HONEYWELL

DPS 6/LEVEL 6 HIGH-SPEED COMMUNICATIONS MANUAL

HARDWARE

Honeywell

DPS 6/LEVEL 6
HIGH-SPEED
COMMUNICATIONS
MANUAL

SUBJECT:

General descripton of the function and operation of the DPS 6/Level 6 High-Speed Communications Interface (HSC)).

SPECIAL INSTRUCTIONS:

Further software information can be obtained from references listed in Table 1.1.

DATE:

April, 1985

ORDER NUMBER:

YM73-00

This document is for guidance only and is subject to change without notice.

CONTENTS

1 INTRODUCTION

1.1 SCOPE AND PURPOSE OF THIS DOCUMENT

This manual describes the function and operation of the HSC600 High-Speed Communications Interface (HSC). Programming considerations are included to the extent necessary to understand the hardware and firmware descriptions. Further software information can be obtained from references. Operational theory in this document is designed to acquaint the reader with the major functional hardware and firmware elements of the HSC600 and to aid in the understanding of the more detailed information presented in the HSC Reference Manual.

1.1.1 Organisation of the Document

The manual is composed of four sections:

1. Section I - Introduction

   Describes the scope and purpose of this document; lists related reference documents; provides a summary description of the HSC within the Level 6/DPS6 system; describes specification and performance characteristics and defines abbreviations and special terminology used within the document.

2. Section II - Theory of Operation - Overview

   Defines the interfaces between the HSC and the Level 6 Megabus* network and with the communications line; describes the inter-relationship between HSC software, firmware and hardware; introduces the major hardware functional components; provides an operational overview in terms of functions performed and operational status; provides a limited description of programming influence on certain hardware elements.

   * Trademark of Honeywell Information Systems Inc.

3. Section III - Theory of Operation - Intermediate

Provides an intermediate level theory of operation of each area of the major block diagram, providing enough detail to be able to follow the Logic Diagrams contained in the HSC Reference Manual.

4. Section IV - Theory of Operation - Cycle Flow

Provides a description of the firmware cycle flows within the HSC and how these relate to specific activities providing an entry point to the detailed firmware listings contained in the HSC Reference Manual.

1.1.2 Reference Documents

See Table 1-1.

Table 1-1 Reference Documents

TITLE	DOCUMENT NUMBER	ORDER NUMBER
Series 60 Level 6 Minicomputer Handbook	-	AS22
Model 33 CPU Manual	71010670-100	EW90
Model 6/34/36 CPU Manual	71010201-200	FL36
Megabus Network Manual		
High Speed Serial Link Controller Specification	41212678	-
High Speed Serial Link LPH Internal Specification	41212494	-
EIA Specification RS-422-A	-	-
HSC600 Reference Manual	41214518	-

1.2 HSC DESCRIPTION

The HSC is a single-board high-speed communications interface for use with the Level 6 Megabus. It supports data transfer and control for one full-duplex HDLC RS-422-A communications line at rates from 9600bps to 2,000,000bps. The full-duplex line is handled as a pair of independently controlled channels, one capable of transmitting, and the other of receiving data between the line and main memory by Direct Memory Access (DMA) operations.

The HSC design is based around two similar microprocessor subsystems - one for receive and one for transmit, which handle communications aspects, and a microsequenced Megabus interface which handles DMA, programmed IO and interrupts. Programmable Array Logic (PAL*) is used for many of the control functions, allowing a compact fast design.

*Trademark of Monolithic Memories Inc.

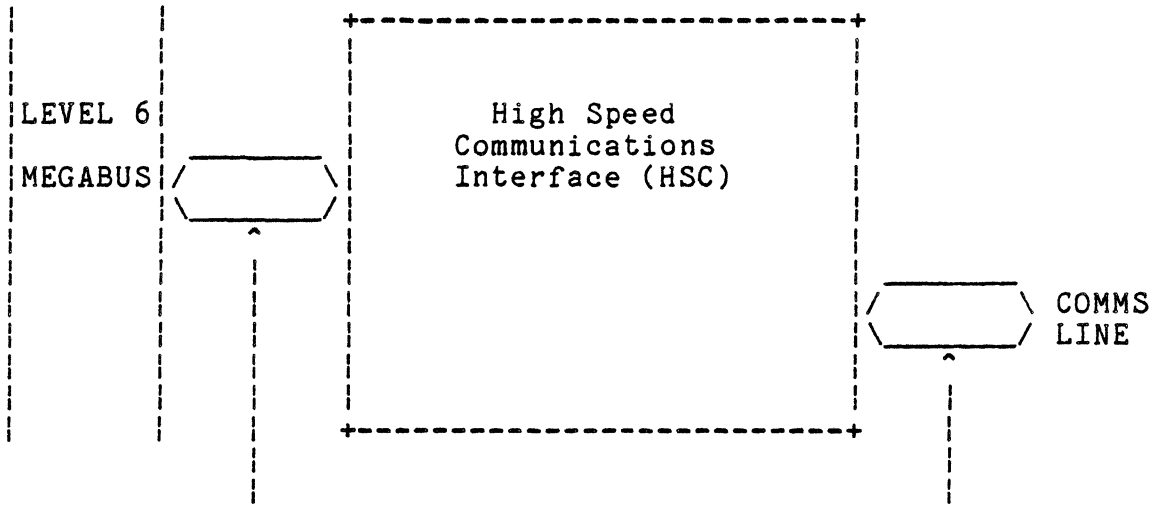


Figure 2-1
HSC/Megabus
Interface

Figure 2-2
HSC/Line
Interface

Figure 1-1 High Speed Communications Interface in Level 6 Configuration

1.2.1 Line Electrical Characteristics

The line driver and receiver characteristics are generally to RS-422-A.

1.2.2 Speeds Supported

2 Mbps*		Pulse Width Modulated (PWM)
1 Mbps	}	
500 Kbps	}	
250 Kbps	}	Separate clocks
125 Kbps	}}	and data
76.8 Kbps	}	
38.4 Kbps	}	
19.2 Kbps	}	
9.6 Kbps	}	

* Not available on YHSC600 A or B

1.2.3 Drive Distances

(from bulkhead connector)

a) 2 Mbps PWM/RS-422
30ft max.

b) 2 Mbps PWM via fibre-optic converters

Short Distance Version	100m max
Long " "	1 Km "

c) Separate Clock and data/RS-422

1 Mbps -	350ft max
500 Kbps	700ft "
250 Kbps	1400ft "
125 Kbps	3000ft "
76.8 Kbps or less	4000ft "

The above assumes cable of the following specification

100 ohm twisted pair, wire gauge 24 AWG or larger
<30 ohm per 1000ft
<20 pF/ft mutual capacitance
<40 pF/ft stray "

It also assumes that the difference in logic ground potential between the two end does not exceed ±7V peak.

1.2.4 Physical Characteristics

The HSC is a single Level 6 module, occupying one backplane slot. Figure 1-2 shows the HSC layout and dimensions. At the base of the diagram are two 50 pin Megabus connectors, while the 15 pin 'D' type line connector is located near the top edge. An 8 section DIP switch in location U14 is set to the channel number of the HSC as required in the system configuration. A hex rotary switch (U233) is accessible at the top edge of the board to set the required bit rate, see section 2.3.2 for further details.

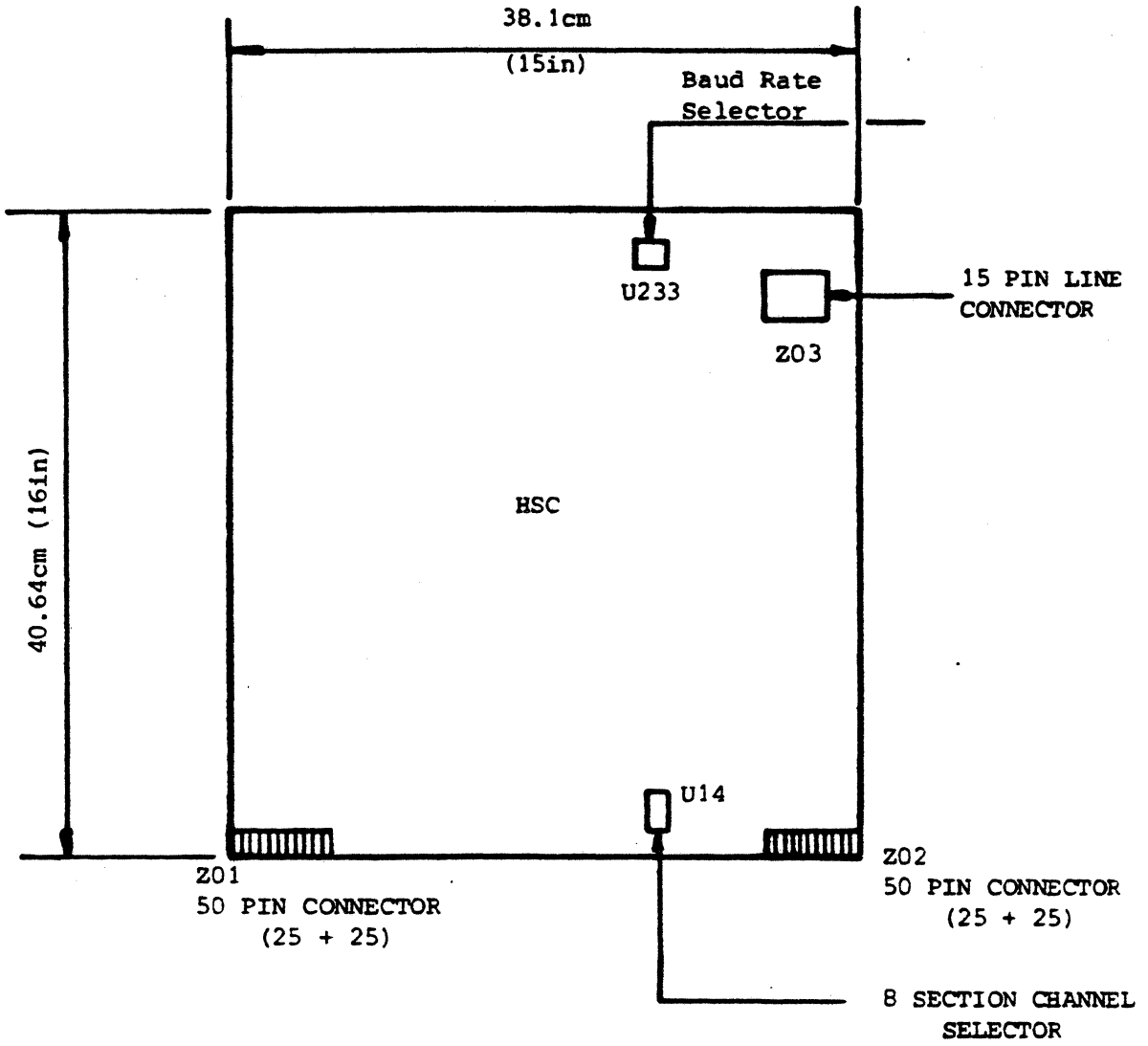


Figure 1-2 HSC Layout and Dimensions

1.2.5 Environmental Requirements

Temperature	:	0 to 40 deg C ambient
Humidity	:	5 to 95% non-condensing

1.3 ABBREVIATIONS/DEFINITIONS

ACK	Acknowledge
CHAR	Character
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DIP	Dual-In-Line Package
DMA	Direct Memory Access
EA	Effective Address
FC	Function Code
FDX	Full Duplex
FIFO	First In First Out (Buffer)
HDLC	High Speed Data Link Control Protocol
ID	Identification (Device)
I/O	Input/Output
LS	Least Significant
LSI	Large Scale Integration
MS	Most Significant
NAK	Negative Acknowledgement
ORU	Optimum Replaceable Unit
QLT	Quality Logic Test
PAL*	Programmable Array Logic
PROM	Programmable Read Only Memory
RAM	Random Access Memory
RFU	Reserved for Future Use
ROM	Read Only Memory
Rx	Receive(r)
TBS	To be Supplied
Tx	Transmit(ter)

* Trademark of Monolithic Memories Inc.

