice connection, data transmission, and data reception will be terminated. If it is possible that the bytes in the FIFO are valid data, then the FIFO must be read as normal and the data transferred to local memory for storage. If it is known that *all* the data in the FIFO are garbage, then the data may be read out of the FIFO and discarded until the FIFO is empty. To determine if the FIFO is empty, bit 1 of the D-Channel Status Register 2 (Receive Byte Available) must be monitored until it indicates that no more bytes are available. After reading out the data in the FIFO, the application may have a need for the contents of the status and error registers. Whether or not the status and error registers are read, the DRCR must be read to bring the receive FIFO out of the aborted state. This is normally done anyway when an error condition occurs, so it will not impact performance. The receiver will then go back to searching for an opening flag for the next incoming data packet.

Last Byte Transmitted Interrupt

The Last Byte Transmitted (LBT) status indication is found in the D-Channel Status Register 2 (DSR2) bit 3, and it will generate an interrupt if the D-Channel Mode Register 3 (DMR3) bit 4 is set. The LBT status is generated when the first bit of the last data byte is transmitted out of the transmit FIFO, and its most useful function is to provide the earliest possible indication that the FIFO is available to be loaded with another packet of data. Consider the case of back-to-back packets. As soon as the first bit of the last byte of data is transmitted to the S interface, the LBT status/interrupt will be generated. The remainder of the data byte, the 2-byte FCS, and the closing flag must still be transmitted. In the time it takes for this transmission to take place, the Transmit Byte Count Register can be loaded with the value for the next packet, and the Transmit FIFO can begin being filled with new data. The result of this is that the closing flag for the first packet will be shared as the opening flag of the second packet. This would be the way to maximize the use of the available bandwidth. One very important item to note is that in this mode of operation, an End of Transmit Packet indication will not be generated until the last back-to-back packet transmission is completed.

```
/* If XMT Threshold caused interrupt */
if (intstatreg & TTHRSH) {
  /* While transmit buffer available and all bytes not yet transmitted
  while ((rd_dsc(Dsc->dsr2) & TBA) && RAM->xmtcnt) {
    /* Write a byte from RAM to XMT FIFO */
    wr_dsc(DSC->d_buff, *Ram->xmtnext++);
    /* Decrement transmit byte count */
    Ram->xmtcnt--;
  }
}
```

INTERRUPT SERVICE ROUTINE
TRANSMIT FIFO HANDLER

```
/* If RCV Threshold caused interrupt */
if (intstatreg & RTHRSH) {
  /* Read the d-channel status register */
  dsr2 = rd_dsc(Dsc->dsr2);
  /* While RCV Byte Available in FIFO */
  while (dsr2 & RBA) {
    /* Read a byte from RCV FIFO and place it in local RAM */
    *RAM->rcvfree++ = rd_dsc(Dsc->d_buff);
      /* If last byte of RCV packet then */
      if((dsr2 = rd_dsc(Dsc->dsr2)) & LBRP
      {
        /* Set status flag and break */
        rcvstatus | = RCV_FLAG;
        break;
      }
  }
}
```

INTERRUPT SERVICE ROUTINE
RECEIVE FIFO HANDLER

Link Layer Tutorial

The functions performed by the OSI Data Link Layer (DLL) are described. An overview of the DLL's role in the communication process is presented, as well as a description of the elements that implement the DLL functions. An explanation of the fundamental differences between X.25's LAPB and ISDN's LAPD is included, followed by a brief description of Frame Relay.

DATA LINK LAYER OVERVIEW

The Data Link Layer (DLL) is the second layer in the OSI reference model. It provides services to the Network Layer (Layer 3) and uses the services provided by the Physical Layer (Layer 1). The functions performed by the DLL are independent of the network media (i.e., telephone wires, satellite, coaxial cables, fiber-optic lines, etc.) and deal only with the establishment and maintenance of the communication link connecting source and destination endpoints.

In this application note, the terms "communication link" and "data link" will be used in describing the functions of the DLL. The distinction between the two can best be understood by using the telephone system as an example. In this case, a circuit-switched point-to-point "communication link" is established when the remote party answers the telephone. The "data link" is established when both parties begin exchanging information. Each party can be viewed as a DLL entity implementing a simple protocol for information exchange. Information between the two parties, which may be conveyed in error or is simply unintelligible due to a poor connection, is corrected when one end of the "data link" asks the other end to repeat the information. If the connection is very poor, the "communication link" is terminated by hanging up. Thus, simply stated, the function of the DLL is to ensure the error-free transfer of information across a communication link.

In the telephone example cited, the exchange of information is always point-to-point, even though the routing of the connection may encompass more than one Central Office Exchange or "Hub." In a Packet-Switched Network, however, all Hubs play an equally important role in ensuring the integrity of the data being sent/received. As shown in Figure 1, data sent between source and destination nodes A and B may encompass more than one Hub. The communication links between Hubs are independent of one another, and each Hub incorporates a DLL entity in the form of a protocol that establishes the rules for proper data exchange.

The information that determines which path is taken through the network, and hence the total number of communication links and data link entities involved, is embedded within defined bit fields of each data packet being transferred. It is the responsibility of the DLL to ensure that this "Network Layer" information and user data are transferred without error, and in proper sequence, from its source Hub (point A) to its destination Hub (point B). Each Hub along the path stores the packets and verifies their integrity before forwarding them to the next Hub. From this point of view, the DLL and the Network Layer can be thought of as always operating on a point-to-point basis, but with the Network Layer having the additional capability of routing data from an incoming point-to-point data link to an outgoing point-to-point data link.

In the case where the destination node B resides on a multi-drop or multipoint link (Figure 2), Data Link 3 can be thought of as implementing individual point-to-point data links that are multiplexed over the same physical channel. In cases such as these where the multipoint network is not geographically dispersed, the Network Layer is defined as being null (not implemented) since routing occurs at a higher layer.

Examples of DLL protocols are Bisynchronous (BISYNC) and Synchronous Data Link Control (SDLC) of IBM's SNA, Digital Data Communications Message Protocol (DDCMP) of DEC's DECNET, Ethernet's Local Area Network (LAN) IEEE 802 standard, and ISO's High-level Data Link Control (HDLC), a subset of which is known as LAPB (Link Access Procedure Balance), and is the accepted DLL protocol of CCITT's X.25 packet switch specification. Both BISYNC and DDCMP are referred to as Character-Oriented Protocols (COP), and HDLC and SDLC are referred to as Bit-Oriented Protocols (BOP). The fundamental difference between the two is in the way Layer 3 data is enveloped before it is released to the physical layer. The material presented in this appplication note deals solely with the specifics of HDLC/SDLC.

Publication #	Rev.	Amendment	Issue Date:
12614	A	/0	7/89

The following sections define the structures used and functions performed by the DLL in more detail.

THE DATA LINK LAYER

Much like the partitioning found in the IEEE 802 LAN standard (IEEE 802 standard partitions its DLL into two sublayers: the Logical Link Control [LLC] and the Media Access Control [MAC] sublayers), the OSI's DLL can also be viewed as consisting of two sublayers: the "frame" sublayer and the "sequence" sublayer. The frame sublayer deals with frame delineation, data transparency, and physical layer error detection; whereas the sequence sublayer deals with commands and responses (the actual protocol) that are used in ensuring that the information arrives error free and in proper sequence over the communication link. The frame sublayer (sometimes referred to as Layer 2–) is specified in ISO-3309 High-level Data Link-Frame Structure, and the sequence layer (sometimes referred to as Layer 2+) is specified in ISO-4335 High-level Data Link-Elements of Procedures.

THE SEQUENCE SUBLAYER

Two modes in the ISO-4335 document determine how the sequence layer is applied over the communication link. These two modes are: the Normal Response Mode (NRM) which is a procedure based on a master-slave principle using two types of stations, primary and secondary, with different functional capabilities, and the Asynchronous Balance Mode (ABM) which is a procedure based on combined stations (stations with both master and slave capability).

In NRM operation the primary station is responsible for data link connection, disconnection, and error recovery. Secondary stations only respond to commands from the primary and, in general, are not allowed to transmit unless given permission to do so from the primary station. This mode of operation is commonly used in half-duplex and multipoint data links, and is the mode of operation most commonly found in IBM SNA environments. IBM's SDLC implements a subset of the HDLC Elements of Procedures and can be found in IBM document GA27-3093-3.

In ABM operation each station is responsible for data link connection, disconnection, and error recovery, and either station can transmit without permission from the other station. ABM stations always operate on a point-to-point basis and never multipoint. This HDLC mode of operation is the "sequence" sublayer specified in CCITT's X.25 DLL specification, and is referred to as Link Access Procedure Balanced (LAPB).

Embedded within the control field of every Layer 2 frame is a Poll (P)/Final (F) bit which is used by the sequence sublayer in maintaining the data link. This bit is used to solicit responses from another station. Its use, however, differs when operating in NRM or ABM (LAPB). In NRM a command frame from the primary station with the P bit set causes the secondary station to initiate transmission. A response frame from the secondary station with the F bit set indicates the final frame of transmission. Data transmission in NRM is always half duplex. In LAPB, however, a command frame from one combined station with the P bit set solicits a response frame from the remote station with the F bit set, but data transmission occurs at full duplex. In both NRM and ABM, timers are started when a frame with the P bit set is sent, and error recovery procedures are instigated if a timeout occurs before a frame with the F bit set is received.

THE FRAME SUBLAYER

The frame sublayer deals with the frame structure used to transport Layer 3 data over the physical layer. It deals with frame delineation, the method used in achieving data transparency, and the algorithm (CRC) used for error detection over the physical layer.

Frame delineation is performed in both SDLC and HDLC by the use of an 8-bit flag character (01111110) that indicates the start and the end of a Layer 2 frame and determines the location of the address field, control field, and the first bit of the information field. The closing flag determines the location of the 16-bit Frame Check Sequence (FCS). In order to preclude the occurrence of flag characters between the opening and closing flags of a frame, all information between them is subjected to a 0-bit insertion/deletion algorithm where the transmitter inserts a 0 bit immediately following the occurrence of 5 consecutive 1 bits. The receiver then does the opposite by deleting any 0 bit that occurs after 5 consecutive 1 bits.

The address field is used for identifying source and destination endpoints on the data link, and for the proper interpretation of command or response frames. In LAPB, for example, the address field of a command frame transmitted to a remote station contains the address of the remote station, whereas the address field of a response frame transmitted from the remote station contains the address of that station.

The control field conveys the functional operation of the data link in the form of commands and responses between Hubs. Examples of commands and responses (covered in ISO-4335) are Receiver Ready (RR), Receiver Not Ready (RNR), Set Asynchronous Balance Mode (SABM), and Disconnect (DISC) for commands and RR, RNR, and Disconnected Mode (DM) for responses. Thus, it is through the control field that the sequence sublayer protocol is implemented.

The Frame Check Sequence (FCS) is a 16-bit CRC character inserted between the end of the information field and the closing flag of a frame, used to ensure data integrity over the physical layer. This 16-bit character is

calculated based on the data sent using a specific algorithm. The receiver uses the same algorithm to calculate the CRC character on the received data and compares the result with the 16-bit CRC character received. A mismatch results in an error and the frame is discarded without alerting the next higher layer. The discarding of the frame eventually results in its retransmission by the sequence sublayer. The algorithm used to generate the 16-bit CRC character is based on the CCITT standard polynomial $X^{16} + X^{12} + X^5 + 1$.

HDLC FRAMES

Three types of HDLC frame formats (Figures 3a and 3b) are used for transporting data over the physical layer: (1) Information frames (I frames), (2) Supervisory frames (S frames), and (3) Unnumbered or Unacknowledged frames (U frames). In addition, an Extended or Non-Extended frame format can be specified.

I frames are used to transport data and are sequentially numbered with send (Ns) and receive (Nr) sequence numbers. The address, control, and FCS fields are the only fields acted on by the DLL entity, while an acknowledged information field is always passed on to the next higher layer (Layer 3). In HDLC, the length of the information field can be any number of bits, whereas IBM's SDLC restricts the length to an integer number of bytes. The maximum information field transmitted in either case, however, is application dependent.

S frames are used to acknowledge correctly received I frames, to request retransmission in case of errors, and to implement flow control (that is, request a temporary suspension of the transmission of I frames).

U frames are used for data link initiation or disconnection, reporting procedural errors, and for transferring data which is not sequenced.

The extended or non-extended frame format refers to the modulus of the sequence numbers, N(s) and N(r), used. Sequence numbers of I frames cycle through a set of numbers 0, 1, 2, ..., M-1, where M is the modulus number. The extended control field uses a modulus number of 128 and the non-extended control field format uses a modulus number of 8. The extended mode is used in cases where long propagation delays are involved, because a station must stop transmitting I frames if it has M-1 unacknowledged I frames outstanding. An implication of the extended mode is that more memory is required because each Hub stores all transmitted I frames, and they are not passed on to the next layer until they have been successfully acknowledged by the corresponding peer sequence sublayer. The selection of Extended/Non-Extended frame format in LAPB, for example, is done via the Set Asynchronous Balance Mode Extended (SABME) or the Set Asynchronous Balance Mode (SABM) U-frame command.

ISDN

The intent of the Integrated Services Digital Network (ISDN) is to integrate a set of services (voice and data) over standard telephone cabling. In order to accomplish this, two types of channels have been defined within the scope of the ISDN; a B channel and a D channel. A B channel refers to one of two full-duplex 64-kb/s channels used to transport digitized voice or data through the network. A D channel refers to a single full-duplex 16-kb/s channel that is shared among two or more users over the physical layer interface, and is used for transporting B-channel signaling and low-speed data. Because this scheme involves communication over a point-to-multipoint link, a new DLL protocol referred to as Link Access Procedure D channel (LAPD) was developed.

Link Access Procedure D (LAPD) is covered in CCITT's I.441 (Q.921). LAPD Layer 2 frames, used to transport data over the D channel, are based on the HDLC framing, with Layer 3 based on CCITT's Q.931. Hence, much of what has already been presented with regard to the DLL (frame sublayer and sequence sublayer) applies directly to LAPD.