2 THEORY OF OPERATION - OVERVIEW

2.1 INTERFACE DESCRIPTION

2.1.1 HSC/Megabus Interface

The HSC attaches to the Level 6 Megabus via the interface shown in Figure 2-1. Terms and mnemonics are listed and defined in Table 2-1. This interface is the control and transfer link between the HSC and any other controller within the system (e.g. CPU, main memory etc.) and provides a path for the address, data, and control information. This interface also supplies the paths for determining priority of a request from the HSC.

2.1.2 HSC/Line Interface

The HSC Line interface is shown in Figure 2-2 and described in Table 2-2. This interface provides the clocks, data and optional supervisary signals for communication between devices. Since the bulkhead connector is different to the board connector, pinouts for this are also shown. Two sets of signal names are shown. The left hand set represents the signal names used on the HSC logic diagrams, while the right hand set are their equivalents in EIA Specification RS-449-A.

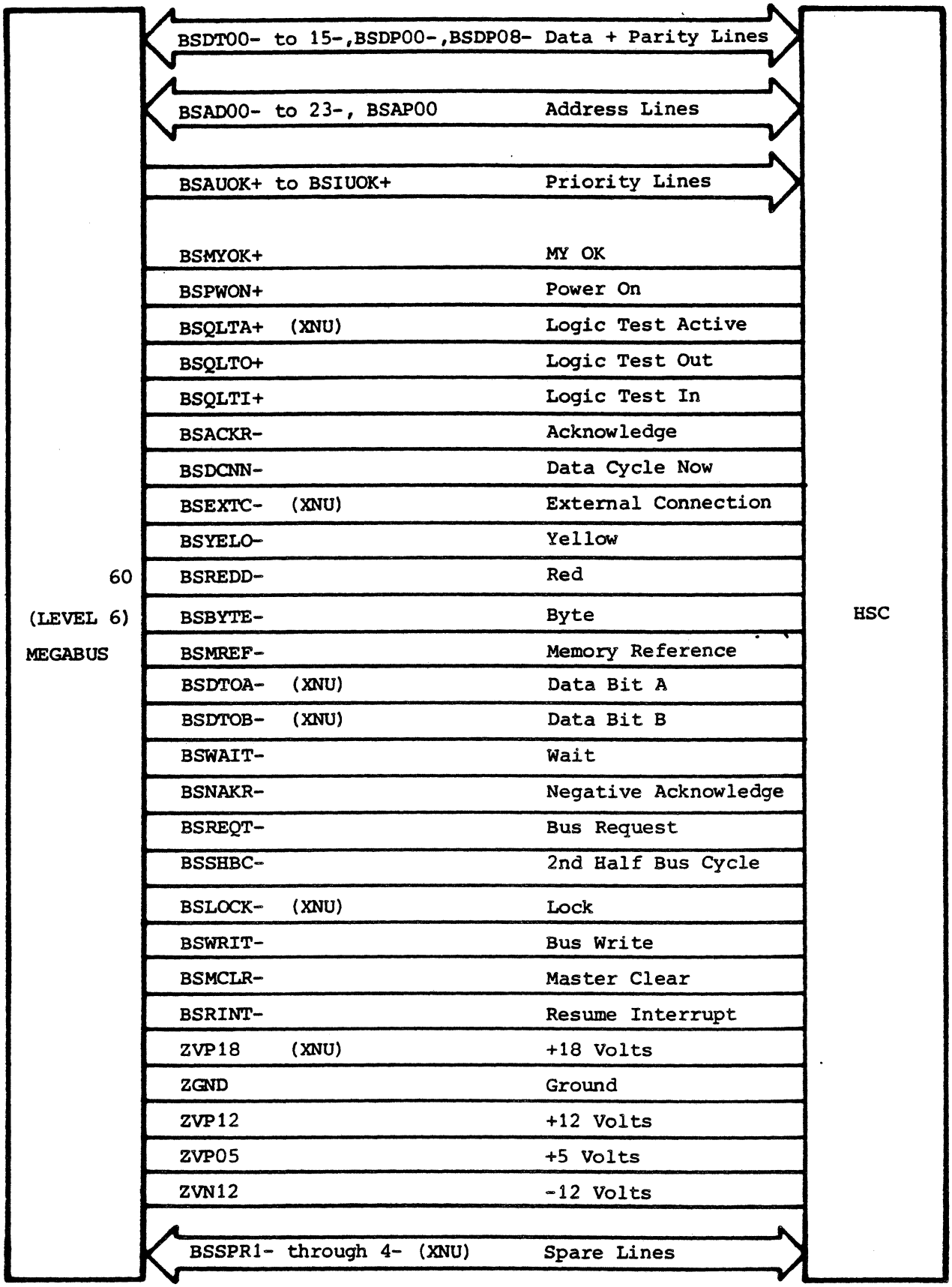


Figure 2-1 HSC/Megabus Interface

Table 2-1 Megabus/HSC Interface Signal Lines

TERM/MNEMONIC	DESCRIPTION
Data Bits 0 through 7 (BSDT00 to 07)	These eight data bit lines represent the most significant byte of data.
Data Bits 8 through 15 (BSDT08 to 15)	These eight data bit lines represent the least significant byte of data.
Data Parity-Left Byte (BSDP00)	This signal contains odd parity for data bits 0 through 7.
Data Parity-Right Byte (BSDP08)	This signal contains odd parity for data bits 8 through 15
Address Bus Bits 0 through 23 (BSDA00 to 23)	These 24 address bit lines contain an address to be used by memory or by a controller or central processor
Address Parity (BSAP00)	This signal contains odd parity for the most significant byte of the Address bus, bits 0 through 7
Priority Lines (BSAUOK through BSIUOK)	These lines are used to establish priority of the units attached to the bus
MY OK (BSMYOK)	This signal when true allows lower priority devices access to the bus
Power On (BSPWON)	This signal is true when all power supplies in the system are operating correctly
Logic Test In (BSQLTI)	This signal from the controller above indicates that one or more controllers further up the bus is running a "quality logic test" (QLT)
Logic Test Out (BSQLTO)	This signal is an OR of BSQLTI and this controllers "QLT running" signal and is used as the BSQLTI for the next lower controller
Acknowledge (BSACKR)	This signal indicates that the information on the bus has been accepted

Table 2-1 Megabus/HSC Interface Signal Lines (cont'd)

TERM/MNEMONIC	DESCRIPTION
Data Cycle Now (BSDCNN)	This signal indicates that the information on the bus is valid
Yellow (BSYELO)	This signal indicates that the accompanying transferred information is correct, but that a correction operation was performed.
Red (BSREDD)	This signal indicates that the accompanying transferred information is in error
Byte (BSBYTE)	This signal indicates that the current transfer is a byte transfer rather than a word transfer
Memory Reference (BSMREF)	This signal indicates that the address lines contain a memory address
Wait (BSWAIT)	This signal indicates that the transfer will be accepted when the bus data register is available.
Negative Acknowledge (BSNAKR)	This signal indicates that the information on the bus has been refused.
Bus Request (BSREQT)	This signal indicates that one or more units on the bus have requested a bus cycle.
Second Half Bus Cycle (BSSHBC)	This signal identifies the second bus cycle in response to a read request
Bus Write (BSWRIT)	This signal indicates that a write or interrupt operation is to be performed
Master Clear (BSMCLR)	This signal initializes the units attached to the bus
Resume Interrupt	This signal is a 200-nanosecond pulse which is issued by the central processor when it is capable of receiving interrupts again.

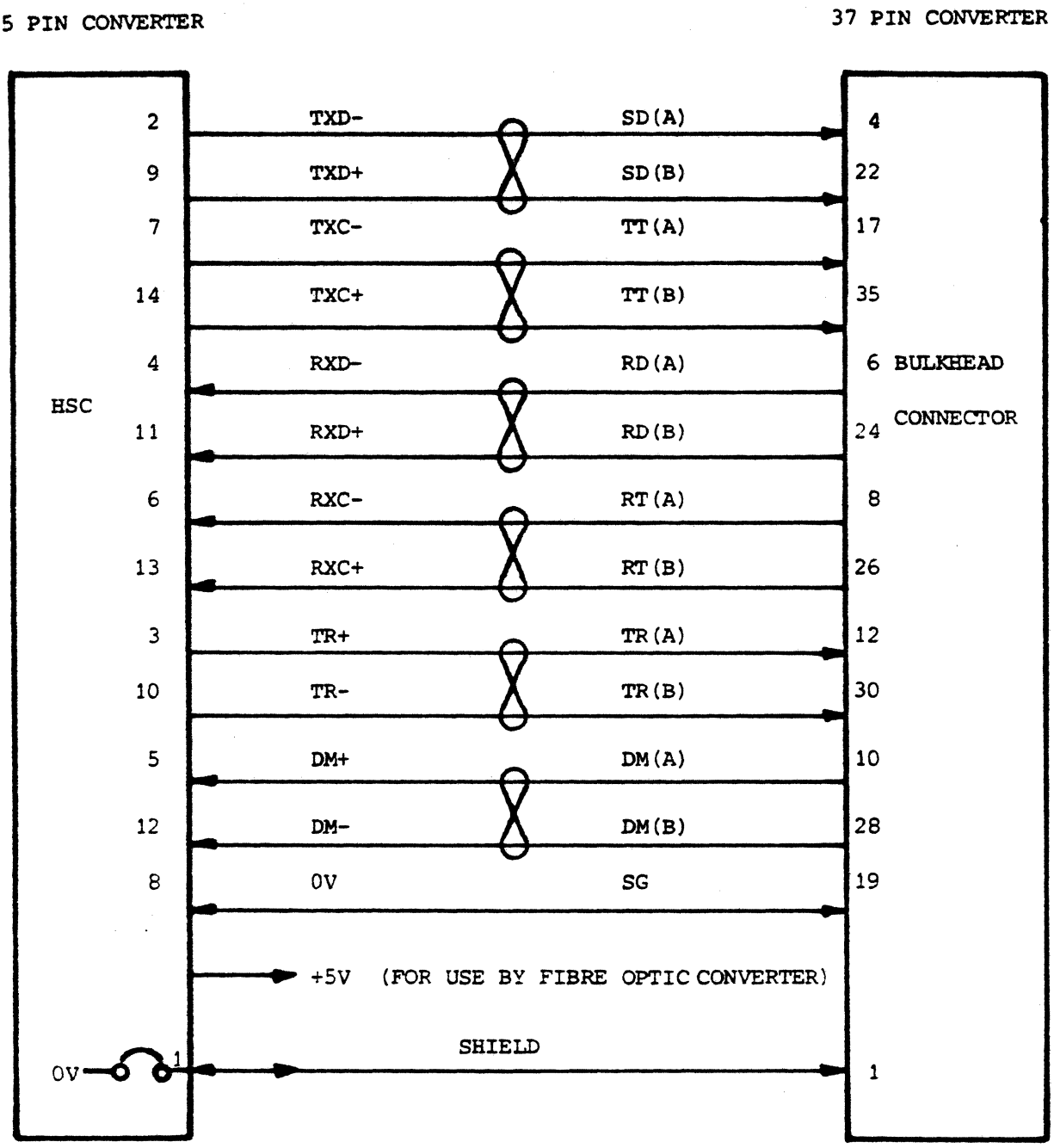


Figure 2-2 HSC/Line Interface

Table 2-2 Line Interface Signals

SIGNAL	DESCRIPTION
SD(A),SD(B)	Differential "Send Data" signals - Output from HSC
TT(A),TT(B)	Differential "Transmit Timing" signals - Clock for SD(A),SD(B) - Output from HSC
RD(A),RD(B)	Differential "Receive Data" signals - Inputs to HSC
RT(A),RT(B)	Differential "Receive Timing" signals - Clock for RD(A),RD(B) - Inputs to HSC
TR(A),TR(B)	Differential "Terminal Ready" signals - Output from HSC
DM(A),DM(B)	Differential "Data Mode" signal (equivalent to RS232 "Data Set Ready") - Input to HSC
SG	Signal Ground

2.2 FUNCTIONAL REQUIREMENTS

The basic functional components necessary for HSC operation are software, firmware and hardware as shown in Figure 2-3 and described in the following paragraphs.

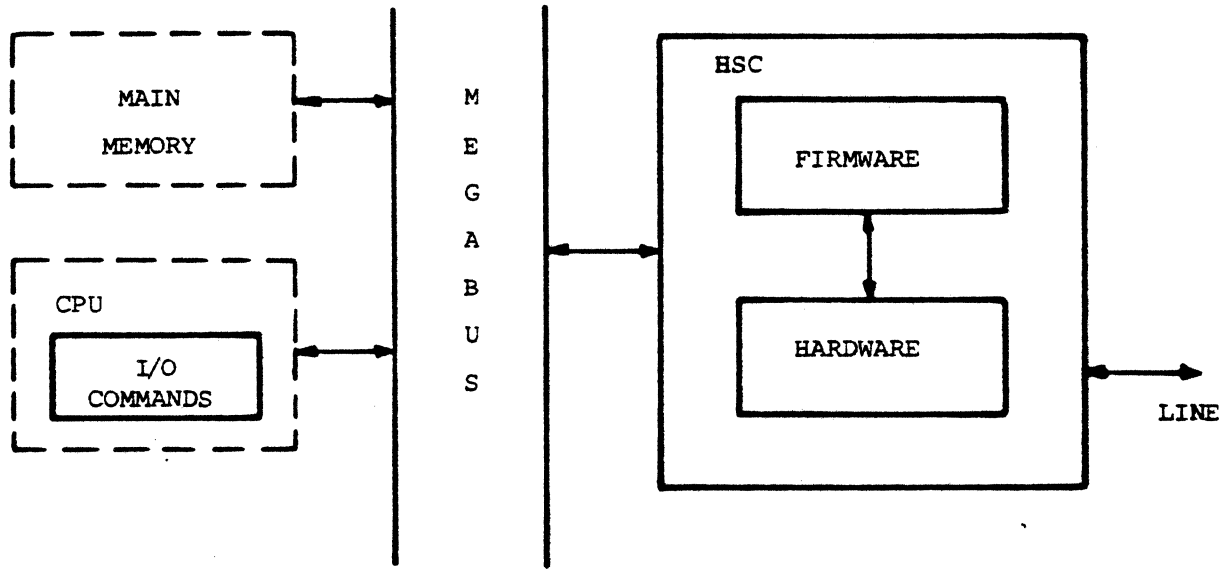


Figure 2-3 HSC Functional Components

2.2.1 Software

HSC operations are a direct result of input or output instructions from the CPU. The general format of the Megabus during I/O commands or HSC responses is shown in Figure 2-4.

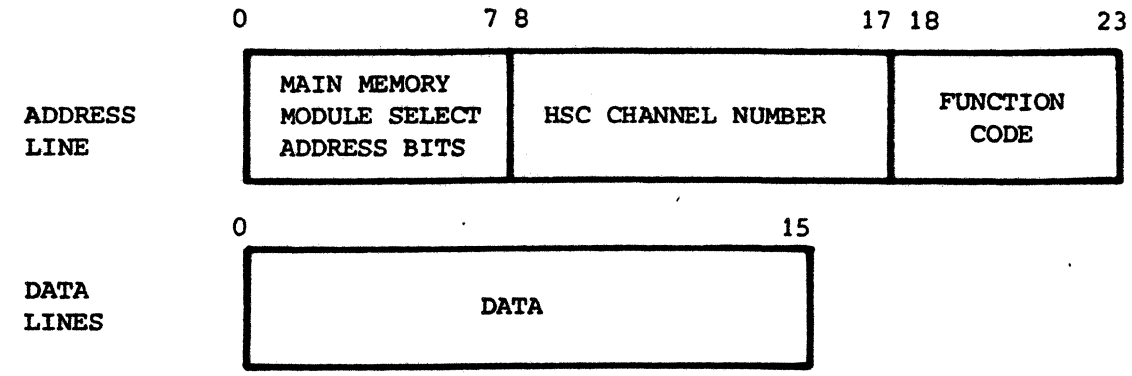
The HSC uses a pair of 10 bit channel numbers. The high order 8-bits of the channel numbers are switch selectable. The LSB is the Direction bit, and is ZERO for the receive channel and ONE for the transmit channel. Bit 9 of the channel number is ZERO.

For output instructions, the Megabus as seen by the HSC is driven such that the address lines carry the main memory module number (IOLD's only), the HSC channel number and a function code indicating the operation to be performed by the HSC. The data lines may carry either control or address information for the HSC.

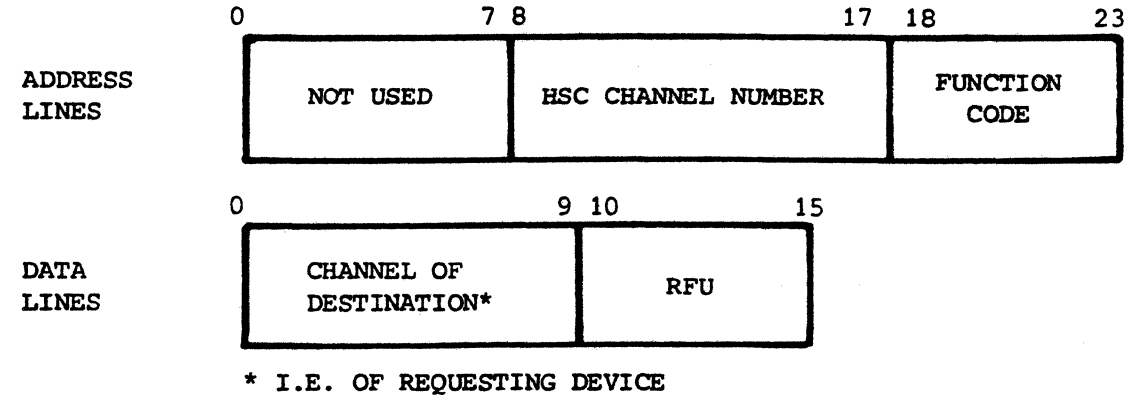
An input instruction drives the Megabus such that the address lines carry the channel number of the HSC and the function code. The data lines carry the number of the channel initiating the command.

In response to an input command, the HSC loads the Megabus address lines with the channel number of the unit to receive the requested information, which is put on the data lines.

A. HSC OUTPUT COMMANDS

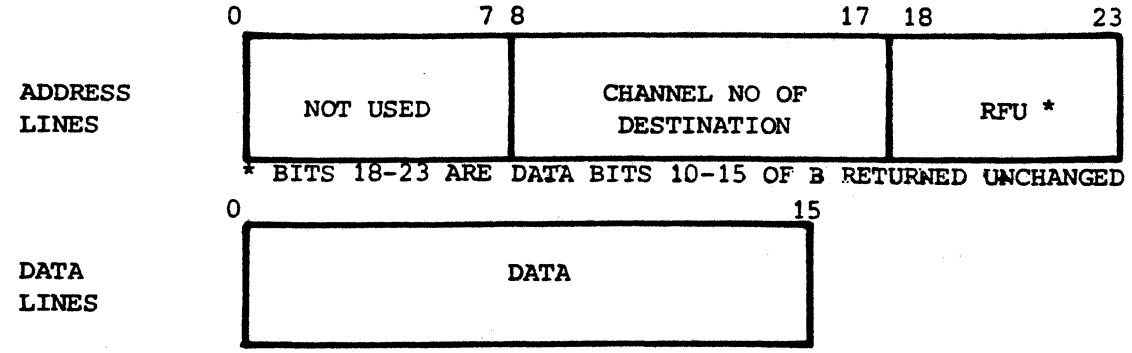


B. HSC INPUT COMMANDS



* I.E. OF REQUESTING DEVICE

C. HSC RESPONSE TO INPUT COMMANDS (CPU)



* BITS 18-23 ARE DATA BITS 10-15 OF B RETURNED UNCHANGED

D. HSC RESPONSE TO DIRECT MEMORY ACCESS COMMANDS (MAIN MEMORY)

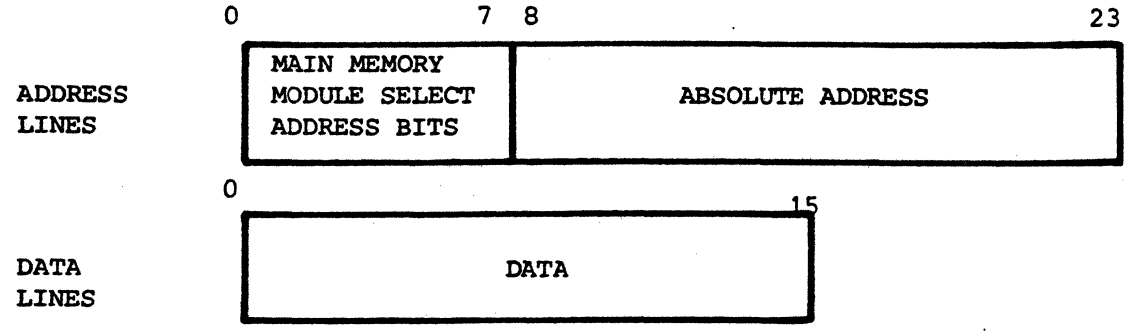


Figure 2-4 Megabus Line Configuration for HSC Instructions

The following I/O commands are provided for control of the HSC, and for diagnostic purposes. The HSC has two channel numbers - the even numbered is the receiver, while the odd numbered is transmitter.

Output Commands

1.	IO	(FC=01)	Output HSC Control
2.	IO	(FC=03)	Output Interrupt Control
3.	IO	(FC=05)	Output Channel Control
4.	IOLD	(FC=09)	Output Address and Range

Input Commands

1.	IO	(FC=02)	Input Interrupt Control
2.	IO	(FC=08)	I/P Address
3.	IO	(FC=0A)	I/P Module Address
4.	IO	(FC=0C)	I/P Range
5.	IO	(FC=18,1A)	Input Status Word
6.	IO	(FC=1C)	Input Dataset Status
7.	IO	(FC=26)	Input Device ID

These commands are described in more detail in the following paragraphs.

2.2.1.1 Output HSC Control (FC=01)

This command, which may be addressed to either channel transfers a control word to the HSC. The following control word bits are defined.

- Bit 0 - Initialize

  Clears down the controller, stopping all transfers. Runs the Quality Logic Test, returning controller to non-busy state with interrupt inhibited. Similar effects are provided by bus Master Clear.

- Bit 1 - Stop I/O

  Stops DMA transfers on both channels. If a transmission is in progress, an abort sequence will be sent. The last state of the status will be preserved. Both channels remain disabled until a start channel - See 2.2.1.3.

- Bit 2 - Terminal Ready

  If set to one sets TR on the line interface. Otherwise resets TR.

- Bit 15 - Loopback

  Sets the controller into loopback for test purposes.

The instruction will always be accepted, with an ACK response.

2.2.1.2 Output Interrupt Control (FC=03)

This command transfers a 16 bit word to the interrupt control register of the addressed channel.

The bits are defined as follows:-

- Bits 0 - 9 - CPU channel no.
- Bits 10 - 15 - Interrupt Level (if=0 disable interrupts)

2.2.1.3 Output Channel Control (FC=05)

This command outputs a control word to the addressed channel. The following bit functions are defined:-

Bit 1 - Start Channel. Enables the channel to respond to IOLD's by performing DMA transfers. Is necessary only to re-enable transfers after a STOP I/O, Stop Channel, or Power Up or Initialise.