The LAPD frame format is shown in Figures 4a and 4b. A major difference in an LAPD frame is that the address field is defined to be 2 bytes instead of 1 as in LAPB, and is more fully utilized. Since LAPB is always point-to-point, its address field is used only to distinguish between command and response frames, whereas in LAPD, the address field is used to distinguish between other Layer 2 entities. The LAPD Command/Response (C/R) bit, however, is used in a similar way that the LAPB address field is used in differentiating command from response frames.

The LAPD address field consists of two subfields: a DLL Service Access Point Identifier (SAPI) used to indicate which protocol entity is to receive the Layer 2 frame, and a Terminal Endpoint Identifier (TEI) used for distinguishing between the various terminals connected on a data link. For example, according to CCITT I.441, a SAPI field received with a value of 0 identifies the *Layer 2 frame* as having call control information for the specified TEI, and a SAPI field received with a value of 16 identifies the Layer 2 frame as having packet information for the specified TEI.

Thus, the most significant functional difference between LAPB and LAPD is that LAPD provides logical channel multiplexing at the data link layer (LAPD supports point-to-multipoint physical layer connections with "logical" channel multiplexing at the data link layer), whereas LAPB provides this same capability at the Network Layer (LAPB supports only point-to-point at the physical and data link layers with logical channel multiplexing oc-

curring at the Network Layer via Logical Channel Number [LCN] assignment).

FRAME RELAY

X.25 data link layer protocol software design is typically based on the use of state event tables that help implement a finite state machine. Using this technique, the software is designed such that it is always in a specific state at any instant in time. These table-driven state machines define all the states that the protocol can be in at any instant in time, plus all the events that can occur while in a specific state, what actions need to occur because of events, and what new state will be entered into as a result of the event occurring. In short, the current state and event information are used to index into a two-dimensional structure to determine what the next state will be. This translates to the use of a rather large case statement since the software must search the table each time an event occurs. With LAPD's multiplexing capability at the data link layer, intermediate Hubs can perform their internetwork switching tasks without having to process the complete DLL protocol, thus minimizing the massive state table searches.

This has prompted the CCITT in adopting a new mode of packet switch service referred to as Frame Relay. What Frame Relay does is combine the LAPD, SAPI, and TEI fields into a single field that is used in much the same way that LCNs are used in X.25's network layer. Each intermediate Frame Relay Hub only examines the first two fields of an LAPD frame when routing a frame, and only detects misaddressed or errored frames (frame CRC bad). Thus, the need for massive state tables is minimized and improvements in throughput are possible since much of the Layer 2 and 3's protocol overhead has been removed and made the responsibility of the communicating endpoints.

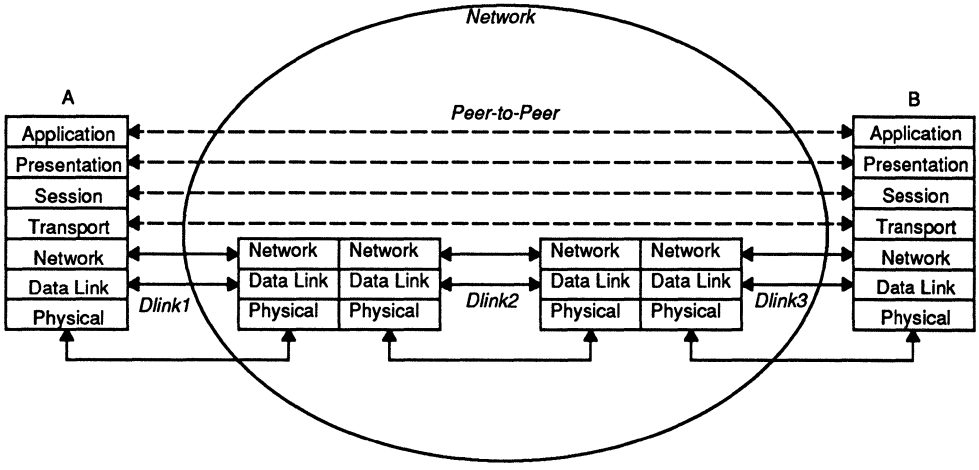


Figure 1

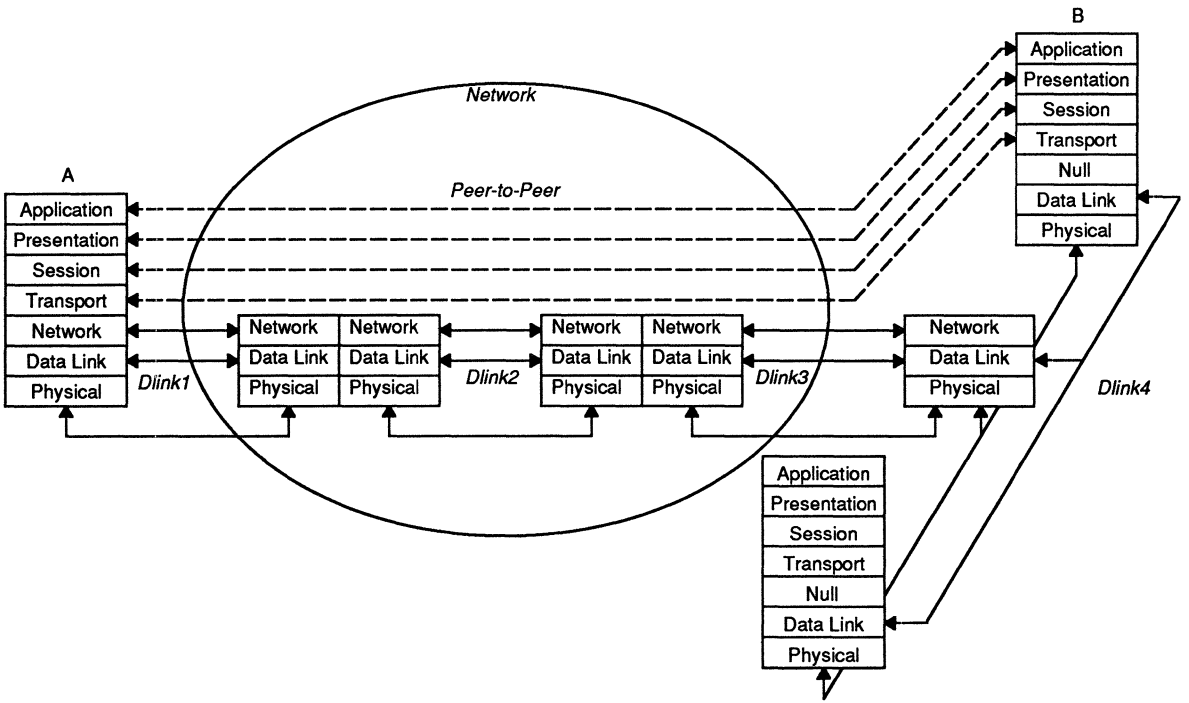

A
Application
Presentation
Session
Transport
Network
Data Link
Physical
Dlink1
Network
Network
Data Link
Data Link
Physical
Physical
Dlink2
Network
Network
Data Link
Data Link
Physical
Physical
Dlink3
Network
Data Link
Physical
Dlink4
B
Application
Presentation
Session
Transport
Null
Data Link
Physical
Network
Peer-to-Peer
Application
Presentation
Session
Transport
Null
Data Link
Physical

Figure 2

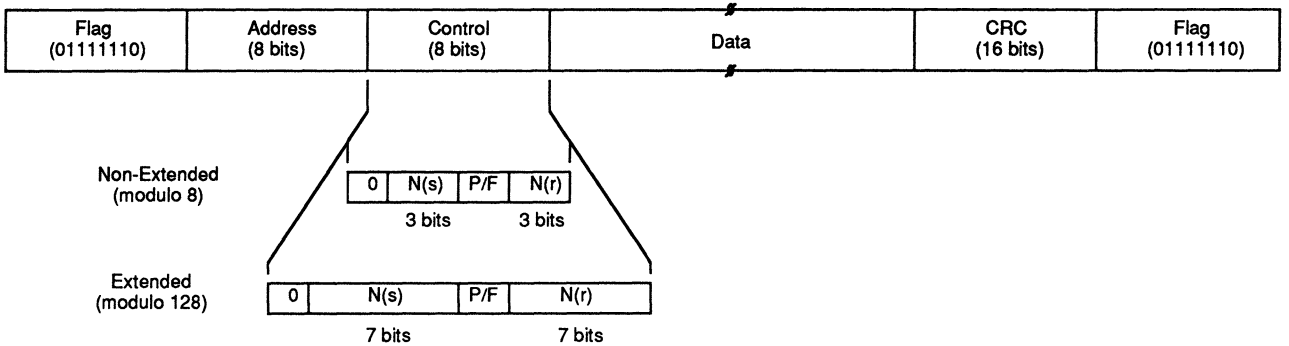


Figure 3a. I Frame

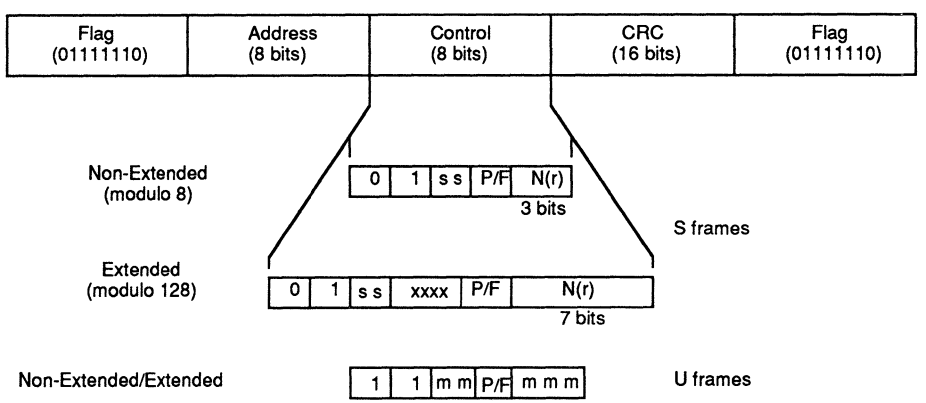


Figure 3b. U and S Frames

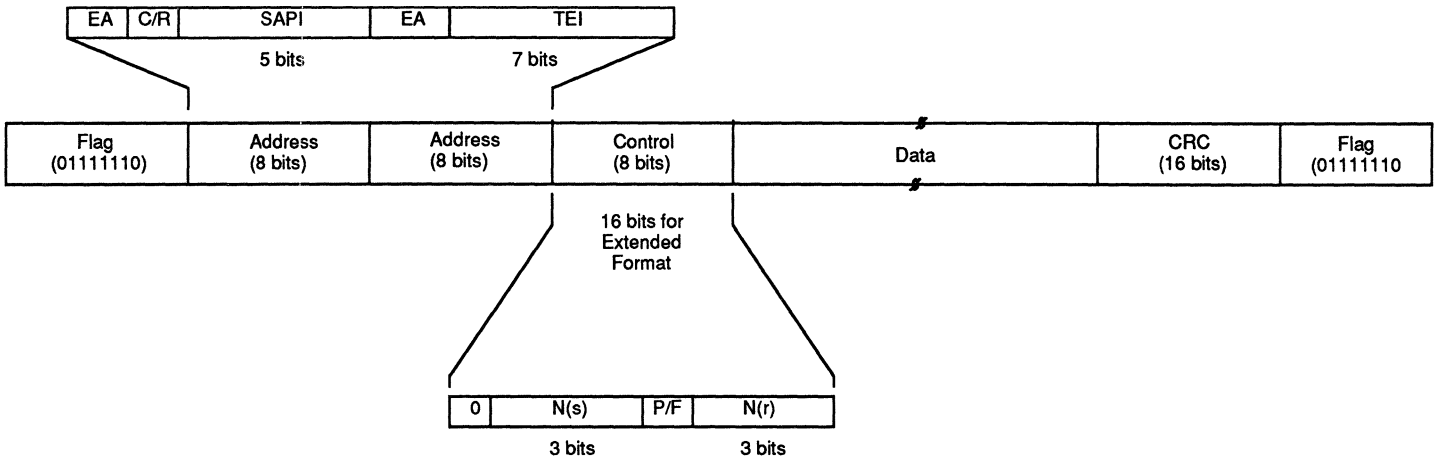


Figure 4a. LAPD I Frames

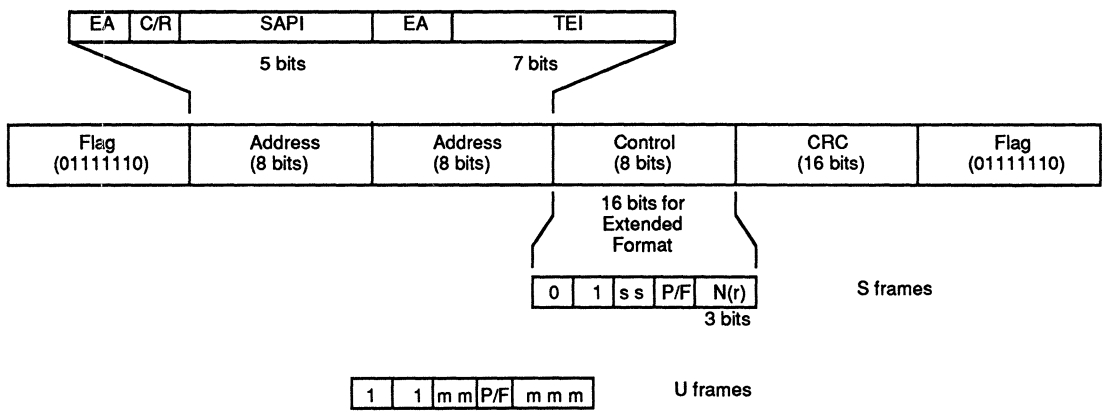


Figure 4b. LAPD S and U Frames

Am79C30A Oscillator Considerations

Care is required in the selection of oscillator components for microprocessors and peripherals to avoid problems in oscillator startup, stability, and accuracy. This selection process is especially critical in a communications application such an ISDN transceiver, where even a small frequency error can measurably degrade S interface performance.

INTRODUCTION

In many microprocessor and peripheral applications, the absolute frequency and stability (over temperature, power supply, and unit-to-unit variation) of the clock source is not critical; it is often sufficient that the oscillator starts reliably and runs at a nominally accurate frequency. In an ISDN S interface transceiver application, however, the frequency accuracy is critical due to CCITT specifications for jitter and timing. Unexpected results may also arise when interfacing the DSC to commercial audio transmission test equipment. Some of these test sets do not operate as true clock slaves, but instead use internal phase lock loops to synchronize to the device under test. If the device clock is even slightly out of specification, synchronization failure or artificially low transmission performance measurements may result.

OSCILLATOR ALTERNATIVES

The designer has three basic options for clocking the DSC: using an external packaged oscillator, constructing an external crystal-based oscillator, or connecting load capacitors and a crystal to the XTAL1 and XTAL2 pins.

Use of an external packaged oscillator presents the minimum number of complications to the designer and is the most expensive solution of the three. The oscillator output is simply connected to the XTAL2 input, and the XTAL1 output is left unconnected. The frequency accuracy, supply tolerance, and stability of the oscillator is specified by its manufacturer, thus the designer does not need to be concerned with discrete component selection. In this case, the designer needs to verify that the oscillator frequency is guaranteed to be 12.288 MHz ±80 ppm in the expected operating environment, and that the oscillator output is electrically compatible with the DSC datasheet requirements for the XTAL2 input. The first point to note is that the XTAL2 input levels are not TTL-compatible; V_{IH} must be at least 0.8 V_{CC}, while V_{IL} must be less than 0.8 volts. This is important to note since packaged oscillators are available with a variety of output characteristics, including CMOS and TTL. In the event that the source oscillator provides TTL levels, a CMOS buffer with TTL-compatible input levels may be used to provide the correct levels to the DSC. It is important to note that adding a pullup resistor to a TTL-output device is absolutely *not* an acceptable method of creating higher levels for clock inputs on CMOS or NMOS devices. On the Low to High transition, the output driver will only actively drive to typically 3.4 V, after which the output voltage will rise exponentially with a time constant dependent on the pullup resistor and parasitic characteristics of the driver and load. The rise time, duty cycle, and jitter of the resulting clock may be unacceptable and will vary greatly with temperature, device, and power supply level and noise.

Constructing an external crystal-based oscillator is an alternative when designing with the DSC, but presents design complexities without offering any compensating benefits. Use of the DSC on-chip oscillator is much simpler than an external design, and the MCLK output is available if it is desired to clock other devices. It is recommended not to attempt to drive other devices by connecting a buffer to the XTAL1 output; the extra load is an additional variable that may negatively impact oscillator performance.

Use of the on-chip oscillator is the most cost-effective approach, and will yield excellent results as long as precautions in component selection and placement are observed. To understand the problems that may be encountered, it is beneficial to review some basic properties of quartz crystals.

QUARTZ CRYSTAL PROPERTIES

Quartz is a piezoelectric material, meaning that an internal E field will form in response to applied mechanical stress. Conversely, application of an external voltage will cause mechanical deformation. It follows that if a quartz sample is driven with an AC voltage, it will cause the crystal to mechanically vibrate. The reactance of the quartz crystal does not vary monotonically with frequency, but instead becomes positive over ranges of frequency that correspond to the natural vibrational modes of the crystal. The fundamental idea behind the

Publication #	Rev.	Amendment	Issue Date:
12615	A	/0	7/89

design of a crystal oscillator circuit is to force the crystal to oscillate at or near a preselected natural frequency of vibration.

The lowest natural frequency of vibration is called the fundamental mode, and is illustrated in Figure 1. There are usually higher modes of vibration, called mechanical overtones, which are not necessarily frequency harmonics of the fundamental mode. In addition, there may be numerous "parasitic" responses where the reactance becomes positive over a narrow frequency range. Generally, crystals are intended to be operated in their fundamental mode, but it is possible to realize oscillation at a selected overtone by using a combination of careful oscillator design and crystal manufacture.

CRYSTAL RESONATOR MANUFACTURE

Manufacture of a crystal resonator begins with the slicing of a crystal wafer from a larger crystal. It is important to note that many key crystal properties are highly dependent on the "cut" of the wafer (that is, the relation between the sample faces and the crystal axes). For instance, the favorability of the various vibrational modes depend greatly upon the cut of the quartz employed. The variation of frequency with respect to temperature is also dependent upon the cut. Thus, the selection of cut is made based upon the intended application of the crystal.

After rough shaping, the crystal is planed in order to coarsely set the frequency. The thinner a wafer of quartz is, the higher the frequency of oscillation will be. Fundamental mode crystals are limited to approximately 40 MHz, since at higher frequencies the wafers become too thin to withstand mechanical vibration. For higher frequencies, crystals are cut to favor an overtone mode of vibration.

After planing, the wafer is fitted with electrodes and a holder. It is common at this point to make the final adjustment of frequency by placing the crystal assembly into a test oscillator circuit and carefully plating the crystal with silver or aluminum. The addition of mass reduces the frequency of vibration, and is continued until the desired frequency is achieved. The assembly is hermetically sealed after plating, and is referred to as a crystal resonator.

For low applied voltages the crystal deformation is elastic, but application of an excessive voltage will result in mechanical deformation beyond the elastic limit of the crystal and it will shatter. The power dissipation in a crystal is specified as its Drive Level, but this parameter is not significant in a semiconductor application.

QUARTZ CRYSTAL CIRCUIT MODEL

For the purpose of circuit analysis, it is convenient to represent a quartz crystal by the equivalent circuit illustrated in Figure 2. The series RLC components are often referred to as the motional parameters, while C_0 represents the capacitance of the electrodes and holder. The series resistance does not play a very significant role in oscillation, as long as the circuit used to drive the crystal has sufficient gain. In general, series resistance should be as low as possible. The crystal equivalent model is said to be in series resonance when L_1 and C_1 are in resonance, that is:

$$F_s = (2 \cdot pi)^{(-1)} \cdot (L_1 \cdot C_1)^{(-1/2)} \qquad \text{eq. 1}$$

This equation assumes that C_0 may be neglected. At frequency F_s the reactance of the crystal equivalent circuit is zero, as illustrated in Figure 1. It is possible to construct circuits in which a crystal will appear purely resistive and oscillate at its series resonance frequency. However, in semiconductor applications, it is not common practice to operate a crystal at series resonance.

THE PARALLEL RESONANT OSCILLATOR

Figure 3 illustrates a typical "parallel" resonant oscillator circuit, where the crystal is used in conjunction with two load capacitors C_{ld} (assumed to be equal for simplicity) and a parasitic capacitance C_{par}, which represents trace

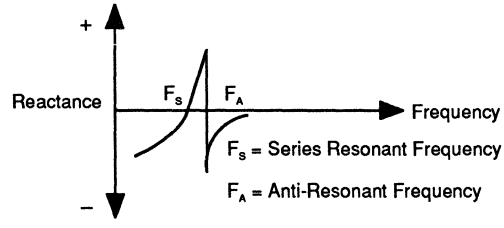


Figure 1. Crystal Reactance versus Frequency

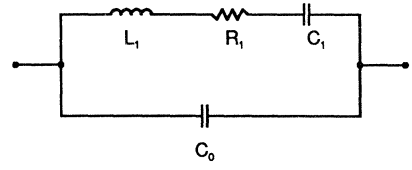


Figure 2. Crystal Equivalent Circuit

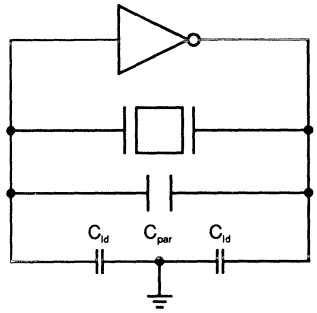


Figure 3. Parallel Resonant Oscillator

and pin capacitance, along with imperfections of the amplifier. It is easily seen that the three capacitors may be represented by a single parallel capacitance:

$$C_L = (1/2) \cdot C_{ld} + C_{par} \qquad \text{eq. 2}$$

The circuit shown in Figure 3 will oscillate at the frequency where the (negative) reactance of C_L is cancelled by the (positive) reactance of the crystal. From Figure 1, it is apparent that the reactance of the crystal in the fundamental mode is positive between frequencies F_S and F_A. Thus, depending upon the load capacitor value C_{ld} and the parasitic capacitance C_{par}, the circuit in Figure 3 will oscillate at a frequency somewhere between F_S and F_A. For many applications the precise frequency of oscillation is not critical, but for applications such as communications it may be necessary to control the frequency more tightly.

CALCULATION OF FREQUENCY VARIATION

The fact that oscillation frequency varies with load capacitance is of great significance in both crystal specification and oscillator component selection. In terms of specification, it is necessary to understand how crystals are calibrated. There is no such thing as a parallel-resonant crystal; the term refers to the method of calibration. When a crystal specification states, for instance, that a crystal is "12.288 MHz, ±0.05%, parallel resonant, 20 pF," it means that the frequency was calibrated to within 0.05% in the presence of a 20-pF series capacitance. It does *not* mean that the crystal requires 20-pF load capacitors! The value of capacitance required depends upon the amplifier circuit, and to a lesser degree upon the board layout. The following equation describes the relationship between frequency deviation (F_d) from the series resonance point and the crystal circuit parameters:

$$F_d = (F_S \cdot C_0) / [2 \cdot r \cdot (C_0 + C_L)] \qquad \text{eq. 3}$$

where $r = C_0/C_1$, and C_L is the total added load capacitance. Note that this equation indicates that not all crystals will respond identically to a change in load capacitance. Let us now apply Equations 1 through 3 to the following actual crystal parameters to illustrate how the various parameters affect oscillator operation:

$F_S = 12.283823$ MHz

$C_0 = 4.25$ pF

$R_1 = 7.7$ ohms

$C_1 = 0.0165746$ pF

$L_1 = 10.128$ mH

12.288 MHz, ± 20 ppm

Parallel resonant, 20.12 pF

Plugging the above values into Equation 3 yields a frequency offset F_d of 4177 Hz. It is easily verified that this corresponds to the difference between 12.288 MHz and 12.283823 MHz. Now suppose we would like to determine the tolerance of capacitance required to ensure 80-ppm frequency accuracy. Since the crystal frequency is calibrated to within 20 ppm, we will specify the tolerance of the external components to allow for an operating range of ±60 ppm. This corresponds to an operating range as follows:

$$12.28726 \text{ MHz} < f < 12.28874 \text{ MHz}$$

The lower limit corresponds to a frequency offset of 3440 Hz from the series resonance point, while the upper limit corresponds to an offset of 4920 Hz. Plugging F_d = 3440 Hz into Equation 3 yields a C_L value of 25.34 pF, while F_d = 4920 yields a C_L value of 16.44 pF. Note that increasing the load capacitance decreases the operating frequency. The 60-ppm frequency accuracy requirement can be met over the range:

$$16.44 \text{ pF} < C_L < 25.34 \text{ pF} \qquad \text{eq. 4}$$

The parasitic capacitance C_{par} of Equation 2 is on the order of 5–8 pF, and can be determined empirically by changing load capacitors and applying the foregoing equations. It is important to note that the value C_{par} will vary slightly from one device to another, thus not all of the error tolerance should be allotted to the load capacitors. Assuming an empirically determined C_{par} of average value 5 pF, selecting load capacitors of value 30 pF and 5% tolerance allows C_{par} to vary between approximately 2.19 and 9.59 pF, which provides an adequate margin.

RECOMMENDATIONS

It is apparent in Equation 2 that the equivalent load seen by a crystal depends upon both the load capacitors and the parasitic capacitances present. For this reason, it is desirable to minimize the variability of parasitic capacitance by placing both the crystal and the load capacitors as close as possible to the IC oscillator pins. This technique has the added benefits of minimizing the capacitive coupling of other signals into the oscillator, and reducing the amount of clock noise coupled into other circuits. It is also preferable to operate with larger load capacitances, since the percentage contribution of both load capacitor error and parasitic capacitance is reduced. If the load capacitor value becomes too large, however, oscillator startup problems may ensue. For power-critical applications, it should be noted that increasing load capacitance will result in increased power consumption.

Although the equations are useful to gain insight into the problem, there is no substitute for actual measurement. It is recommended to select a load capacitor value for a given crystal by lab measurement of the DSC MCLK output frequency (so as not to load the oscillator circuit), and then using the manufacturer's parameter values in the foregoing equations as a check. This procedure need only be done during initial design; it does not need to be repeated on a per-unit basis. It is also recommended to select load capacitors with minimum temperature variation, for instance, ceramic NPO capacitors.

REFERENCES

"Quartz Crystal Devices," John R. Vig, *Electronic Engineers' Handbook*, McGraw-Hill, 1982.

Design of Crystal and Other Harmonic Oscillators, Benjamin Parzen, John Wiley and Sons.

Introduction to Quartz Crystal Unit Design, Virgil E. Bottom, Van Nostrand Reinhold.

Designing Quartz Crystals into Integrated Circuits and Microprocessors, John B. Fisher, Standard Crystal Corporation Application Note.

The author would like to gratefully acknowledge many helpful inputs from the applications staff of Saronix Corporation, Palo Alto, California.

Electret Handset Interface

The electret-type microphone, incorporating a capacitive microphone and built-in FET, is rapidly becoming the microphone of choice for telephone applications. A bias circuit is required for the FET, along with a gain stage for pre-amplification of the low output voltage.

THE ELECTRET HANDSET

The electret handset is a popular choice among telephone designers for a variety of reasons. It is inexpensive, light, and requires less bias current than the conventional carbon-type microphone. Furthermore, the microphone provides excellent dynamic range and noise performance, and its small physical size makes it suitable for many different handset designs.

The electret microphone provides an output voltage when pressure from the speaker's voice varies the charge on an internal capacitor, which in turn varies the gate voltage of a FET. Since a drain current must exist for the FET amplifier, the electret microphone requires an external bias resistor.

The microphone satisfies part of its gain requirement with the FET amplifier, but requires an additional gain stage to boost the signal up to a level compatible with the Am79C30A MAP receiver. It should be noted that the bias and gain requirements will vary from one handset manufacturer to another, and may require some adjustment for compliance with any applicable sound pressure level tests. For the handset tested, a gain of approximately 24 dB was selected to provide a subjectively comfortable listening level.