Bit 2 - Stop Channel. Disables the channel from performing DMA transfers. Sets channel not busy aborting any transfer in progress and causing a termination interrupt if enabled. (No interrupt if no transfer in progress). IOLD's will then be accepted, but no DMA transfers will take place until a Start Channel is issued, followed by an IOLD.

2.2.1.4 Output Address and Range (FC=09)

This command transfers a 24-bit buffer address and 16 bit range to the address and range registers of the addressed channel. It also starts the read or write DMA transfer according to the direction bit of the channel address, provided the channel has been enabled by a 'Start Channel'.

On the transmit channel this instruction will receive a 'NAK' response if the device is in the process of sending a frame.

On the receive channel, this instruction will receive an 'ACK' response unless a previous IOLD is active and data is being received, when a NAK response is returned. This feature allows the receiver buffer to be changed at any time when no data is being received.

2.2.1.5 Input Interrupt Control (FC=02)

This command causes the contents of the addressed channel's Interrupt Control Register to be sent to the CPU.

2.2.1.6 Input Address (FC=08)

This command causes the addressed channel's DMA address (bits 8-23) to be sent to the CPU, for diagnostic purposes.

2.2.1.7 Input Module No. (FC=0A)

This command causes the addressed channel's DMA module number (address bits 0 to 7) to be sent to the CPU as the Most Significant byte of the input word. The L.S. byte will be zero. (Used for diagnostic purposes).

2.2.1.8 Input Range (FC=0C)

This command sends the addressed channel's range to the CPU, either to determine the residual range (usually on the receive channel only) or for diagnostic purposes.

2.2.1.9 Input Status (FC=18 or 1A)

This command causes the addressed channel's Status register contents to be sent to the CPU.

The format of the Status register is shown below:

Unused bits will be returned as zeros.

Bit No.	Meaning: Receive Channel	Transmit Channel
0	HSC Ready, QLT OK	HSC Ready, QLT OK
1	-	-
2	Data Rate Error	Data Rate Error
3	Abort Detected	-
4	CRC Error	-
5	Frame Complete	Frame Complete
6	No data/clock	-
7	Over-run	-
8	Terminal Ready	Terminal Ready
9	Data Mode	Data Mode
10	Receive Busy	Transmit Busy
11	-	-
12	-	Correctable Memory Error
13	Unavailable Resource	Unavailable Resource
14	-	Bus Parity Error
15	-	Uncorrectable Memory Error

2.2.1.10 Input Data Set Status (FC=1C)

This command inputs the top 10 bits of the status as defined above except that Bit 1 becomes 'Data Mode'. Bits 10-15 are returned as zeros.

2.2.1.11 Input Device I/D (FC=26)

This command inputs the HSC device identifier, when addressed to the receive channel. It is NAK'ed on the Tx channel.

2.2.2 Hardware

The HSC hardware is organised into 6 fundamental logic areas as shown in Figure 2-5 and described in the following paragraphs.

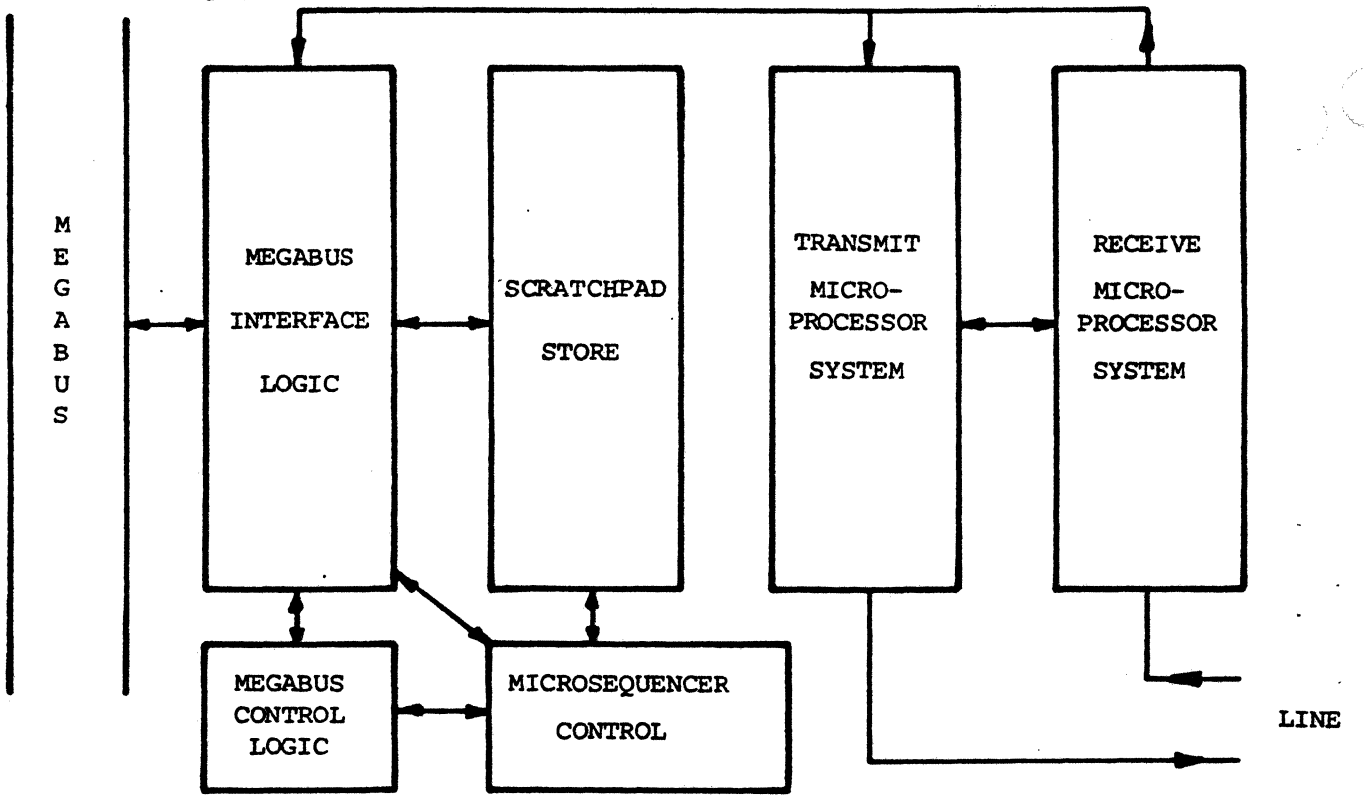


Figure 2-5 HSC Hardware Major Functional Components

2.2.2.1 Megabus Interface Logic

This consists of the transceivers which communicate with the Megabus.

2.2.2.2 Megabus Control Logic

This provides the necessary timing and contention-resolving logic to allow Megabus access.

2.2.2.3 Scratchpad Store

This provides storage for the essential registers concerned with DMA transfers (i.e. Address, Range and Interrupt Control registers for each channel).

2.2.2.4 Microsequence Control

This logic provides the sequences of control signals necessary to move data between the Megabus and the Scratchpad store, or between the megabus and the Tx and Rx microsystems. It also controls the incrementing of DMA address registers and decrementing of range. It is involved in most I/O instructions, and all DMA and interrupt operations of the HSC.

2.2.2.5 Transmit Microprocessor System

This microprocessor is responsible for all the transmit channel data manipulation except the main-memory access.

2.2.2.6 Receive Microprocessor System

This microprocessor is responsible for all the receive channel data manipulation except the main-memory access.

2.2.3 Firmware

There are three sets of firmware on the HSC, two of these are in conventional microprocessor environments, namely the Tx and Rx micros, while the third is in the microsequencer which handles most of the megabus-related activities.

The Tx firmware is responsible for taking byte-wide data provided by DMA and converting this via the components under its control into the transmitted frames, with detection of error conditions, setting up of bit rates and QLT logic tests.

The Rx firmware is responsible for assembling received frames into a byte-wide data stream to be stored via DMA, detection of error conditions and preparation of status, and QLT logic tests.

The microsequencer firmware, contained in PALs manipulates the various multiplexers, counters, scratchpad store and bus tranceivers, to provide the necessary functions for DMA read and write, I/O and interrupt service.

2.3 SUMMARY OPERATIONAL DESCRIPTION

2.3.1 Operational Overview

The function of the HSC is to transmit and receive HDLC frames over the communications line. An HDLC frame is illustrated in Fig. 2-6 and consists of 6 sections as follows:-

a) Opening flag

b) Address field (A-field)

c) Control field (C-field)

d) Information field (I-field)

e) Cyclic Redundancy Check (CRC)

f) Closing Flag

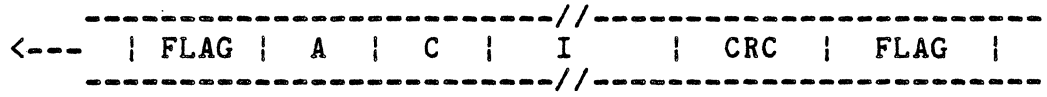


Figure 2-6 An HDLC Frame

Between frames, or during idle times the HSC will transmit continuous flag characters.

Flags are 8-bit sequences of binary 01111110 and serve as opening and closing delimiters of the frame. They are also used as time-fillers between frames.

The A-field is an 8-bit sequence containing the address of the source or destination (depending on the protocol).

The C-field is an 8-bit sequence which details the type of frame (e.g. Information or Supervisory).

The I-field can be any multiple of 8-bits comprising the information to be transferred, or may be non-existant for some types of frame.

The CRC is a 16-bit check sequence obtained by continuous division of the frame contents (A,C and I) by a standard polynominal. The polynominal used is

$$x^{16} + x^{12} + x^{5} + 1$$

The remainder constitutes the CRC.

To avoid ambiguity between the data and the flag sequence of -01111110- so causing premature frame termination at the receiver, a system of zero insertion and deletion is used. Except when flags are required, after every five consecutive ones a zero is inserted, this being removed at the receiver.

If for some reason the transmitter detects an error or does not wish to complete a frame, an abort sequence of at least seven ones is sent.

The software is responsible for providing and interpreting the A,C and I fields, while the HSC performs the framing, zero-bit insertion and deletion and the CRC generation and checking.

2.3.2 Configuration and Set-up

The channel number of the HSC is set up prior to installation by setting a DIP switch at location U14 on the board.

This switch sets the M.S. two hexadecimal digits of the address of the receiver channel. The transmit channel is then hexadecimal 0040 greater. The configuration of the switch is detailed in Table 2-3. For example channel 4A00 would be configured by setting switches 2,5 and 7 OFF, and all the rest ON.

TABLE 2-3 CHANNEL NUMBER SELECTION

	DIPSWITCH SECTION NUMBER			
FIRST DIGIT--	1	2	3	4
SECOND DIGIT--	5	6	7	8
HEXADECIMAL VALUE				
0	ON	ON	ON	ON
1	ON	ON	ON	OFF
2	ON	ON	OFF	ON
3	ON	ON	OFF	OFF
4	ON	OFF	ON	ON
5	ON	OFF	ON	OFF
6	ON	OFF	OFF	ON
7	ON	OFF	OFF	OFF
8	OFF	ON	ON	ON
9	OFF	ON	ON	OFF
A	OFF	ON	OFF	ON
B	OFF	ON	OFF	OFF
C	OFF	OFF	ON	ON
D	OFF	OFF	ON	OFF
E	OFF	OFF	OFF	ON
F	OFF	OFF	OFF	OFF

The bit-rate may by configured in-situ be means of a hexadecimal rotary switch accessible from the front of the board. The settings are as in Table 2-4.