BIAS AND GAIN CIRCUITRY

The circuit shown in the attached figure was designed with several objectives. First, it was designed to operate using only a 5-V power supply. Second, it was essential that the circuit would perform adequately with an inexpensive operational amplifier. Finally, it was necessary that the circuit would provide satisfactory performance in a noisy environment, for instance, on a personal computer board.

In Figure 1, the equivalent circuit for the electret microphone shows a variable capacitor driving a FET. The series combination of the 6.8-kohm and 3.3-kohm resistors provides approximately 0.5 mA of bias current for the FET. The 6.8-kohm resistor and the 20-microfarad capacitor serve to filter noise from the 5-V power supply, since any power supply noise would otherwise appear at the microphone output and be inseparable from the voice signal. This is especially critical since the noise would be boosted by 24 dB through the amplifier.

The operational amplifier is a typical inverting configuration, with the positive input biased to approximately 2.4 volts by the DSC AREF signal. The 10-kohm and 1-µF capacitor to ground form an RC filter to remove any low-level switching noise from the AREF signal, because this noise could be objectionable after being boosted by the amplifier. The 560-pF capacitor in the amplifier feedback path removes high frequencies from the signal path, thereby ensuring that they are not converted to in-band noise by operational amplifier imperfections. The value of this feedback capacitor may need to be varied for different models of op-amps.

The 1-µF capacitor on the output of the amplifier is necessary to AC couple the signal to the analog input of the DSC. The 15-kohm resistor ensures that the analog input is biased with the voltage provided by the DSC AREF pin. The analog input must be biased by the AREF voltage, otherwise degraded transmission performance at low signal levels will occur.

TEST RESULTS AND RECOMMENDATIONS

The described circuit was built and tested using low-cost TLC271CP and TL061ACP operational amplifiers manufactured by Texas Instruments. A Wandel and Goltermann PCM4 test set was used to perform Signal to Distortion, Gain Tracking, Idle Channel Noise, Relative Group Delay, and Absolute Gain measurements in accordance with CCITT Recommendation G.714. All tests were passed with satisfactory margins using the aforementioned op-amps on a four-layer PCB installed in an IBM PC-AT.

In designing audio interfaces for the Am79C30A DSC, it is important to note that the weakest link in the audio path is the circuitry external to the DSC. Care must be taken to ensure that microphone bias current is adequately filtered. Board layout is also critical to optimize audio performance. Audio traces should be as short as possible, and routed away from high-speed digital signals such as clocks. Splitting the ground and V_{CC} planes

Publication #	Rev.	Amendment	Issue Date:
12616	A	/0	7/89

of the PCB is an effective technique and is described in a companion application note on DSC Layout Hints. The ground points of the various analog components of the microphone circuit should be placed as close as possible to one another on the board in order to minimize the effect of different ground current return paths.

In prototyping analog circuitry, it must be understood that wirewrap boards and high-gain audio amplifiers do not mix very well. Long wires, crosstalk, insufficient power supply decoupling, stray inductances and capacitances, and poor grounding all contribute to a higher noise floor and degraded amplifier stability.

As a final precaution, note that care must be taken in op-amp selection. The amplifier must be capable of running on 0- and 5-V power supplies and must have sufficient voltage swing around the nominal 2.4-V AREF voltage to handle the maximum expected audio signal without distortion.

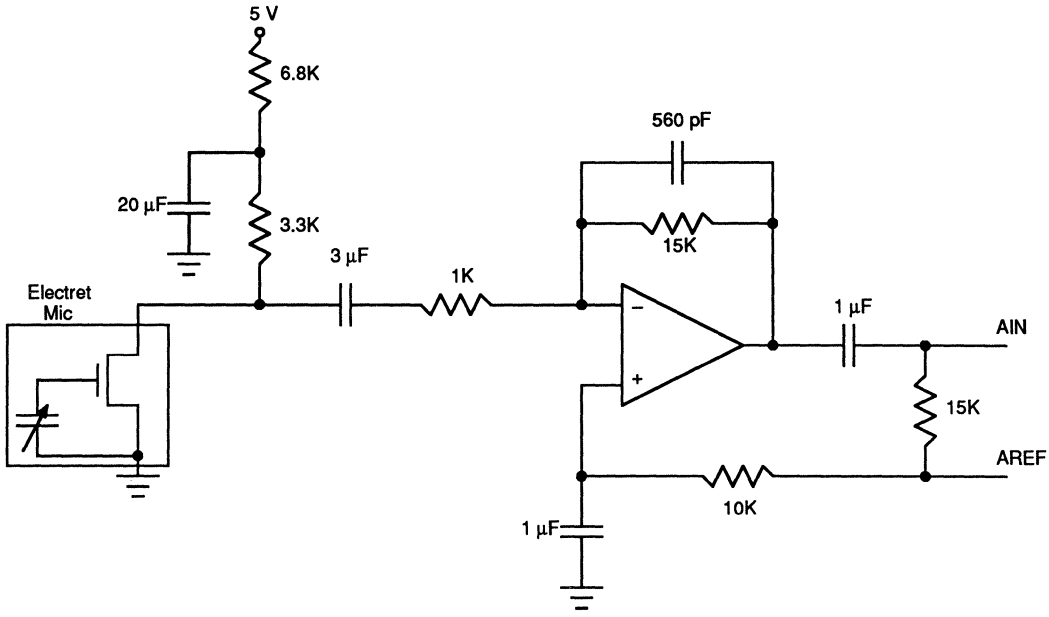


Figure 1. Electret Handset Interface

Protocol Reference Model Tutorial

The introduction of the Open System Interconnect (OSI) model for networks has had a profound effect on data communications and distributed applications processing. The acceptance of this standard continues to widen as the OSI model was selected as the basic structure for the software-intensive ISDN. It is important for engineers working in this area to be familiar with the basic terminology and concepts of the OSI reference model and the developing ISDN reference model.

The data communications industry in the early 1970s was characterized by incompatibility between product offerings from different vendors. Although some of these differences were due to a lack of applicable standards, manufacturers were also reluctant to share design information because they wanted to protect their customer bases. As the size of the market grew, it became increasingly apparent that it was in everyone's best interest to provide the capability to interconnect equipment from different manufacturers. In 1978, the International Standards Organization (ISO) began work on developing the "Open System Interconnect" reference model to provide a framework for the orderly communication of "application processes" across different computer networks. These application processes may be human users or computer programs. The committee considered input from a variety of international trade and standards organizations and produced a working standard within three years. The work of this committee was also supported by the CCITT in their recommendation X.200.

The framework chosen by the ISO is a seven-layer structure in which each layer provides specialized services that logically belong together. There are, of course, many different ways in which the functions may be organized, and the standard acknowledges that it would be difficult to provide the optimality of any given approach. In order to achieve agreement, the committee was also required to take note of existing solutions and standards. A brief description of each layer, its purpose, and services provided follows.

The Physical Layer is the lowest layer of the architecture and, as defined in CCITT X.200, "provides mechanical, electrical, functional, and procedural means to activate, maintain, and deactivate physical connections for bit transmission between data link entities." For instance, the physical layer may include functions such as connector types and pinouts, line code definitions, clocking and timing requirements, voltage levels, block formats, activation and deactivation procedures, loopback modes, and error notification. A good example of a physical layer protocol is RS-449. In the ISDN environment, a key physical layer protocol is CCITT I.430, which describes the four-wire S or T interface.

The Data Link Layer resides directly above the physical layer and provides for the transfer of units of information between the two ends of the physical link, as well as the establishment and release of the data link connection. Framing, flow control, error detection, and error correction are possible functions to include in this layer. Multiplexing, another key feature of this level, provides multiple logical connections on a single physical link. HDLC and SDLC are examples of common Layer 2 protocols. In the ISDN application, CCITT Q.921 (LAPD) is the key Layer 2 protocol for data link signaling on the D channel.

The Network Layer is the third layer of the architecture and is responsible for the addressing, switching, and routing functions that are required to guarantee transparent transmission of data through a system of computers. It is important to note that a network may not always be sourcing or sinking data, but may be performing a relay function as illustrated in Figure 13/X.200. The switching of data through an ISDN is an example of a relay function. The network must guarantee transparency of such a relay function to higher protocol layers. In the ISDN application at Layer 3, CCITT Q.931 provides call setup and teardown functions. However, in the special case of "nailed-up" D-channel data service, there is no routing to be done, and only X.25 is required at Layer 3.

The Transport Layer is responsible for providing the required performance at minimum cost based upon the current state of the network, and is the lowest layer that is concerned with end-to-end functions. As seen in Figure 13/X.200, no transport layer service is required for the intermediate system in a relay network.

The Session Layer coordinates the interaction of the application processes at each end regarding session establishment, release, management, and error reporting. There is no flow control at the session layer.

Publication #	Rev.	Amendment	Issue Date:
12617	A	/0	7/89

The Presentation Layer provides for the conversion of data between different representations, for instance, code, character, or format conversions.

The Application Layer provides all the services directly required by the application process, since it is the only layer in the model that communicates directly to the application processes. Services will vary greatly depending upon the application, including, for example, identification of an intended communication partner by name or address, or determination of the availability of the requested party.

One of the objectives in the definition of the OSI layers was to simplify the amount of communication required between layers. Each layer, except the application layer, is defined to provide services to the next higher OSI layer. Adjacent layers interact with predetermined requests and responses, called primitives. Note that the OSI model does not attempt to define primitives, but leaves this to the individual protocols used in specific applications. At each layer there is a "peer protocol" to govern the interaction between entities in the layer, as illustrated in Figure 12/X.200.

Entities within a layer can communicate only through the services provided by the next lower layer, and it is assumed that physical layer entities interact directly.

In spite of the detailed description of the functions in each layer, it is important to note that the OSI model does not constrain the implementation architecture; the definition of system management functions is left completely to the user. The CCITT is working on a series of recommendations for Management Entities, but they are a long way from completion and are far more complex than the OSI model itself.

Figure 13/X.200 illustrates communication through a relay system that passes data between two systems that are only connected through the intermediate network. An ISDN may be viewed as such a relay system, but with the possibility of ongoing participation even after a call is established. In recent years the CCITT has been developing Recommendation I.320, the objective of which is to model the interconnection and exchange of user information and control information in an ISDN environment. In this sense, I.320 may be viewed as a specific application of the principles of X.200 and not as a radically different recommendation. Although I.320 is still evolving, it is useful to examine the emerging ISDN Protocol Reference Model. In the February 1988 draft of I.320, it is stated that "the support of outband signaling, the ability to activate supplementary services during the active phase of the call, imply a separation between control information and user information." To reflect this separation, I.320 introduces the concept of a control plane (C plane) and user plane (U plane). Both of these planes have seven layers in the manner of the OSI model, but not all layers are necessarily present. The purpose of the U plane is the transfer of information among user applications, whereas the C plane is intended to establish or modify a network connection or to provide supplementary services. The C and U planes do not connect directly, but instead communicate via primitives through the plane management function M, which is not layered (at this time). This basic reference model is illustrated in Figure 2/I.320, which is reproduced from the 1985 version of I.320. In the February 1988 draft of I.320, the model was further refined to split the C plane into local and global control planes. This alteration was made to reflect the fact that control data handled by an entity may concern an adjacent (local) entity or non-adjacent (remote or global) entity. The resultant Generic Protocol Block is illustrated in Figure 3/I.320, but it must be emphasized that I.320 is not finalized and thus is subject to considerable change.

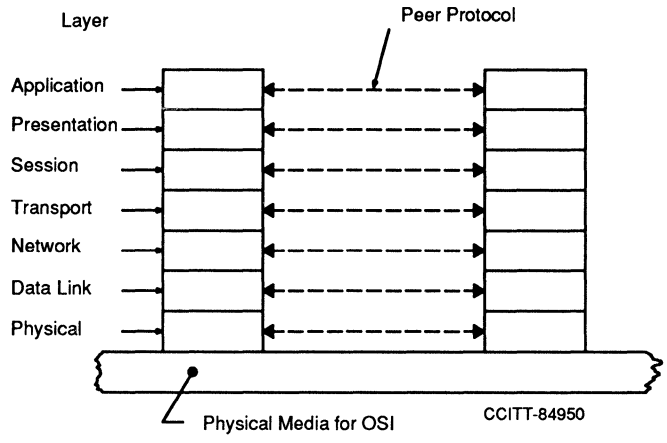


Figure 12/X.200. Seven-Layer Reference Model and Peer Protocols

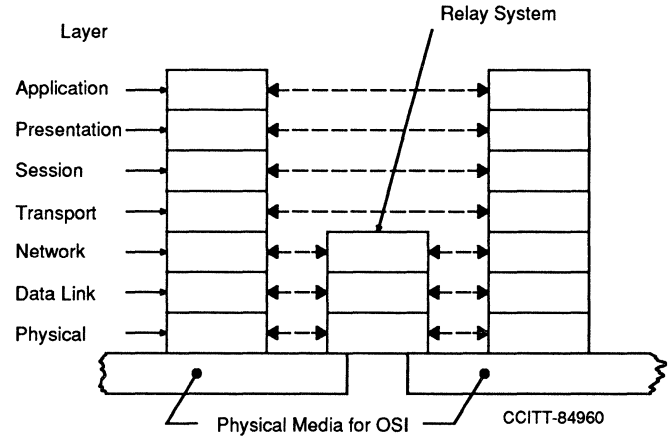
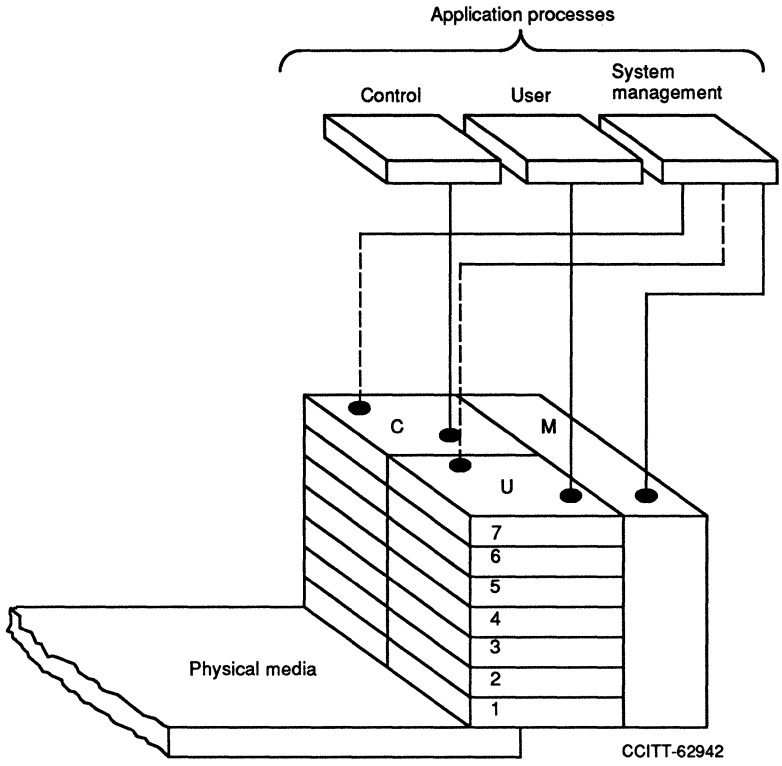


Figure 13/X.200. Communication Involving Relay Open Systems



Note: Peer-to-peer protocols associated with U and C are not shown.

Figure 2/I.320. Interactions Associated with a Protocol Block

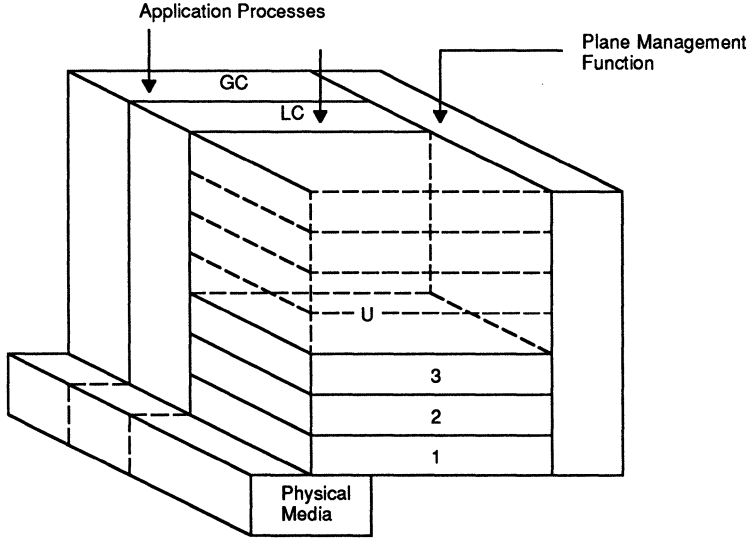


Figure 3/I.320. Generic Protocol Block

RECOMMENDED FOR FURTHER READING

OSI Explained: End-to-End Computer Communications Standards, John Henshall and Sandy Shaw, John Wiley & Sons, 1988.

Telecommunication Technology, R.L. Brewester, John Wiley & Sons, 1986.

Telecommunications Networks, Protocols, Modeling, and Analysis, Mischa Schwartz, Addison-Wesley, 1987.

CCITT Recommendation X.200: "Reference Model of Open Systems Interconnection for CCITT Applications."

CCITT Recommendation X.210: "Open System Interconnection Layer Service Definition Conventions."

CCITT Draft Recommendation I.320: "ISDN Protocol Reference Model."

Key Design Hints for the DSC/IDC

Due to the high level of integration of the Am79C30A/32A DSC/IDC, it is easy to overlook important design information when reading the data sheet. The following list of key design hints has been compiled to streamline the design process. A comprehensive series of ISDN application notes and tutorials is available from Advanced Micro Devices; please contact an AMD sales office or factory for current information.

1. The AREF pin *must* be used to bias the AINA and AINB inputs. There is a data sheet parameter Vios that states that the analog inputs must be biased to within 5 mV of AREF. AREF is *nominally* 2.4 volts; normal device-to-device variation will exceed the 5-mV Vios specification. If a voltage other than AREF is used, transmission performance at low signal levels will be degraded.

2. The recommended method of biasing the AINA or AINB inputs is to use a 15–100 kohm resistor between the input and AREF. The signal source should be AC-coupled to the analog input. Care should be taken such that the RC formed by the biasing resistor and blocking capacitor does not distort the input signal. A 3-dB point below 10 Hz is recommended.

3. The AREF output must not be loaded with a capacitor, since it may cause the internal buffer amplifier to become unstable. For some applications involving significant gain external to the DSC, the AREF output may require a simple RC noise filter. In this case, the AREF output should be isolated from the capacitor by a resistance of greater than 1 kohm to ensure stability.

4. The DSC/IDC should be provided with decoupling capacitors, situated as closely as possible to the package power leads. In general, 0.1-µF ceramic capacitors are sufficient, but bulk decoupling capacitors will be required if the LS1 and LS2 loudspeaker outputs are driving a heavy load.

5. The DSC/IDC is constructed on a single substrate, and therefore the device power pins must not be from separate supplies. If there is a DC offset between the analog and digital power supply pins, excessive current may flow through the device substrate.

6. The LS1, LS2, and EAR1, 2 outputs are intended to be used differentially. Although it is possible to use only a single output, the rejection of power supply noise and internal digital noise is improved if the outputs are used differentially.

7. It is necessary to observe the maximum loading specification for the LS and EAR outputs. When used differentially, the EAR outputs must see a minimum of 540 ohms between them. Similarly, the LS outputs must see a minimum of 40 ohms. The maximum capacitive loading in either case is 100 pF.

8. The LS and EAR outputs do not need to be "matched" to the load. The LS and EAR outputs are voltage drivers and do not assume the presence of any particular load impedance. As long as the maximum loading specification is met, the LS and EAR outputs will function satisfactorily. In some cases an external resistor may be used to center the desired output volume, for instance, while driving a 150-ohm earpiece with the EAR outputs.

9. For Revision D and prior releases, it is not recommended to unnecessarily access the ATGR, FTGR, GX, GR, GER, STG, X, R, MMR1, and MMR2 registers while audio is active. When the microprocessor accesses these registers, it may interfere with the MAP calculations, resulting in degraded transmission performance or audible noise.

10. When using programmable gains and filters in the MAP, it is necessary to consider dynamic range effects such as truncation error and clipping. In case of questions in any particular application, please contact AMD applications staff for assistance.

11. All MAP tone generators are referenced with respect to the +3-dBm0 overload voltage, that is, a 0-dB tone yields a +3-dBm0 output. Care must be taken to avoid clipping when adding tones to signals as, for example, when generating DTMF waveforms.

12. The RC connected to CAP1/CAP2 must be situated as close as possible to the DSC package to reduce the amount of noise coupled in from other signal traces.

Publication #	Rev.	Amendment	Issue Date:
12618	A	/0	7/89

13. It is necessary to observe the XTAL2 frequency accuracy requirement of 12.288 MHz, ±80 ppm. Since crystals from different manufacturers will vary, it is necessary to measure the DSC oscillator output frequency at the MCLK pin and, if necessary, adjust the value of the crystal load capacitors as part of the initial design procedure. An application note of oscillator considerations is available from AMD.

14. If driving the XTAL2 pin with an external oscillator, it is necessary to observe the data sheet input voltage and rise/fall time requirements. Note that the XTAL2 levels are not TTL-compatible.

15. The DSC, as any sensitive analog device, requires care in board layout. An application note of DSC board layout hints is available from AMD.

16. The sidetone path defaults to –18 dB attenuation. If it is desired to disable the sidetone path, it is necessary to enable the sidetone block and program it for infinite attenuation.

17. The LIU transformers, series resistors, and IC LIU output drivers must be considered as a functional unit. Transformers that meet CCITT I.430 requirements with other transceivers are not necessarily appropriate for use with the DSC, and vice versa. An application note of DSC transformer and series resistor considerations is available from AMD.

18. Interrupts should be masked when reading or writing any indirect or multi-byte DSC registers, to prevent the possibility of an interrupt occurring and destroying the contents of the Command Register.

General Information

This section was excerpted from Chapter 1 of the *Z85C30 Technical Manual*, order #07513C.

INTRODUCTION

The Z85C30 and Z8530 SCCs (Serial Communications Controller) are dual channel, multiprotocol data communications peripherals designed for use with 8- and 16-bit microprocessors. The SCC functions as a serial-to-parallel, parallel-to-serial converter/controller. The SCC can be software-configured to satisfy a wide variety of serial communications applications, including: Bus Architectures (full- and half-duplex), Token Passing Ring (SDLC Loop mode), and Star configurations (similar to SLAN).

The SCC contains a variety of internal functions including on-chip baud rate generators, digital phase-lock loops, and crystal oscillators, which dramatically reduce the need for external logic. In addition, SDLC/HDLC enhancements have been added to the Z85C30 that allow it to be used more effectively in high speed applications.

The SCC handles asynchronous formats, synchronous character-oriented protocols such as IBM Bisync, and Synchronous bit-oriented protocols such HDLC and IBM SDLC. This versatile device supports virtually any serial data transfer application (telecommunications, cassette, diskette, tape drivers, etc.).

The device can generate and check CRC codes in any Synchronous mode. The SCC also has facilities for Modem controls in both channels. In applications where these controls are not needed, the Modem controls can be used for general purpose I/O.

With access to 14 Write registers and 7 Read registers per channel, the user can configure the SCC so that it can handle all asynchronous formats regardless of data size, number of stop bits, or parity requirements. The SCC also accommodates all synchronous formats including character, byte, and bit-oriented protocols.

Within each operating mode, the SCC also allows for protocol variations by handling odd or even parity bits, character insertion or deletion, CRC generation and checking, break/abort generation and detection, and many other protocol-dependent features.

Unless otherwise stated, the functional description in this Technical Manual applies to both the NMOS Z8530 and CMOS Z85C30. When the enhancements in the Z85C30 are disabled, it is completely downward compatible with the Z8530.

CAPABILITIES

- Two independent full-duplex channels
- Synchronous/Isosynchronous data rates:
  - Up to 1/4 of the PCLK (i.e., 4 Mbit/sec. maximum data rate with 16 MHz PCLK Z85C30)
  - Up to 1Mbit/sec. with a 16 MHz clock rate (FM encoding using DPLL in Z85C30)
  - Up to 500 Kbit/second with 16 MHz clock rate (NRZI encoding using DPLL in Z85C30)
- Asynchronous capabilities:
  - Up to 250 Kbits/sec with 16 MHz (x16 mode) PCLK
  - 5, 6, 7, or 8 bits per character
  - 1, 1-1/2, or 2 stop bits
  - Odd or Even Parity
  - x1, 16, 32, or 64 clock modes
  - Break generation and detection
  - Parity, Overrun and Framing Error detection
- Character-Oriented synchronous capabilities:
  - Internal or external character synchronization
  - 1 or 2 sync characters in separate registers
  - Automatic CRC generation/detection
- SDLC/HLDC capabilities:
  - Abort sequence generation and checking
  - Automatic zero bit insertion and deletion
  - Automatic flag insertion between messages
  - Address field recognition
  - I-Field residue handling
  - CRC generation/detection
  - SDLC Loop mode with EOP recognition/loop entry and exit
- Receiver data registers quadruply buffered. Transmitter data register doubly buffered
- NRZ, NRZI, or FM encoding/decoding and Manchester decoding
- Baud-rate generator in each channel
- A DPLL in each channel for clock recovery
- Crystal oscillator in each channel
- Local Loopback and Auto Echo modes

In addition, the Z85C30 provides enhancements which allow it to be used more effectively in high speed SDLC/HDLC applications. These enhancements include:

- 10 x 19-bit SDLC/HDLC frame status FIFO
- 14-bit SDLC/HDLC frame byte counter
- Automatic SDLC/HDLC opening Flag transmission
- Automatic SDLC/HDLC Tx Underrun/EOM Flag reset
- Automatic SDLC/HDLC CRC generator preset
- TxD forced High in SDLC NRZI mode when in mark idle
- RTS synchronization to closing SDLC/HDLC Flag
- DTR/REQ DMA request deactivation delay reduced
- External PCLK to $\overline{\text{RTxC}}$ or $\overline{\text{TRxC}}$ synchronization requirement removed for one fourth PCLK operation
- Reduced Interrupt response time
- Reduced Read/Write access recovery time (Trc) to 3 PCLK best case (3 1/2 PCLK worst case)
- Improved WAIT timing

Other enhancements which make the Z85C30 more user friendly include:

- Write data valid setup time to negative edge of write strobe requirement eliminated
- Write Registers WR3, WR4, WR5 and WR10 are readable
- Complete reception of SDLC/HDLC CRC characters
- Lower priority interrupt masking without $\overline{\text{INTACK}}$ generation

INTRODUCTION

This section was excerpted from the *Am79C401 Technical Manual*, order #09559B.

The Am79C401 Technical Manual provides information to the user concerning the operation and programming of the major functional modules contained in the Am79C401 Integrated Data Protocol Controller™ (IDPC™). This manual is divided into five chapters and one appendix.

Introduction—This chapter provides an overview of the IDPC, concentrating on how it fits into various system architectures.

Hardware—The Hardware chapter covers the specifics of each major functional block, emphasizing how each block is controlled, and the operation and generation of external interface signals.

Applications—This chapter provides detailed design examples, concentrating on the IDPC's external interfaces.

Programming the IDPC—The Programming chapter contains three sections. The first section introduces the programmable features of the IDPC. The second section provides a series of tutorials on system-level operation. The final section consists of detailed programming examples for the major functional blocks.