TABLE 2-4 BIT RATE SELECTION

Hex Setting	Bit Rate	
0	9,600 bps	\
1	19,200 "	¦
2	38,400 "	¦
3	76,800 "	¦
4	125,000 "	> Clock & Data
5	250,000 "	¦
6	500,000 "	¦
7	1,000,000 "	/
8	2,000,000 "	Pulse-width Modulated*

* Not supported on YHSC600A or B

After power-up or master clear from the control panel, the HSC runs its QLT logic check and if successful is initialized, with DMA and interrupts inhibited. This state may also be reached by means of an Output Control - initialize instruction, although this is not normally necessary. Prior to performing serial data transfers, each channel requires set-up as follows:-

a) Output Channel Control - Start Channel
   - this enables DMA

b) Output Interrupt Control
   - enables interrupt on the specified level

These operations are separate for each channel.

2.3.3 Data Transfer

After a channel has been set-up as above, data transfer is initiated by issuing an IOLD instruction giving the buffer address and range (in bytes).

For the transmit channel the IOLD causes the HSC to make memory read requests to read data from the buffer. The data is passed a byte at a time to the transmit microprocessor which causes them to be output as part of a frame. When end of range is detected this is signalled to the Tx micro which generates the CRC and closes the frame. An end-of-range interrupt is sent to the CPU.

For the receive channel the IOLD causes the receive micro to search the incoming serial stream looking for an opening flag (i.e. a flag followed by a non-flag character). When this happens it starts assembling data bytes which are passed to the DMA logic. Here these are assembled into words and passed to the main memory buffer area. This process continues until either end-of-range is detected or the end of frame is sensed, when a termination interrupt is sent.

The CPU can at any time issue a Stop I/O command which will terminate DMA transfers for the HSC. Alternatively a specific Stop Channel can be issued to stop either the Tx or Rx channel. If a valid IOLD was in force at time of issue of the stop I/O or Stop Channel, a termination interrupt will be sent.

3 THEORY OF OPERATION - INTERMEDIATE

3.1 HSC MAJOR BLOCK DIAGRAM (FIGURE 3-1)

The HSC hardware is composed of the major functional areas summarised below and described in detail in the following subsections.

3.1.1 Functional Areas

3.1.1.1 Megabus Interface Logic

This logic consists of the bus tranceivers for data, address and control which form the primary interface to the Megabus.

3.1.1.2 Megabus Control Logic

This logic performs the board address recognition and function validation for incoming bus cycles, and the requesting and control of out-going bus cycles on the Megabus.

3.1.1.3 Data and Address Multiplexers

These multiplexers perform the selection of appropriate information to be put on the Megabus data and address lines for purposes of I/O, DMA and Interrupts.

3.1.1.4 Scratchpad Register File

This is a dual port register file of 8 words, used to store the DMA address, DMA range and Interrupt control words for both TX and RX channels. Its outputs are used by the address and data multiplexers and are fed to an Up-Down counter for adding or subtracting 1 or 2 for range and address calculations. It takes its input via a 2-way multiplexer either from the up-down counter, or from the Megabus Data and Address receivers via holding registers.

3.1.1.5 Microsequencer

This logic is a microprogrammed sequence controller which controls the various multiplexers, the register file and up-down counter for all DMA and interrupt operations and most I/O.

3.1.1.6 Tx and Rx FIFO Logic

The interface between the TX and RX micro-systems and the Megabus is via these two FIFO's. Word to byte conversion is carried out at this point, since the micros have 8 bit data width.

3.1.1.7 Tx Microprocessor System

This logic comprises a microprocessor, RAM and PROM memories and an HDLC communications controller which perform the serialization of transmit data from the Tx FIFO. The baud rate generaion is under control of the Tx micro, as is the PWM/Loopback logic.

3.1.1.8 Rx Microprocessor System

This is similar in design to the Tx micro but performs the reverse role, i.e assembling serial data into parallel data to be sent to the Rx FIFO. At end of frame it compiles status which is also sent via the FIFO. A pair of registers are provided to allow communication between the two microprocessors.

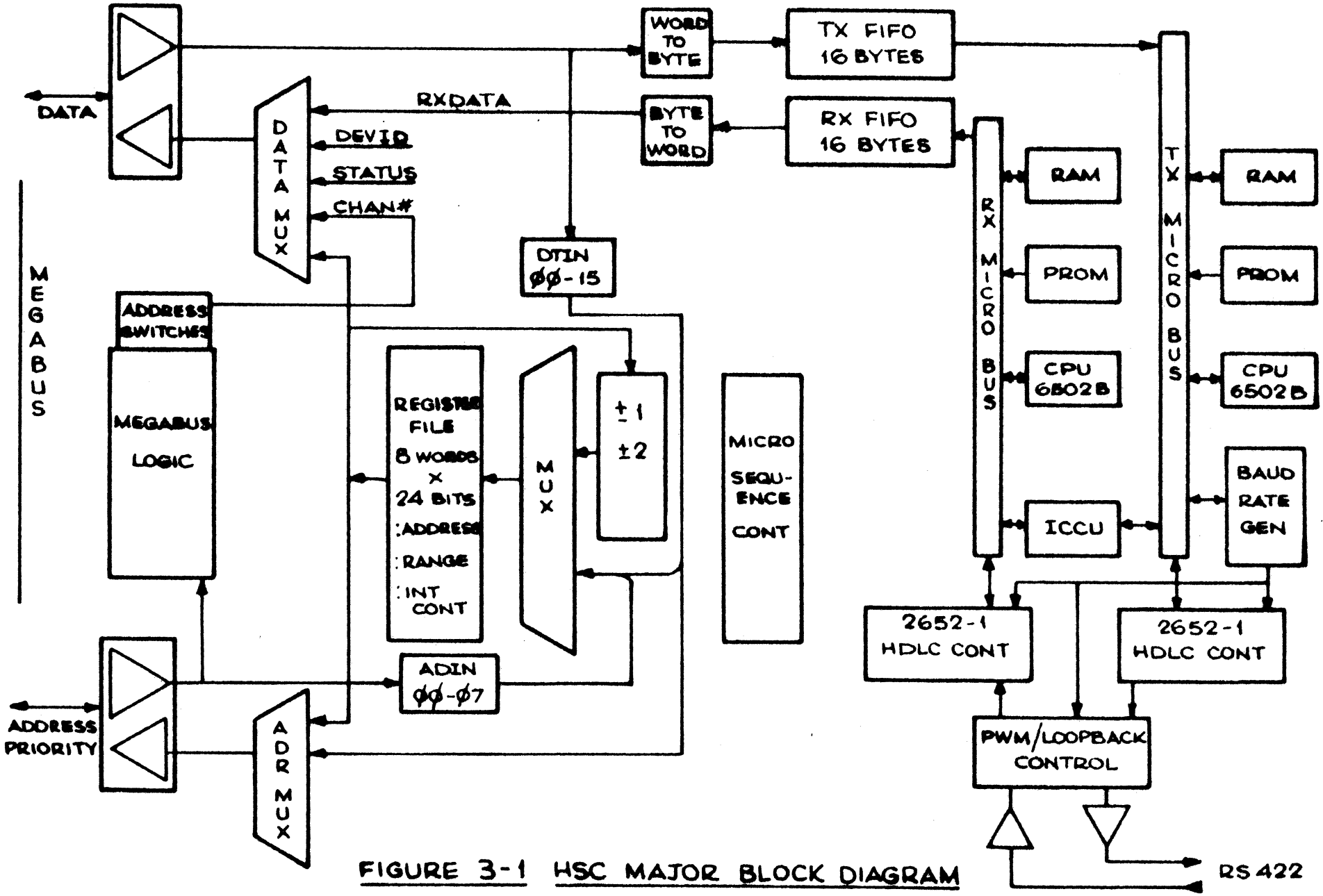


FIGURE 3-1 HSC MAJOR BLOCK DIAGRAM

3.1.2 Data Flow and Control Paths

Data flow depends on the operation being performed.

3.1.2.1 Output Instructions

The incoming instruction is checked for validity and if valid and the HSC is able to perform the instruction the HSC returns an ACK response and clocks the data and address bits into holding registers (DTINxx and ADINxx). For most output instructions (including IOLD's) the data is routed via the input multiplexer to the scratchpad register file for storage. In the case of IOLD this includes the MS 8 bits of the address bus. The storage of the data in scratchpad is performed under microsequencer control and does not form part of the Megabus cycle. The exceptions to this are "Output Control" and "Output Channel Control" where the data bits carry control functions which are decoded elsewhere for direct use.

3.1.2.2 Input Instructions

The incoming instruction is checked for validity and if valid an ACK response is generated, and the HSC loads the data and address bits from the Megabus into holding registers DTINxx and ADINxx. The microsequencer then sets up the data multiplexers to select the data source, which may be in the scratchpad (e.g. DMA range) or elsewhere (e.g. status). If the scratchpad is the source then its address bits are set up appropriately. The address mux selects DTINxx (which now contains the channel number of the device requesting the read), and a Megabus cycle is requested to return the required data.

3.1.2.3 DMA Memory Read

When a DMA memory read cycle is required the microsequencer loads the DMA range of the Tx channel into the up-down counter which is switched to Count Down mode, and reads the DMA address. It also checks for potential end-of-range and decides whether a word or byte transfer is required. If a word transfer is required the range is decremented by 2 and stored back

in the scratchpad, otherwise it is decremented by 1 and stored back. The DMA address is now loaded into the up-down counter - which is switched to Count Up mode, and a memory-reference bus cycle requested. The address multiplexer selects the scratchpad output (i.e. DMA address) while the data multiplexer selects the board address which is required by the memory controller as a return address. When the bus cycle is ACK'ed by the memory, the address is incremented by 1 or 2 in the counter and stored back in the scratchpad. At this point the microsequencer returns to its idle state or processes some other activity. When the memory responds with a Second Half Bus Cycle (SHBC) containing the data this is immediately ACK'ed by the HSC and the data loaded into a register in the Word to Byte Logic associated with the Tx FIFO. Here it is then loaded byte by byte into the FIFO. (The DTINxx register is not modified by an SHBC, since this may already be loaded for an I/O operation). From the FIFO the data is read in due course by the Tx micro and loaded into the Tx communications processor.

3.1.2.4 DMA Memory Write

This sequence operates in a similar way to the DMA read, except that the Rx range and address are used. Also at bus cycle time it is the receive data from the Rx FIFO byte to word conversion logic which is put on the data bus. In this case there is no second half bus cycle, since the memory takes both address and data on the first cycle.

3.1.2.5 Interrupts

When either the Tx or Rx channel requires to interrupt the CPU, the microsequencer accesses the appropriate interrupt control word from the scratchpad. The "CPU Number" part of the word is sent to the address multiplexer, while the "Level Number" part is sent to the data multiplexer, where it is concatenated with the HSC channel number. A bus cycle is requested to interrupt the CPU.

If an ACK response is received the interrupt request is satisfied. If NAK is received the interrupt is suspended by the HSC hardware, but re-activated and reissued when the CPU issues "RESUME". This continues as necessary until an ACK response is obtained.

3.2 MEGABUS INTERFACE AND CONTROL LOGIC

3.2.1 Megabus Signals and Operations

The megabus is composed of a collection of signals (described in table 2.1) which fall into the following groups.

(a) Address - BSAD00 through BSAD23

(b) Data - BSDT00 through BSDT15

(c) Qualifier - BSMREF, BSWRIT, BSSHBC and BSBYTE

(d) Parity - BSAD00, BSDP00, BSDP08

(e) Control - BSREQT, BSDCNN, BSACKR, BSNAKC, BSWAIT, BSMCLR

(f) Priority - BSAUOK through BSIUOK, BSMYOK

(g) Status - BSQLTI, BSQLTO, BSYELO, BSREDD, BSPWON

Table 3-1 shows a summary of the bus operations of relevance to the HSC in terms of the signals of groups a,b,c above. BSBYTE is only used when writing to memory to indicate a single byte transfer.

The various fields of the address and data are used as follows.

1) Except for memory reference operations and IOLDs, address bits 0-7 are unused, and bits 8-17 form the destination channel address. (For full-duplex devices, bit 17 is zero for the receive channel or 1 for the transmit channel). Address bits 18-23 form the "function code" with bit 23 set for a write or reset for a read operation.