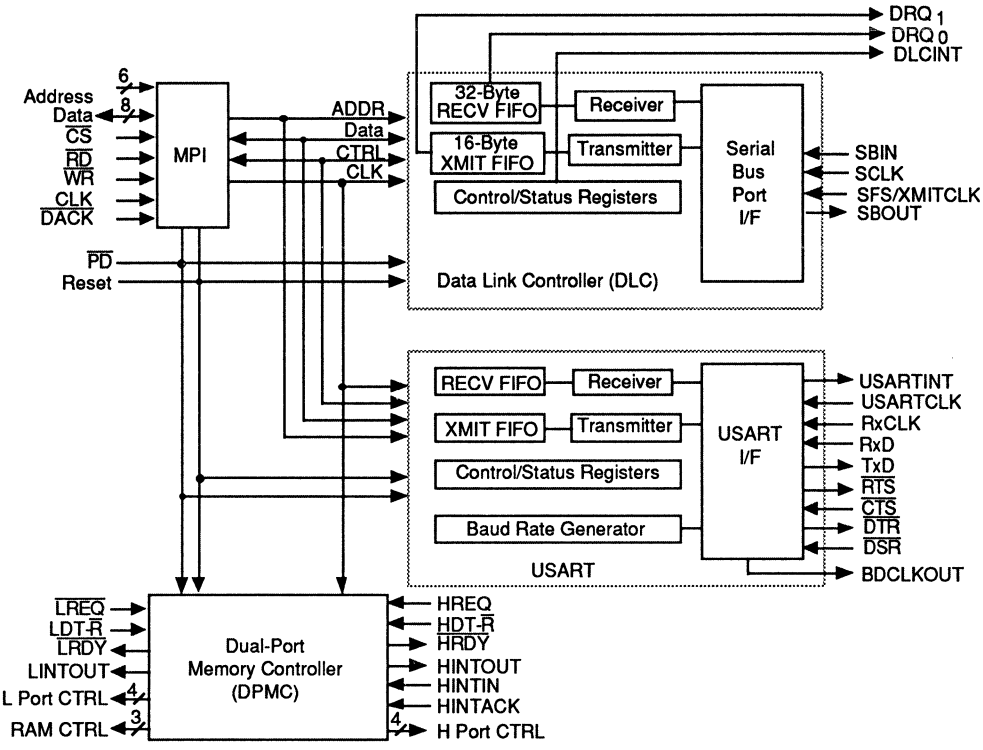
Low-Level Device Driver—This chapter contains the Reference Guide for the Am79LLD401 Low-Level Device Driver. This software interfaces the IDPC to the AmLink™ LAPD/LAPB software package. The source code, with an unlimited binary distribution license for both software packages, may be purchased from AMD for a nominal one-time fee.

Appendix—The Appendix contains pin definitions and user-accessible register descriptions for the Am79C401 IDPC.

INTRODUCTION TO THE Am79C401 IDPC

When designing equipment for packet data networks, designers are concerned with performance, flexibility, and cost. The Am79C401 Integrated Data Protocol Controller (IDPC) from Advanced Micro Devices addresses these three concerns by integrating three key building blocks into a single integrated circuit. As shown in Figure 1-1, the Am79C401 IDPC consists of three major blocks: the Data Link Controller (DLC), the Universal Synchronous/Asynchronous Receiver/Transmitter (USART), and the Dual-Port Memory Controller (DPMC). The DLC is the heart of the Am79C401 IDPC; it is responsible for processing bit-oriented protocols such as the HDLC and its derivatives SDLC, LAPB, and LAPD. The USART block is a super-set of the industry-standard 8250 UART. The USART is useful in building terminal adapter devices that connect existing terminals to Bit-oriented Protocol-based networks such as X.25, ISDN, and SNA. The Dual-Port Memory Controller (DPMC) provides the circuitry required to convert inexpensive static RAM into dual-port memory. Dual-port memory is required whenever the communications processor shares the system bus with a host processor. The dual-port memory allows messages

Figure 1-1 IDPC Block Diagram



and data to be passed between the two processors. An example of this type of application is an X.25 network interface, installed in a personal computer. By integrating these basic building blocks into a single integrated circuit, the cost of building data communications products is significantly reduced.

DATA LINK CONTROLLER

Designers of data communication equipment supporting bit-oriented protocols such as HDLC, SDLC, LAPB, and LAPD have had to choose between performance and flexibility. The protocol controller integrated circuits currently on the market offer either performance or flexibility, but not both. The reason for this is that in order for one protocol controller IC to have the flexibility to handle all of the various protocols, the higher layers of the protocols must be handled by software running on an associated microprocessor. Handling part of the protocol processing in software can substantially degrade performance, as measured by overall throughput. In an attempt to increase performance, some IC manufacturers have designed ICs that are dedicated to one specific protocol, performing most, if not all, of the protocol processing in hardware. While this improves throughput, the flexibility to handle multiple protocols or even slight variations in the target protocol is sacrificed. In the development of the Am79C401 IDPC, Advanced Micro Devices has taken a different approach to the problem. Three key elements of this approach are: the careful

partitioning of the tasks to be performed in hardware versus software, separating the movement of data from the processing of packets, and the optimization of the hardware/software interface.

The Hardware/Software Boundary—Careful consideration is required in determining which of the bit-oriented protocol functions to perform in hardware versus software. Many of the functions, such as flag and abort detection, 0-bit insertion, and CRC generation and checking, are best performed in hardware. Other functions are best left to the software, including: sequence number checking, transmission of acknowledgment packets, and retransmission of unacknowledged packets. There are two reasons for handling these functions in software: (1) each of the various protocols handles these functions in a slightly different manner, and (2) the amount of hardware required increases prohibitively as the window size increases. (Window size refers to the number of packets one transmitter can send out before an acknowledgment is received regarding the first packet sent. For example, if the window size is four, the transmitter can send four packets, then it must stop transmitting until an acknowledgment is received for the first packet. This requires the transmitter to provide hardware to store a history of all outstanding packets—an expensive proposition considering window sizes of eight or more are not uncommon.) Most protocol controllers that are designed to process multiple protocols divide the above-mentioned tasks between hardware and software in a similar fashion. The DLC in the Am79C401 IDPC provides hardware support for several additional functions that are often delegated to software. Two examples of these are: minimum packet size checking and maximum packet size checking. By handling these tasks in hardware, the software does not need to perform a bounds check every time a new byte of data is received. Aside from the obvious advantage of reduced software overhead, this also allows the software to be partitioned into two separate functions: data movement and packet processing. As you will see later, this is the key to providing high performance while retaining the flexibility to handle multiple protocols.

Separating Data Movement From Packet Processing—The key factor in determining overall throughput is the rate at which packets can be processed. For reasons of flexibility, this processing takes place in software (dedicated hardware is faster, but expensive and inflexible). In the International Standards Organization's Open Systems Interconnection (ISO-OSI) seven-layer model, the bit-oriented protocol resides at Layer_2. Layer_2 is given unpacketed data from Layer_3, which it packetizes and transmits via Layer_1. On the receive side, packets are received, verified, then returned to a non-packet format, and passed up to Layer_3. While this is an oversimplified view of a fairly complex process, it does point out a key fact: the Layer_2 software deals with packets, not the movement of individual bytes of data. If the software can be partitioned such that it deals only with completely received packets of data, not bits and pieces of packets, the time spent processing the packet can be substantially reduced.

The DLC in the IDPC has been designed to completely separate the software involved in data movement from the software responsible for packet processing. One piece of software is responsible for the movement of data, often via DMA, while a separate software module processes the status information regarding complete packets. In the IDPC, a special status reporting mechanism has been designed that stores the status information concerning a packet and then reports it to the packet-processing software after the entire packet has been received, moved through the 32-byte receive FIFO, and

stored in an off-chip buffer. Up until the time that an entire packet has been received, moved from the receive FIFO, and placed in buffer memory, the packet processing software is uninvolved. In fact, the DLC does not notify the software about the packet until the last byte of the received packet has been placed in off-chip memory. At this time, the DLC notifies the software that a packet has been received, and provides the appropriate status information concerning the packet. In this manner, the packet processing software is presented with a complete packet of data and status information pertaining to that packet at the same time. If the status information indicates that the packet has been received without errors, it is acknowledged and the packet is passed to the user. If the packet contained errors or was aborted, all that is required is to reassign the buffer location in memory. The DLC can store status information for up to four previously received packets before the microprocessor has to read the status from the first packet. This greatly increases the maximum allowed interrupt latency.

Optimization Of The Hardware/Software Interface—The percentage of the processor's time that must be spent interfacing to the DLC is critical to performance. Three major factors affect this overhead: time spent identifying which register contains the pertinent information, time spent accessing that register, and time spent locating the desired information within the register. The time required to identify the register containing the condition that caused the interrupt is based on the efficiency of the interrupt-reporting structure. In the DLC, the source of an interrupt is reported via the Interrupt Source Register. This register contains bits that directly point to status registers that can generate interrupts. Additionally, the actual status information for the two most common interrupt-generating events is reported directly in the interrupt source register. These two conditions, which make up 95% of all interrupts, are the valid-packet-received and valid-packet-transmitted indicators. Once the source of an interrupt has been identified, the appropriate status register must be read. The time required to read the register can be cut in half if the register is directly mapped into the processor's address space, instead of indirectly accessed via a pointer register. In the DLC, all registers are directly memory mapped. The third factor contributing to the efficiency of the software interface is the time required to find information once a register has been read. The key to reducing this time is to organize the individual registers such that the most often required information is in either the least significant or most significant bit locations. Once a status register is read, the software typically performs a test, shift, test routine until it finds a bit that is set. The time spent finding the set bit depends on the number of shift/tests required. If the most frequent conditions are indicated by bits closest to one end of the register, the performance can be more than doubled.

Both flexibility and performance are attained by optimizing the partition between software and hardware, separating the movement of data from the processing of packet status, and optimizing the interface between software and hardware resident status registers.

USART

The function served by the Universal Synchronous/Asynchronous Receiver/Transmitter (USART) depends on the application of the IDPC. If the IDPC is embedded in a terminal or host computer (such as a PC), the USART provides a second serial channel, separate from the DLC. In terminal adapter applications, existing terminals that are not "network-ready" are inter-

faced to networks such as SNA, X.25, or ISDN. In this case, the USART provides the connection to the terminal, while the DLC provides the network interface.

The USART is a super-set of the industry standard 8250 Universal Asynchronous Receiver/Transmitter (UART). The 8250 UART provides basic asynchronous RS-232 serial data communication service, including baud rate generation, and is the standard UART used in the IBM PC™ and its compatibles. The IDPC USART starts with this base and adds three features:

- **Four-Byte Transmit and Receive FIFO Buffers**—The FIFOs increase software performance by reducing the number of interrupts that must be serviced, and increasing the time allowed to respond to an interrupt.
- **Special Character Recognition**—It is normal practice to embed control characters into the serial data stream between a computer and a terminal or printer. This requires the software to inspect each received character to determine if it is a control character, resulting in substantial overhead. The special character recognition hardware in the USART performs this function automatically, eliminating the software overhead. The user can designate up to 128 separate characters as being "special." Whenever a designated character is received, a maskable interrupt is generated to notify the user.
- **Synchronous/Transparent Mode**—In terminal adapter applications, it is often desirable to place data onto the network exactly as they are received from the terminal, with all framing bits included. This is referred to as a transparent channel. The USART synchronous/transparent mode provides this transparent channel by blindly receiving data in 8-bit portions. On every cycle of the receive clock, a data bit is received into the USART, including framing bits and idle bits. When 8 bits have been received, they are loaded into the FIFO. After several of these 8-bit portions of data have been received, they can be combined into a packet and transmitted over a network via the DLC. On the other side of the network, the receiving DLC processes the packet and places the field containing the data into memory. The USART on the receiving end can then re-transmit the 8-bit blocks of data, without the addition of framing bits. The result is a data stream that is identical to that received by the original USART, allowing any protocol to be transmitted over the network.

DUAL-PORT MEMORY CONTROLLER

When packet network hardware is built into a computer, such as a card installed in a PC, it is desirable to use a dedicated microprocessor to perform the communication tasks. This reduces the overhead placed on the host system's microprocessor by the communication functions. Normally, Layers 1, 2, and 3 of the ISO-OSI model will run on the communication processor, while Layers 4 and above will be handled by the host processor.

With the software functions divided between two processors, a communication mechanism is required that allows commands and data to be passed back and forth. The most straightforward vehicle is a shared memory interface, with a means for each processor to alert the other. To implement the shared memory, a section of each processor's memory must be common, and thus accessible by both processors, for example, a dual-port RAM. A means is

required to allow one processor to indicate to the other that a message (command or data) is available. A system of interprocessor interrupts provides this function. The shared memory is divided into buffer spaces and a set of mailboxes. The buffers are used to pass data to be transmitted and data that has been received back and forth between the host and the communications processor. The mailboxes are used for passing commands and status. When one processor has either a command or some status information for the other processor, it is placed in the appropriate mailbox. The sending processor then generates an interrupt to the other processor. The receiving processor responds by reading the mailbox and clearing the interrupt.

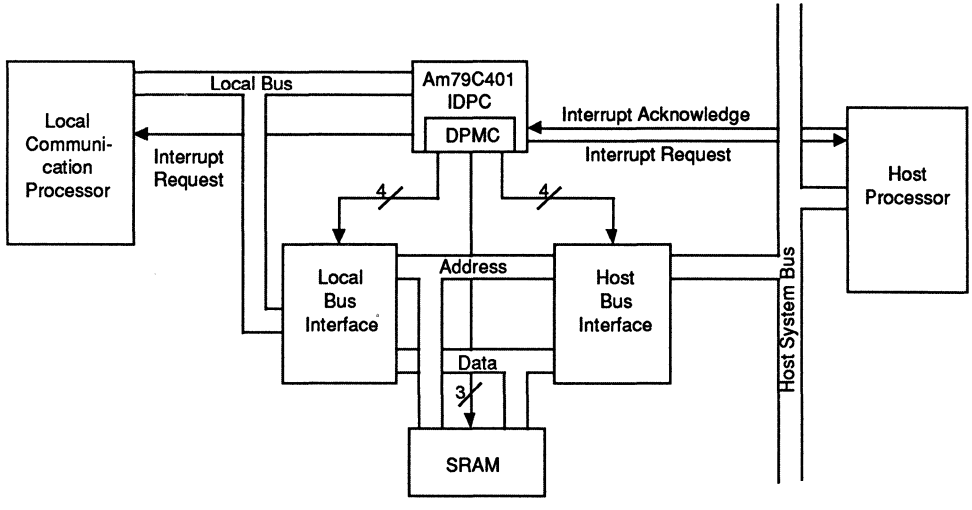
The IDPC's Dual-Port Memory Controller (DPMC) provides the support hardware to build a low-cost shared memory interface. The DPMC's bus arbitration unit allows low-cost static RAM to be used as dual-port memory. Hardware is provided for implementation of the interprocessor interrupt system.

The DPMC performs the memory bus access arbitration between the communications and the host processors. Each processor accesses the RAM as if it were the RAM's sole owner transparent to software. The DPMC generates the RAM cycle timing and outputs the appropriate chip select, output enable, and write enable signals. In the event of conflicting access requests, the DPMC holds off one of the processors for one memory cycle time by deactivating that processor's Ready signal. The interconnection between the RAM, the host's system bus, and the communication processor's address/data bus, is made via the bus interface blocks (see Figure 1-2). These blocks consist of buffers and latches that control the flow of addresses and data between the two processors' address and data buses and the RAM. The DPMC generates the control signals for the bus interface blocks.

The DPMC also provides hardware support for the interprocessor interrupt structure. The communications processor can generate an interrupt to the host processor by setting a bit in a register located in the IDPC. The setting of

5

Figure 1-2 DPMC Memory System Interconnection



this bit drives an IDPC pin (HINTOUT), which is connected to an interrupt request line to the host processor. The host processor can clear the interrupt request by pulsing a pin on the IDPC. The host can generate an interrupt request to the communications processor by pulsing another pin on the IDPC. The communications processor clears this interrupt request by writing to a register in the IDPC.

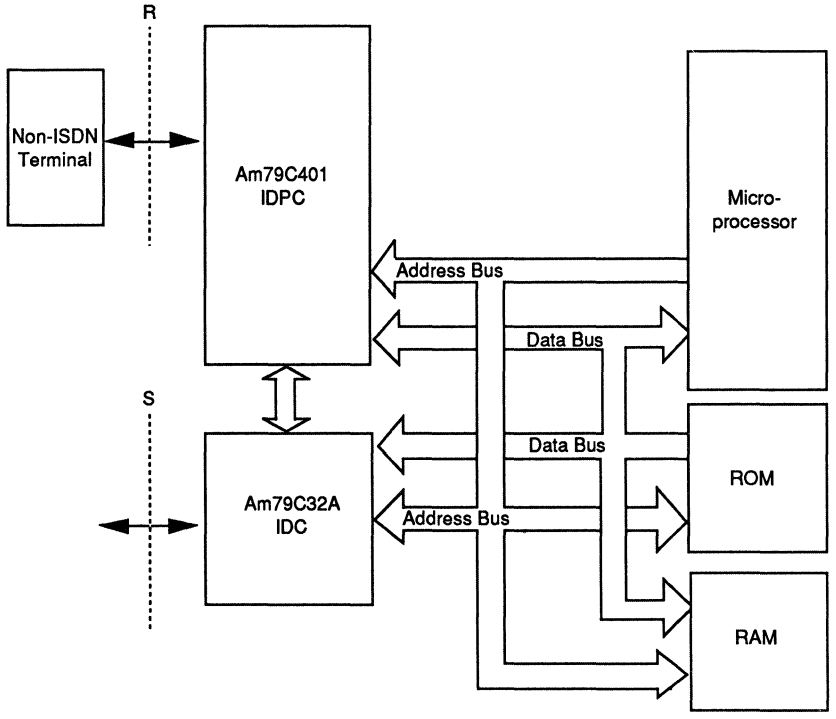
APPLICATIONS

The IDPC provides the building blocks necessary to build terminal adapters and embedded communications processors. Additionally, the IDPC can be used in applications requiring separate synchronous and asynchronous communication channels. Whether the network is SNA, X.25, ISDN, or any other bit-oriented protocol-based network, two types of devices are needed: terminal adapters that allow non-network-compatible equipment to be interfaced to the network, and embedded communications processors that integrate the network interface directly into the computer or terminal.

Terminal Adapter

The terminal adapter is a self-contained device that allows non-network-equipped terminals or computers to be connected to a network. Figure 1-3 shows the block diagram of a terminal adapter, including: a transceiver, pro-

Figure 1-3 **ISDN Term Adapter Application**

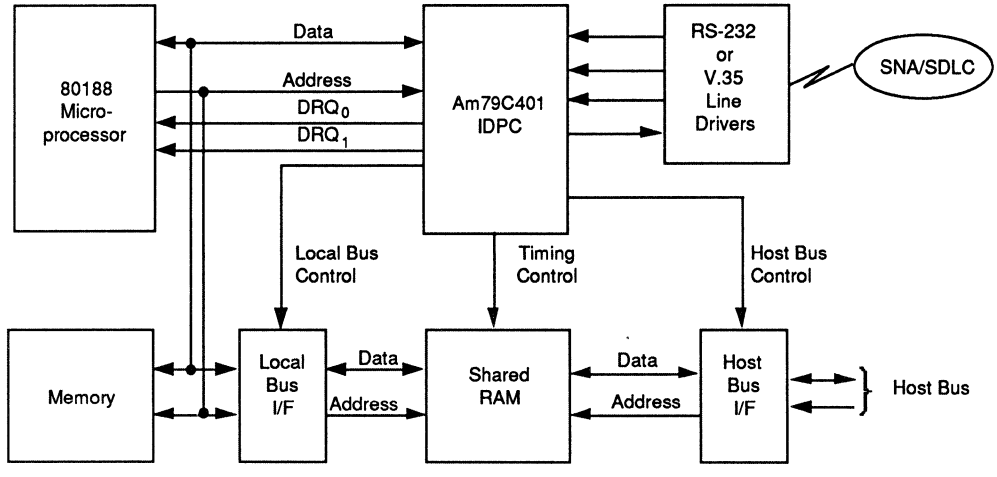


viding the physical Layer 1 connection to the network; an High-Level Data Link Control (HDLC) protocol controller; a USART, providing the terminal interface; and a microprocessor, with RAM and ROM, to process both user data and call control. The HDLC protocol controller and the USART are provided by the Am79C401 IDPC. In this example, an 80188 microprocessor provides the processing power.

Embedded Communication Processor

When the network interface is built into the computer or terminal, the communication processor is connected directly to the host's system bus. Figure 1-4 shows the block diagram of an embedded communication processor. The DLC in the Am79C401 IDPC provides HDLC packet protocol processing for the network. The Dual-Port Memory Controller supports a shared memory interface to the host processor. In this example, the network software runs on the 80188 microprocessor.

Figure 1-4 Typical SNA Application



Associated Publications

ISDN LITERATURE	ORDER #
IOM-2 Interface Reference Guide	11896A
ISDN Sales Reference Guide	11971A
World Network Solutions Brochure	10655A
ISDN Support Tools Brochure	11818B
ISDN Solutions Brochure	12035B
SLIC/SLAC LITERATURE	
Am7901A/B Data Sheet	01520D
Am7905A Data Sheet	07004A
Am7950 Data Sheet	05203C
Am7953/57/ Am79M53/M57 SLIC Family Data Sheet	11701A
Am7950 SLIC Application Notes	07030A
Analog Subscriber Line Solutions Brochure	10234A
SLIC/SLAC/DSLAC Evaluation Board User's Manual	09539B
MODEM LITERATURE	
Am79101 Data Sheet	04262D
Am7910/11 Data Sheet	09833C
MODEM Technical Manual	09560C
ARTICLE REPRINTS	
"Interface the ISDN to your PC with a Voice/Data Board"	10362A
"Performance Oriented Controller for ISDN & SNA Applications"	10618A

AMD Development Tools

ISDN	ORDER #
AmLink Layer 2 LAPD/LAPB Software	AmLINK
Am79C30A/32A Low-Level Driver Source Code	Am79LLD30/32
Am79C401 Low-Level Driver Source Code	Am79LLD401
Am79B320 ISDN Terminal Coprocessor Board Kit	Am79B320KIT
AmLink3 Layer 3 Switch Specific (5ESS) Software Kit	AmLink3SW/RBCS2/24
AmLink3 Layer 3 Switch Specific (D60/D70) Software Kit	AmLink3SW/JBCS2/
AmLink3 Layer 3 Switch Specific (DMS-100 Stimulus) S/W Kit	AmLink3SW/SBCS2/
AmLink3 Layer 3 Switch Specific (DMS-100 Functional) S/W Kit	AmLink3SW/SBCS2/
Am79C30A AmMap User's Guide	AmMap
SLIC/SLAC	
AmSLAC SLAC Coefficient Development Software	AMSLAC
SLIC/SLAC/Program Evaluation Board Kit	Am79PROGKIT
SLAC Evaluation Board	Am79SLACBD
SLIC Evaluation Board	Am79SLICBD

(Contact your local AMD Sales office for pricing and availability)

Glossary

TWO-WIRE/FOUR-WIRE

In some countries, such as Germany, the wiring inside older buildings and to the central office is two-wire. For full-duplex data communication on two wires, Germany has adopted a ping-pong type of transmission known as UP0. UP0 is easier to implement than 2B1Q and 4B3T, but can only be used for short distances.

In the United States and some other countries, the wiring inside of most houses and office buildings is at least four-wire (allows S/T interface). However, the wiring from the buildings to the central offices is two-wire (U interface). Upgrading the two-wire lines to four-wire is usually cost prohibitive, so encoding techniques have been adopted to provide full-duplex data communication over the existing two wires. Two-wire basic rate is only point-to-point and does not support multiple terminals over a single connection.

2B1Q (**2** Binary to **1** Quarternary) is a type of transmission selected for the two-wire U-interface in the United States. The dominance of the U.S. market makes 2B1Q a likely selection for most other countries.

4B3T (**4** Binary, **3** Ternary) is the type of transmission selected for the two-wire U-interface in Germany.

B-CHANNEL (Bearer Channel)

B channel is a 64 Kbps clear channel in both basic rate and primary rate contexts. Clear channel means the B channel is intended to carry any variety of user information streams across the worldwide network. The B channel is digital information (digitized voice, data, text, etc.) and does not communicate with the switching network.

BASIC RATE

The basic rate for ISDN is a data rate of 144 Kbps, comprising two full-duplex 64 Kbps B channels and a single full-duplex 16 Kbps D channel (commonly referred to as 2B+D). The B channels can be used for either voice or data, and the D channel is used for call-control signaling and packet data. Basic rate is divided into two main sections, four-wire ("S"/"T" interface) and two-wire solutions.

BIT-STUFFING

This is a type of rate adaptation that uses uses bit-stuffing (i.e., adding non-data dummy bits) to bring the data rate up to 64 Kbps. In addition, multiple channels can be time-multiplexed to bring up the data rate (i.e., several 9600 Bps channels can be placed within the 64 Kbps channel). Since many terminals have data rates much lower than 64 Kbps, this is an efficient multiplexing solution. However, bit-orienting does not support statistical multiplexing or even error-checking and re-transmission.

CALL-CONTROL SIGNALING

Of the various signaling methods, call-control signaling is the method used for call setup and teardown. Dial tone, ringing, busy signal, and connect are all examples of events controlled by call-control signals. The first priority of the D channel is to handle call-control signaling.

CCITT

(**C**onsultative **C**ommittee for **I**nternational **T**elephony and **T**elegraphy). The CCITT is part of the ITU (**I**nternational **T**elecommunications **U**nion), one of the oldest organizations within the United Nations. The CCITT recommends standards to allow world-wide telecommunications compatibility, including ISDN.

D CHANNEL (Delta Channel)

The D channel communicates to the switch network and is specified to be 16 Kbps (basic rate) or 64 Kbps (primary rate). The primary intent of the D channel is to carry signaling information for network switching. It can also be used in other applications (e.g., packet data) at a lower priority. The D channel is required for call-control signaling only a very small percentage of the time. Therefore, sending packet data over the D channel is an efficient means of transmitting data while still having the B channels for other uses.

DATA PROTOCOL

Data Prtocol is a set of rules for data communication between like machines or entities. These rules specify a set of interactions that must occur in the same formats and have the same syntax rules. Data protocols on the 64 Kbps B channel can be categorized into either full data rate or rate adaption protocols. Both support packetized data.

DMI (**D**igital **M**ultiplexed **I**nterface)

DMI is a freely licensed specification from AT&T. It is divided into four modes, three of which are commonly used in full data rate B channel transmission.

DMI mode 0 is the mode used for a 64 Kbps synchronous clear channel.

DMI mode 1 is a 56 Kbps synchronous channel in which one bit per byte of the B channel is reserved for a status field (i.e., 8 Kbps of overhead).

DMI mode 2 is the data channel protocol mode which supports standard synchronous or asynchro-

nous data transmission up to 19.2 Kbps. DMI mode 2 primarily supports terminal interface applications.

DMI mode 3 protocol provides end-to-end error-controlled data transport (for either full rate or rate adaption) using the three lower levels of the OSI model as defined in X.25 (Layer 3) and Q.921/LAPD (Layer 2). It is primarily used in computer-to-computer applications. DMI mode 3 also provides support for other network gateways using packet-oriented rate adaption (i.e., SNA, Ethernet, etc.).

FULL DATA RATE (64 Kbps data)

Since the B channel is a clear channel, the CCITT offers no specific recommendations for B-channel protocols. CCITT's X.25 packet protocol, which includes LAPB for Layer 2, will be the most common non-proprietary selection for packetized data. Proprietary network protocols such as AT&T's DMI modes 0 and 1, and IBM's SNA may also be chosen.

HDLC (High-level Data Link Control)

HDLC is an international standard for packet data communication protocols defined by the ISO (International Standards Organization). Most protocols for packet data communications (i.e., SDLC, LAPB and LAPD) are subsets of HDLC.

I.430

I.430 is the CCITT recommendation for the "S" and "T" interfaces including the electrical and functional specifications.

ISDN

(Integrated Services Digital Network). ISDN is an international telecommunications standard that provides end-to-end digital connections for networks. ISDN is the evolution from the existing analog telephone network to a new digital network, which provides for both full-duplex voice and full-duplex data communication simultaneously through a single telephone line. Existing wire and data services will be universally accessed and new, high-value services will become available.

ISO (International Standards Organization)

ISO is an international organization that sets world wide standards in telecommunications, as well as other fields.

LAPB LAYER 2 (Link Access Protocol, Balanced)

LAPB Layer 2 is the OSI Layer 2 HDLC protocol within X.25 that establishes and maintains an error-controlled point-to-point link between a terminal and the packet network.

LAPD (Link Access Protocol, D channel).

See Q.921/LAPD.