TABLE 3-1 SUMMARY OF MEGABUS OPERATIONS

BSSHBC / BSWRIT / BSMREF / OPERATION	MEGABUS SIGNALS					
	ADDRESS (BSADxx)			DATA (BSDTxx)		
	0...7	8.....17	18....23	0...9	10....15	
IO-OUTPUT	010	N/U	CHANNEL No	FUNCTION (ODD)	----DATA----	
IO-INPUT REQUEST	000	N/U	CHANNEL No	FUNCTION (EVEN)	CPU No.	L
IO-INPUT RESPONSE	001	N/U	CPU No.	L	----DATA----	
IOLD	010	MEM. MOD.	CHANNEL No	FUNCTION	---MEMORY--- ADDRESS	
IO-OUTPUT INTERRUPT CONTROL	010	N/U	CHANNEL No	FUNCTION (ODD)	CPU No.	LEVEL
INTERRUPT	010	N/U	CPU No.	ZERO	CHAN No.	LEVEL
MEMORY WRITE	110	-----MEMORY ADDRESS----			----DATA----	
MEMORY READ REQ.	100	-----MEMORY ADDRESS----			CHAN No.	K
MEMORY READ RESP	001	N/U	CHANNEL No.	K	----DATA----	

2) For memory reference operations the whole of the address field is used as memory address.

3) For IOLD operations, address bits 0 to 7 carry the memory module number - i.e. most significant bits of the memory address.

4) For output instructions the data bits carry the information to be transferred.

5) For input instructions data bits 0 to 9 carry the source channel number to which the requested data must be sent, while bits 10 to 15 carry the 'L' field which is returned as "function code" and may have some significance to the originator.

6) Memory read requests, like input operations use data bits 0-9 for the source channel number, while bits 10-15 are the 'K' field which is returned to the originator in the "function code".

7) For interrupts address bits 8 to 17 carry the CPU number to which the interrupt is addressed - this is derived from the Interrupt Control Register. Data bits 0 to 9 carry the channel number of the interrupting device, while bits 10 to 15 carry the interrupt level - derived from the Interrupt Control Register.

Table 3-2 summarises the valid Function Codes for the HSC.

TABLE 3-2 HSC FUNCTION CODES

COMMAND	BSAD18-23						FUNC CODE HEX
	18	19	20	21	22	23	
Output Control	0	0	0	0	0	1	01
Output Channel Control	0	0	0	1	0	1	05
Output Interrupt Control	0	0	0	0	1	1	03
IOLD (address)	0	0	1	0	0	1	09
(range)	0	0	1	1	0	1	0D
Input Interrupt Control	0	0	0	0	1	0	02
Input Address	0	0	1	0	0	0	08
Input Module Number	0	0	1	0	1	0	0A
Input Range	0	0	1	1	0	0	0C
Input Status	0	1	1	0	0	0	18
Input Status	0	1	1	0	1	0	1A
Input Data Set Status	0	1	1	1	0	0	1C
Input Device I/D	1	0	0	1	1	0	26

3.2.2 Megabus Master Cycle Logic (Figure 3-2)

When the HSC requires a transfer via the megabus to the CPU or main memory it sets (via the microsequencer) the cycle request flip-flop CYCREQ+, having previously set up the data and address fields via their respective multiplexers.

Once CYCREQ+ has been set, the MYREQT flop is set when the Megabus becomes not busy (BSBUSY-). MYREQT drives the megabus signal BSREQT. This request prevents other devices from initiating any request, and holds the priority network BSMYOK+ low to lock out lower priority devices.

With MYREQT set, and with the completion of any previous bus cycle MYDCNN+ will be set via SETDCN- provided the HSC has the highest priority of any boards requesting. This is indicated by BSAUOK+ through BSIUOK+ being high.

Once MYDCNN+ is set, the HSC enables its data, address and qualifier drivers, and awaits a response from the addressed unit.

The response (ACK, NAK or WAIT) resets MYDCNN, and except for a WAIT response, resets MYREQT via MYREQR+. A WAIT response leaves the request flop set, so that retries are performed until either an ACK or NAK is received.

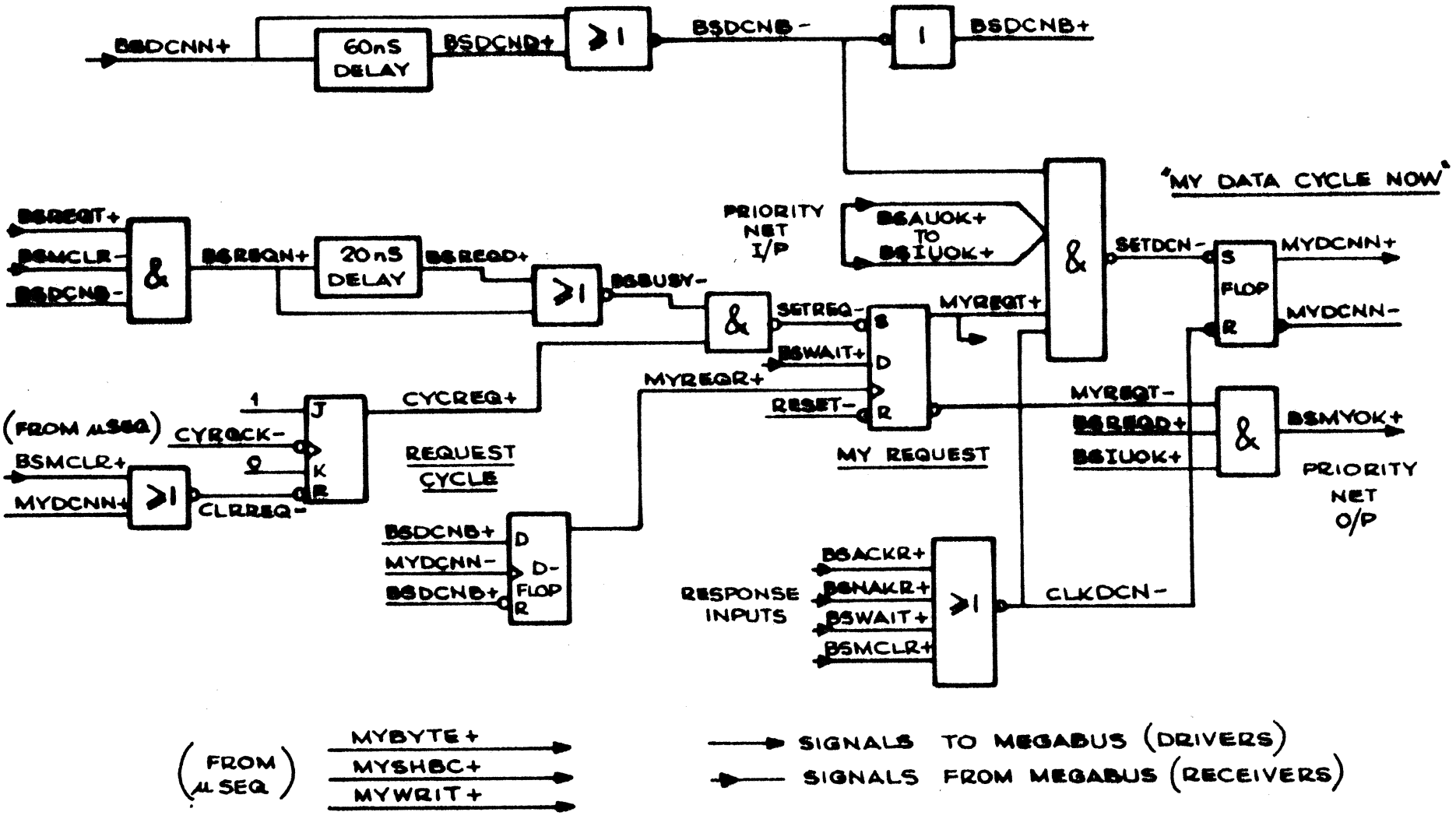


FIGURE 3-2 MASTER CYCLE LOGIC

3.2.3 Slave Response Logic (Figure 3-3a)

This logic generates the ACK, NAK or WAIT reponse to cycles addressed to the HSC.

The bus address signals BSAD08 through BSAD15 are compared to the board address by an 8-bit comparator. The comparator is disabled when BSMREF is true, to avoid responding to memory access cycles. If BSMREF is false and the address compares MYCHAN- goes low, enabling the four AND gates which drive the inputs of a quad D-type register. These gates are controlled by ENBACK, ENWAIT from the function validation logic, and the second-half bus-cycle signal BSSHBC. The register is clocked by BSDCND, a delayed version of BSDCNN (to allow settling time for the logic). On the rising edge of BSDCND one of the flip-flops MYACKR, MYWAIT, MYNAKR or MCYRCV will be set according to Table 3-3. The states not listed are invalid and are not produced by the function validation logic. MYACKR and MCYRCV are OR'ed together (MYACKN) to drive the BSACKR driver. MYACKR is produced in response to I/O instructions, while MCYRCV is in response to memory second-half bus-cycles and is used to strobe memory data into the HSC. MYACKR is used to clock the data and address information from I/O cycles into holding registers (ADINxx and DTINxx) for subsequent use. Except for output control (FC=01) and channel control (FC=05) it sets flip flop IOCACK. This flop requests the microsequencer (section 3.6) to complete the I/O function, and is reset on completion thereof.

TABLE 3-3 SLAVE RESPONSE AND VALIDATION LOGIC

ENBACK	ENWAIT	BSSHBC	RESPONSE SET
0	0	X	MYNAKR
1	0	0	MYACKR
0	1	0	MYWAIT
1	0	1	MCYRCV

X = don't care

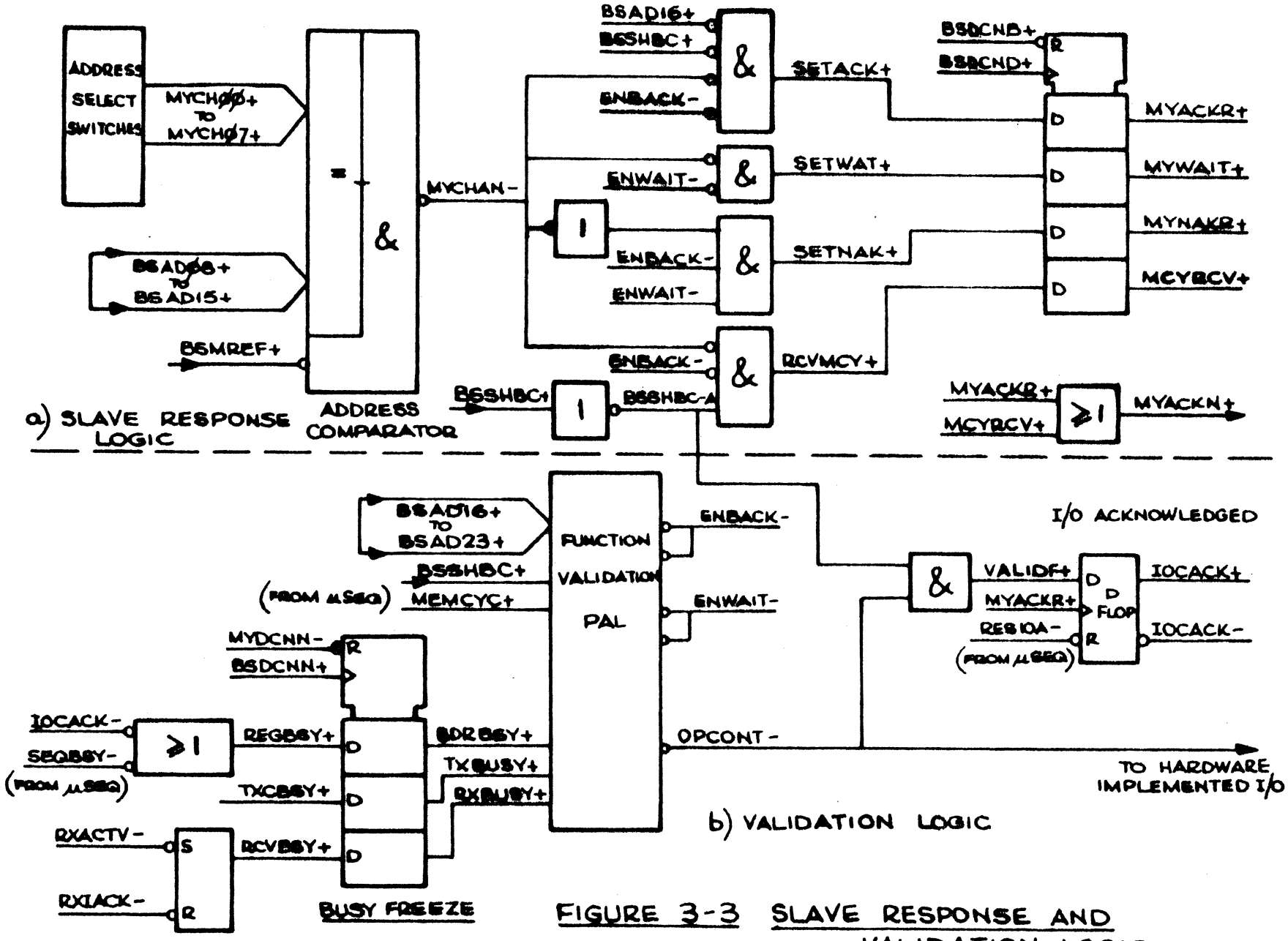


FIGURE 3-3 SLAVE RESPONSE AND VALIDATION LOGIC

3.2.4 Function Validation and BUSY Logic

(Figure 3-3b)

This logic analyses the function code of incoming bus cycles and the BUSY state of the HSC, and decides on the appropriate response. At the beginning of a bus-cycle a register is clocked with BSDCNN to freeze three separate busy signals:-

a) REGBSY+ - indicating that the micro sequencer is temporarily busy. This is an OR of SEQBSY from the microsequencer and IOCACK, a flop indicating that an I/O request is pending on the micro-sequencer.

b) TXCBSY+ - Transmit Channel Busy - the TX channel is processing an IOLD.

c) RCVBSY+ - Receiver Busy - communications data is currently being received, i.e. an HDLC frame is actually in the process of reception.

This is derived from the RXACTV+ signal from the RX communications controller via a latch. The latch is reset only when the RX termination interrupt has been ACKnowledged. (RXIACK).

The corresponding outputs of the BUSY- status register, BDRBSY, TXBUSY, RXBUSY are inputs to the functon validator. The other inputs are BSSHBC, MEMCYC (a flip-flop indicating that a memory second-half bus-cycle is expected), and the function-code bits of the address bus, - BSAD16 to BSAD23.

The function validator has three outputs, ENBACK, ENWAIT and OPCONT. The first two control the bus response, while OPCONT is made true only for "Output Control" and "Output Channel Control" function codes, which are not to be processed by the micro-sequence logic. The function validator is implemented by a Programmable Array Logic (PAL*) device. Table 3-4 defines its action.

TABLE 3-4 FUNCTION VALIDATOR

INPUTS						OUTPUTS		
FUNCTION Code (BSAD17 -BSAD23)	BSSHBC	MEMCYC	BDRBSY	TXBUSY	RXBUSY	ENBACK	ENWAIT	OPCONT
X	1	0	X	X	X	0	0	0
X	1	1	X	X	X	1	0	0
01 or 05	0	X	X	X	X	1	0	1
02,08,0A 0C,18,1A	0	X	0	X	X	1	0	0
1C,26,03	0	X	1	X	X	0	1	0
09,0D	0	X	X	X	1	0	0	0
	0	X	1	X	0	0	1	0
(RXIOLD)	0	X	0	X	0	1	0	0
89,8D	0	X	X	1	X	0	0	0
	0	X	1	0	X	0	1	0
(TXIOLD)	0	X	0	0	X	1	0	0
OTHER	0	X	X	X	X	0	0	0

*PAL is a trademark of Monolithic Memories Inc.

In summary,

1) Second-half bus-cycles, provided they are expected (MEMCYC true) will always receive an ACK response, otherwise NAK.

2) Output Control (FC=01) or Channel Control (FC=05) will always receive an ACK response.

3) All valid input instructions to either channel will receive an ACK if BDRBSY is false, or a WAIT if BDRBSY is true.

4) Output Interrupt Control (FC=03) will receive an ACK if BDRBSY is false or WAIT if it is true.

5) An IOLD (FC=09,0D) will receive NAK if that channel is BUSY (RX- or TXBUSY). If the channel is not BUSY then an ACK or WAIT is sent depending on BDRBSY.

3.2.5 Reset and Output Control Logic (Fig.3-4)

3.2.5.1 RESET

At power up (BSPWPN-high) or as a result of CLEAR button depression on the control panel (BSMCLR+ high), BCLEAR- is driven low. BCLEAR- low sets the CLEAR flop (CLEARF- low) and resets the RESET TIMER (4 bit counter) driving RESET- low. RESET- is supplied to the TX and RX microsystems and among other things causes the QLT check to be run.

RESET- low holds RESTOP- high, enabling counting, so that when power is established, and the CLEAR button is released, (BCLEAR- high) the counter counts SCLOCK+ pulses. SCLOCK+ is an 8MHz clock derived from a master 24MHz oscillator in the microprocessor subsystem. After 8 counts (1 microsec) RESET- switches to the high state, and since CLEARF- is now also high (RESET- low resets the CLEAR F/F) RESTOP- goes low inhibiting further counts.

The reset sequence can also be invoked via the 'Output Control' instruction (see following paragraphs). In this case INITLZ- goes low and is clocked into the CLEAR F/F by OPCPLS- causing CLEARF- to fall which enables the counter via RESTOP-. The counter will normally be holding state 8 at this time. It therefore counts through to 15 and then zero, dropping RESET- (which resets CLEAR F/F, but holds RESTOP- high). Counting continues through to state 8 when RESET- goes high again, and the counting is disabled.

3.2.5.2 Output Controls

"Output Control" (FC=01) and "Output Channel Control" (FC=05) instructions addressed to the HSC cause MYOPCT+ to go true which then sets the O/P CONTROL PULSE F/F on the rising edge of BSDCND+. The flop is reset at the end of the Megabus cycle by BSDCNB+ going false. Thus OPCPLS- is pulsed low for at least 60ns. These two instructions are always ACK'ed by the bus control logic, the data and address fields being clocked into holding registers ADINxx and DTINxx. Relevant address and data bits are decoded by a register PAL device, setting STOPRX, STOPTX, LPBACK and SETDTR, and CLEARF (via INITLZ-) on the back edge of OPCPLS-, as follows.

```
FC=01:-          ADIN21 = 0
                 CLEARF = DTIN00     (Initialise)
                 LPBACK = DTIN15
                 SETDTR = DTIN02
                 STOPRX = 1 if DTIN00 = 1  (STOP)
                 STOPTX = 1 if DTIN00 = 1  (I/O )

FC=05, RX channel:-      ADIN21 = 1, ADIN17 = 0

                 STOPRX = 1 if DTIN02 = 1 (Stop Channel)
                 STOPRX = 0 if DTIN01 = 1 (Start Channel)

FC=05, TX Channel:-      ADIN21=1, ADIN17=1

                 STOPTX = 1 if DTIN02 = 1 (Stop Channel)
                 STOPTX = 0 if DTIN01 = 1 (Start Channel)
```

LPBACK when true causes an internal loopback of the transmit serial data output to the receive serial data input for test purposes. SETDTR is an input to the TX micro, which is output as Terminal Ready (TR) on the line interface. STOPRX and STOPTX are disabling signals for the RX and TX channels respectively. They are true after a RESET, preventing any DMA operation and facilitating register testing by the Test and Verification program. Channels must be 'started' by a start channel operation before normal operation can proceed.

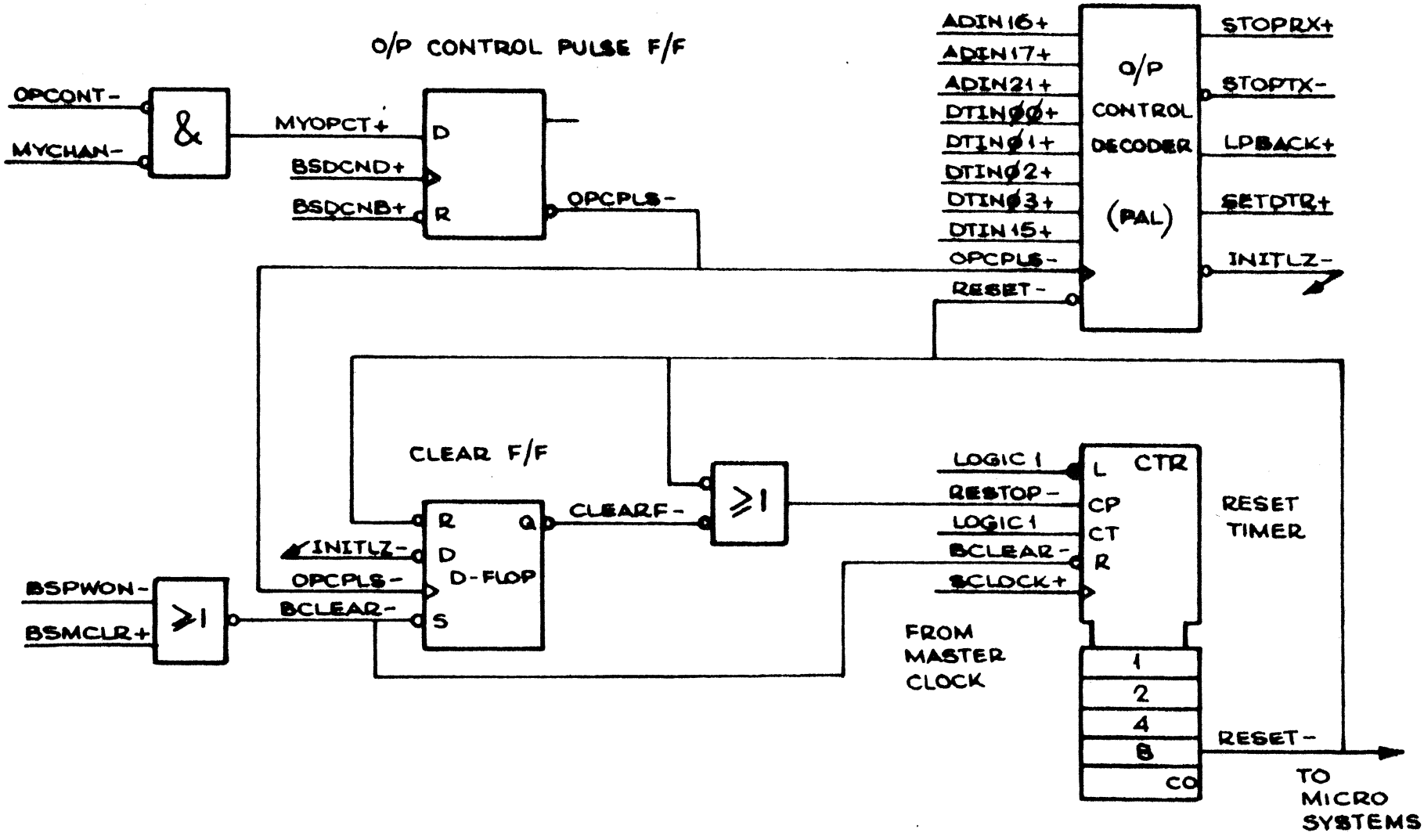


FIGURE 3-4 RESET AND OUTPUT CONTROL LOGIC

3.3 DATA AND ADDRESS MULTIPLEXERS

These multiplexers provide the interface between the Megabus drivers and the data and address sources on the board.