MULTIPLE LOGICAL LINKS

Multiple logical links are connections to multiple addresses on a virtual network (no single dedicated transmission path). One data terminal can communicate simultaneously with multiple addresses at the same cost as one connection. LAPD Layer 2 or X.25 Layer 3 are two protocols supporting multiple logical links.

OSI MODEL

(Open Systems Interconnection). The OSI is a hierarchical reference model for network functions in data communication developed by the International Standards Organization (ISO). Related functions are grouped together in seven layers, and each layer is a foundation to support the mechanics of the next higher level function. This divorces an upper layer from the concerns or changes in technology of its neighboring lower layer. OSI is similar to IBM's SNA protocol layers.

PACKET-SWITCHING

A key benefit of ISDN, packet switching is a data-handling technique in a communications network. It breaks apart all messages to be transmitted into units called packets. Packets are routed based on the amount of traffic on different network paths, and all packets may not follow the same route. Each packet must be transmitted with a destination address. The destination computer reassembles the data, and packets can arrive at different times over different network paths and still be correctly interpreted.

PRIMARY RATE

The primary rate is composed of multiple B channels and one 64 Kbps D channel. The U.S. uses 23 B channels (23B+D), making a primary rate of 1.544 Mbps. Europe uses 30 B channels (30B+D), for a rate of 2.048 Mbps. Both countries employ the Q.921 and Q.931 CCITT recommendations for their Layer 2 and Layer 3 (D channel signaling) protocols. Primary rate is typically used for computer-to-switch (PBX or central office) or switch-to-switch (PBX to central office) connections.

Q.921/LAPD

Q.921/LAPD is the CCITT recommendation for the ISDN user-network interface data link layer (Layer 2 of the OSI model). This is also known as LAPD (Link

Access Protocol, D channel), and is used in both point-to-point and point-to-multipoint applications. LAPD can be used on the B channels as well (V.120, DMI mode 3) and many times is preferred because only one software package needs to be supported for all channels. LAPD supports multiple logical links in Layer 2 (in contrast to X.25, where multiple links require Layer 3, and are not supported in the LAPB Layer 2).

Q.931/Q.932

Q.931/Q.932 is the CCITT recommendation for the ISDN user-network interface network layer (Layer 3 of the OSI model).This is the highest protocol layer specified for the ISDN. All higher layers are application-dependent (user-specified).

RATE ADAPTATION

Rate adaptation is needed when the message's bit rate is less than the 64 Kbps transmission rate. Two types of rate adaptation are possible: bit-stuffing and packet-oriented.

"S"/"T" REFERENCE POINT

"S" is the CCITT reference point that corresponds to the interface between the terminal equipment (and/or telephone) and the network termination. "T" is the CCITT reference point which corresponds to the interface between two parts within the network termination. The "T" interface is a functional and electrical subset of the "S" interface. In practice, they can be considered identical. Both interfaces are four-wire interfaces. This allows each "S" interface to serve up to eight separate terminals in point-to-multipoint configuration.

SNA

(Systems Network Architecture). SNA is a structure of data protocols developed by IBM, and it predates the OSI model.

STATISTICALLY MULTIPLEXED

Transmission of packets is statistically multiplexed in the sense that if several users are trying to transmit packets over a given line, each user has an equal opportunity to gain access to the line. Statistically, no one user should be able to dominate the line. Each packet is sent as soon as transmission space is available.

TRANSPARENT MODE

Transparent mode allows for voice and data communications over the B channels of the ISDN using any protocol scheme, as long as both ends use the same protocol. The transmitter and receiver must be speaking the same language. This feature makes ISDN very flexible.

"U" REFERENCE POINT

(National option)

The U reference point resides between the subscriber premises and the local exchange (a relatively long distance). It has intentionally been left undefined by CCITT. Physical differences between the wiring in different countries may cause selection of different solutions. The U.S. uses 2B1Q, whereas Germany has chosen a technique called 4B3T.

UP_0

UP_0 is a ping-pong-type transmission. It is used in Germany for a two-wire option of the four-wire "S" interface. UP_0 is not a "U" interface CCITT recommendation.

V.110/ECMA 102

V.110 is the CCITT recommendation for interfacing non-ISDN equipment to ISDN equipment using a bit-stuffing technique. ECMA (European Computer Manufacturer's Association) 102 is the ECMA's version of V.110. ECMA and V.110 are now nearly identical, and can be treated as a single document. Specialized hardware and software is required for this type of rate adaption. ITAC is designated to support V.110.

V.120/V.tad

V.120 uses LAPD, provides rate adaption via statistical multiplexing, and supports multiple logic connections. V.120 can use standard HDLC integrated circuits such as AMD's IDPC (Am79C401). V.120 requires little or no external hardware beyond the Am79C401 to accomplish rate adaptation.

X.25

This is the CCITT international packet- oriented protocol used primarily at OSI Model Layer 3. X.25 is in widespread use on data networks today. X.25 Layer 3 allows data layers to be communicated across the network simultaneously between multiple logical links. This is a key benefit of ISDN because it is the most efficient use of the switching network.

X.30

X.30 is the recommendation which addresses the terminal adaption procedure for X.21-based terminals. This is similar to V.110 except that V.110 supports V.24-based (RS232) terminals. It is the serial data control signaling that differentiates the two types of terminals.

X.31

X.31 is a procedural specification that allows X.25 to be accessed by a higher level application.

ACRONYMS

AmLink™ ISDN software package through Layer 2 (LAPB, LAPD, Q.921)

AmLink3™—ISDN software package through Layer 3 (AmLink + Q.931 + X.25)

ARCOFI™—Audio Ringing Codec Filter (Am2160)

CEME—Coordinating Entity/Management Entity

CEP—Compression Expansion Processor (Am7971)

Codec—Coder/Decoder circuit

DLC—Data Link Controller

DSC—Digital Subscriber Controller (Am79C30A)

DSL—Digital Subscriber Loop

EPIC™—Extended PCM Interface Controller (Am2055)

HSCC™—High Level Serial Communications Controller (Am82520)

HSCX™—High Level Serial Communications Controller Extended (Am82525)

IBC™—ISDN Burst Transceiver Circuit (Am2095)

IDC™—ISDN Data Controller (Am79C32A)

IDEC™—ISDN D-channel Exchange Controller (Am2075)

IDPC™—Integrated Data Protocol Controller (Am79C401)

IEC™—ISDN Echo Canceller Circuit (Am2090, 4B3T)

IEC-Q™—ISDN Echo Canceller Circuit (Am2091, 2B1Q)

IOM-2™—ISDN Oriented Modular Interface - Revision 2

ISAC-P™—ISDN Subscriber Access Controller (Am20950)

ISAC-S™—ISDN Subscriber Access Controller - S/T Interface (Am2085)

ITAC™—ISDN Terminal Adaptor Circuit (Am2110)

ITCB™—ISDN Terminal Coprocessor Board (Am79B320)

IVDT—Integrated Voice/Data Terminal

LI—Line Interface

LIU—Line Interface Unit

MAP—Main Audio Processor

ME—Management Entity

MPI—Microprocessor Interface

NLP—Network Layer Packet

NLS—Network Layer Signalling

OSC—Oscillator

PAD—Packet Assembler/Disassembler

QEPC™—Quad Exchange Power Controller (Am7938)

SBC™—ISDN S/T Bus Interface Circuit (Am2080)

SBCX™—ISDN S/T Bus Interface Circuit Extended (Am2081)

TA—Terminal Adaptor

V/D—Voice/Data

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Sales Offices

North American

ALABAMA		(205) 882-9122
ARIZONA		(602) 242-4400
CALIFORNIA,		
Culver City		(213) 645-1524
Newport Beach		(714) 752-6262
Roseville		(916) 786-6700
San Diego		(619) 560-7030
San Jose		(408) 452-0500
Woodland Hills		(818) 992-4155
CANADA, Ontario,		
Kanata		(613) 592-0060
Willowdale		(416) 224-5193
COLORADO		(303) 741-2900
CONNECTICUT		(203) 264-7800
FLORIDA,		
Clearwater		(813) 530-9971
Ft. Lauderdale		(305) 776-2001
Orlando (Casselberry)		(407) 830-8100
GEORGIA		(404) 449-7920
ILLINOIS,		
Chicago (Itasca)		(312) 773-4422
Naperville		(312) 505-9517
KANSAS		(913) 451-3115
MARYLAND		(301) 796-9310
MASSACHUSETTS		(617) 273-3970
MICHIGAN		(313) 347-1522
MINNESOTA		(612) 938-0001
NEW JERSEY,		
Cherry Hill		(609) 662-2900
Parsippany		(201) 299-0002
NEW YORK,		
Liverpool		(315) 457-5400
Poughkeepsie		(914) 471-8180
Rochester		(716) 272-9020
NORTH CAROLINA		(919) 878-8111
OHIO,		
Columbus (Westerville)		(614) 891-6455
Dayton		(513) 439-0470
OREGON		(503) 245-0080
PENNSYLVANIA		(215) 398-8006
SOUTH CAROLINA		(803) 772-6760
TEXAS,		
Austin		(512) 346-7830
Dallas		(214) 934-9099
Houston		(713) 785-9001

International

BELGIUM, Bruxelles	TEL	(02) 771-91-42
	FAX	(02) 762-37-12
	TLX	846-61028
FRANCE, Paris	TEL	(1) 49-75-10-10
	FAX	(1) 49-75-10-13
	TLX	263282F
WEST GERMANY,		
Hannover area	TEL	(0511) 736085
	FAX	(0511) 721254
	TLX	922850
München	TEL	(089) 4114-0
	FAX	(089) 406490
	TLX	523883
Stuttgart	TEL	(0711) 62 33 77
	FAX	(0711) 625187
	TLX	721882
HONG KONG,	TEL	852-5-8654525
Wanchai	FAX	852-5-8654335
	TLX	67955AMDAPHX
ITALY, Milan	TEL	(02) 3390541
		(02) 3533241
	FAX	(02) 3498000
	TLX	843-315286
JAPAN,		
Kanagawa	TEL	462-47-2911
	FAX	462-47-1729
Tokyo	TEL	(03) 345-8241
	FAX	(03) 342-5196
	TLX	J24064AMDTKOJ
Osaka	TEL	06-243-3250
	FAX	06-243-3253

International (Continued)

KOREA, Seoul	TEL	822-784-0030
	FAX	822-784-8014
LATIN AMERICA,		
Ft. Lauderdale	TEL	(305) 484-8600
	FAX	(305) 485-9736
	TLX	5109554261 AMDFTL
NORWAY, Hovik	TEL	(03) 010156
	FAX	(02) 591959
	TLX	79079HBCN
SINGAPORE	TEL	65-3481188
	FAX	65-3480161
	TLX	55650 AMDMMI
SWEDEN,		
Stockholm	TEL	(08) 733 03 50
(Sundbyberg)	FAX	(08) 733 22 85
	TLX	11602
TAIWAN	TEL	886-2-7213393
	FAX	886-2-7723422
	TLX	886-2-7122066
UNITED KINGDOM,		
Manchester area	TEL	(0925) 828008
(Warrington)	FAX	(0925) 827693
	TLX	851-628524
London area	TEL	(0483) 740440
(Woking)	FAX	(0483) 756196
	TLX	851-859103

North American Representatives

CANADA		
Burnaby, B.C.		
DAVETEK MARKETING		(604) 430-3680
Calgary, Alberta		
DAVETEK MARKETING		(403) 291-4984
Kanata, Ontario		
VITEL ELECTRONICS		(613) 592-0060
Mississauga, Ontario		
VITEL ELECTRONICS		(416) 676-9720
Lachine, Quebec		
VITEL ELECTRONICS		(514) 636-5951
IDAHO		
INTERMOUNTAIN TECH MKTG, INC		(208) 888-6071
ILLINOIS		
HEARTLAND TECH MKTG, INC		(312) 577-9222
INDIANA		
Huntington - ELECTRONIC MARKETING CONSULTANTS, INC		(317) 921-3450
Indianapolis - ELECTRONIC MARKETING CONSULTANTS, INC		(317) 921-3450
IOWA		
LORENZ SALES		(319) 377-4666
KANSAS		
Merriam – LORENZ SALES		(913) 384-6556
Wichita – LORENZ SALES		(316) 721-0500
KENTUCKY		
ELECTRONIC MARKETING CONSULTANTS, INC		(317) 921-3452
MICHIGAN		
Birmingham - MIKE RAICK ASSOCIATES		(313) 644-5040
Holland – COM-TEK SALES, INC		(616) 399-7273
Novi – COM-TEK SALES, INC		(313) 344-1409
MISSOURI		
LORENZ SALES		(314) 997-4558
NEBRASKA		
LORENZ SALES		(402) 475-4660
NEW MEXICO		
THORSON DESERT STATES		(505) 293-8555
NEW YORK		
East Syracuse – NYCOM, INC		(315) 437-8343
Woodbury – COMPONENT CONSULTANTS, INC		(516) 364-8020
OHIO		
Centerville – DOLFUSS ROOT & CO		(513) 433-6776
Columbus – DOLFUSS ROOT & CO		(614) 885-4844
Strongsville – DOLFUSS ROOT & CO		(216) 238-0300
PENNSYLVANIA		
DOLFUSS ROOT & CO		(412) 221-4420
PUERTO RICO		
COMP REP ASSOC, INC		(809) 746-6550
UTAH, R^2 MARKETING		(801) 595-0631
WASHINGTON		
ELECTRA TECHNICAL SALES		(206) 821-7442
WISCONSIN		
HEARTLAND TECH MKTG, INC		(414) 792-0920

Advanced Micro Devices, Inc. 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088, USA
Tel: (408) 732-2400 • TWX: 910-339-9280 • TELEX: 34-6306 • TOLL FREE: (800) 538-8450
APPLICATIONS HOTLINE TOLL FREE: (800) 222-9323 • (408) 749-5703

8/9/89
Printed in USA

ADVANCED MICRO DEVICES, INC.
901 Thompson Place
P.O. Box 3453
Sunnyvale,
California 94088-3453
(408) 732-2400
TWX: 910-339-9280
TELEX: 34-6306
TOLL-FREE
(800) 538-8450

APPLICATIONS HOTLINE
(800) 222-9323
(408) 749-5703

Printed in USA
12556A