3.3.1 Data Multiplexer (Figure 3-5)

The data multiplexer comprises a main 4 input multiplexer for all sixteen data bits, with an auxiliary 2 input multiplexer for the MS 10 bits. The auxiliary and MS 10 bits of the main multiplexer are tri-state devices (74LS257 and 74LS253) while the L-S 6 bits of the main multiplexer are two-state devices (74LS153) to allow logic zero outputs to be forced.

The MUXes are controlled by input select bits DMSEL1+ and DMSEL2+, multiplexer enable signal AUXMUX for the MS 10 bits and LOWMUX for the LS 6 bits. All these controls are outputs from the micro-sequencer. The signal sources selected and their use are shown in Table 3-5.

TABLE 3-5 DATA MUX SELECTION

Control/Select States				Source	
LOWMUX-	AUXMUX-	DMSELI+	DMSEL2+	Selected	Operation
0	1	0	0	RXDTxx	DMA write to memory
0	1	1	0	SPOD08-23	I/O read of Address, Range or Interrupt Control
0	1	0	1	STATxx	Status read
0	1	1	1	DEVIxx	Board I/D read
1	0	X	1	SPOD00-07 zeros	Module Number Read
1	0	X	0	MYCH00-07 0,SPRAD2, zeros	DMA read from memory (OWN CHANNEL NUMBER)

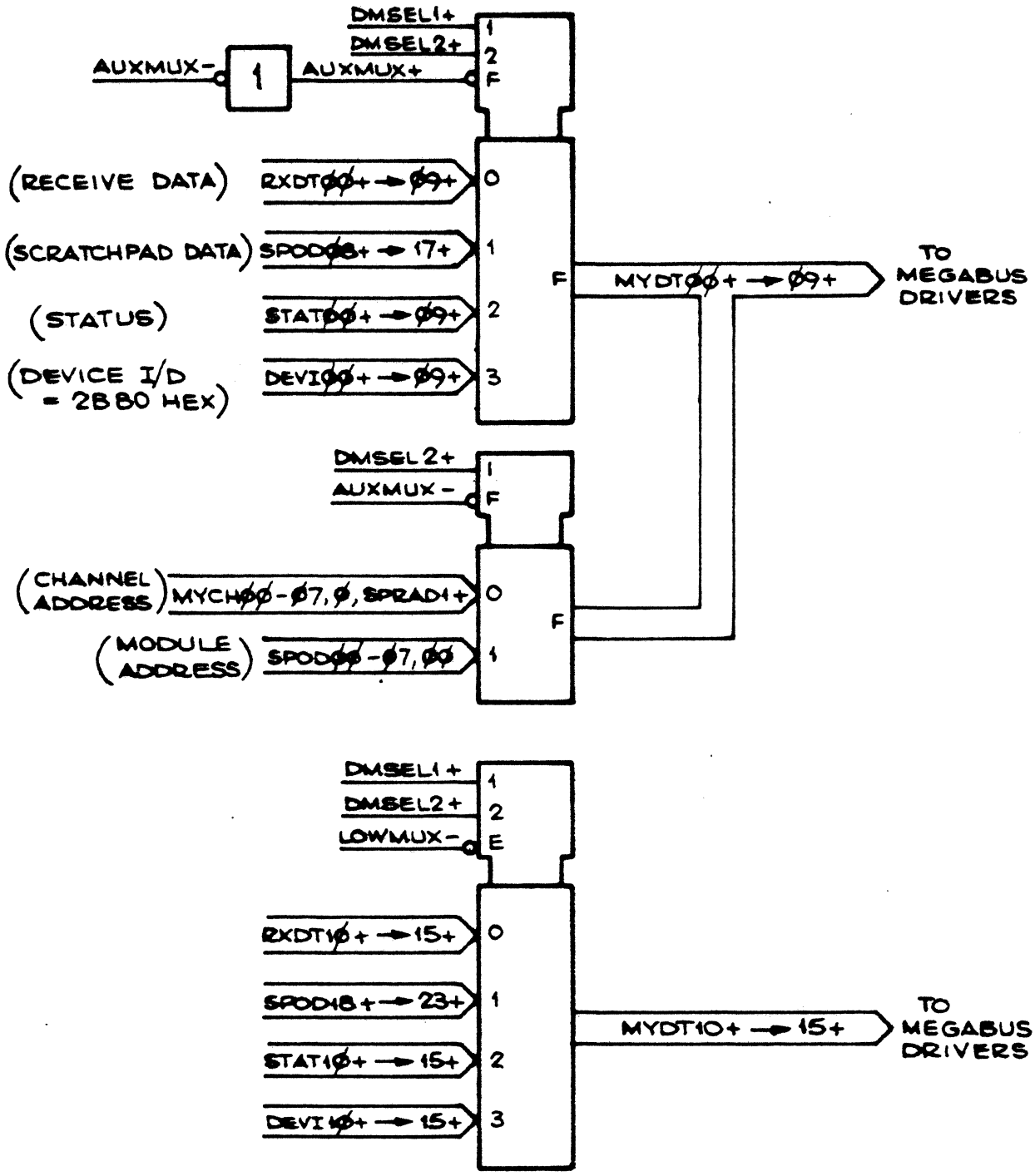


FIGURE 3-5 DATA MULTIPLEXER

The status information is obtained from a separate 2-input MUX which selects Rx or Tx status under control of ADIN17+. Since some status bits may be asynchronous, their multiplexers are equipped with storage (74LS399) which is loaded by IOSEQU+ to "freeze" the status prior to requesting the Megabus cycle for reply.

3.3.2 Address Multiplexer (Figure 3-6)

The address multiplexer is a 24-bit, two input MUX, logically divided into two parts. The MS part (MYAD00+ to MYAD17+) is normally permanently enabled*, while MYAD18+ through MYAD23+ are enabled by ADLOEN-. Input selection is performed by ADRSEL+. Table 3-6 shows the signal sources and their use.

TABLE 3-6 ADDRESS MUX SELECTION

Control Input ADRSEL+	ADLOEN-	Source Selected	Operation
0	0	SPOD 00-23	DMA memory read or write (SPODxx is memory address)
0	1	SPOD 00-15, 00000000	Interrupt (CPU No. from SPODxx)
1	0	00000000, DTINxx	I/O Read Second Half Bus Cycle (DTIN contains CPU No. + L field)

ADRSEL and ADLOEN are outputs from the microsequencer.

*Bits 8 to 15 are tristate devices, parallelled with a tristate buffer whose inputs are MYCH00+ to MYCH07+. When QLTADR+ is true the buffer is selected instead of the MUX, allowing the HSC to effectively address itself on the Megabus. This feature is included for future enhancements to the QLT system.

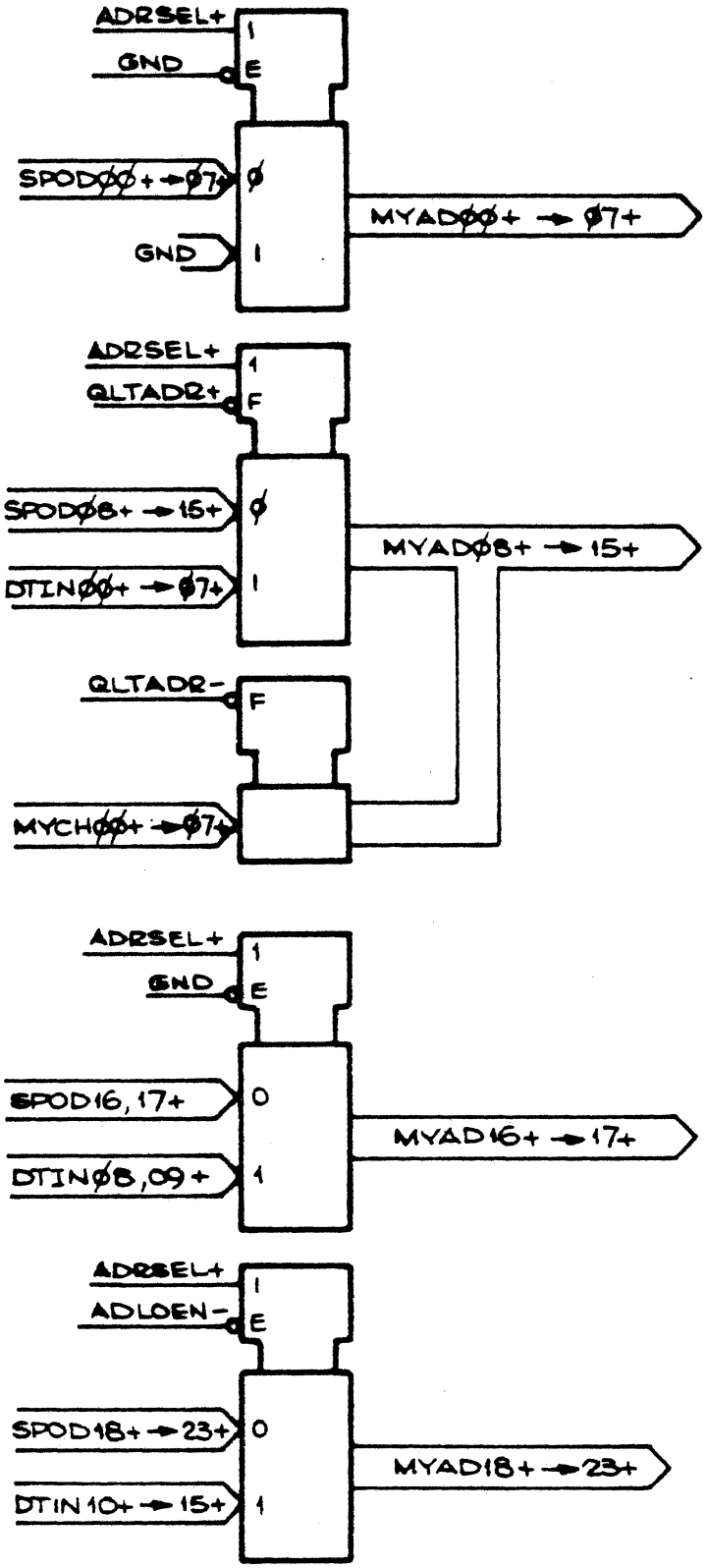


FIGURE 3-6 ADDRESS MULTIPLEXERS

3.4 SCRATCHPAD REGISTER FILE (FIGURE 3-7)

The scratch pad register file is an 8-word bank of registers composed of 74LS670 4x4 register arrays. These devices have separate read and write addressing. The first 4 registers are 24 bits long (6 x 74LS670) and provide storage for the RX and TX DMA address, 2 registers being spare. The second four registers are only 16 bits long (4x74LS670) and are used to store the RX and TX DMA ranges, and their interrupt control words. The file is addressed, written and read under control of the microsequencer. SPRAD1 and 2 are the read address select bits, while SPWAD1 and 2 are the write address select bits. SPREN1 enables the 24-bit long registers to be read, while SPREN2* enables the 16-bit register to be read. Writing to a selected 24 bit register is performed by SPWEN1- pulsing low; similarly SPWEN2- writes to the 16-bit registers. The register addressing and use is shown in Table 3.7.

TABLE 3-7 REGISTER ADDRESSING

SPXAD1+	SPXAD2+	SPXEN1-	SPXEN2-	REGISTER
	1	0	0	0	RX DMA ADDRESS (24 bit)
	1	0	0	1	TX DMA ADDRESS (24 bit)
0 = LOW	1	0	1	0	SPARE
1 = HIGH	1	0	1	1	SPARE
	0	1	0	0	RX DMA RANGE (16 bit)
	0	1	0	1	TX DMA RANGE (16 bit)
	0	1	1	0	RX INTERRUPT CONTROL (16 bit)
	0	1	1	1	TX INTERRUPT CONTROL (16 bit)

* SPREN2 IS THE INVERSE OF SPREN1

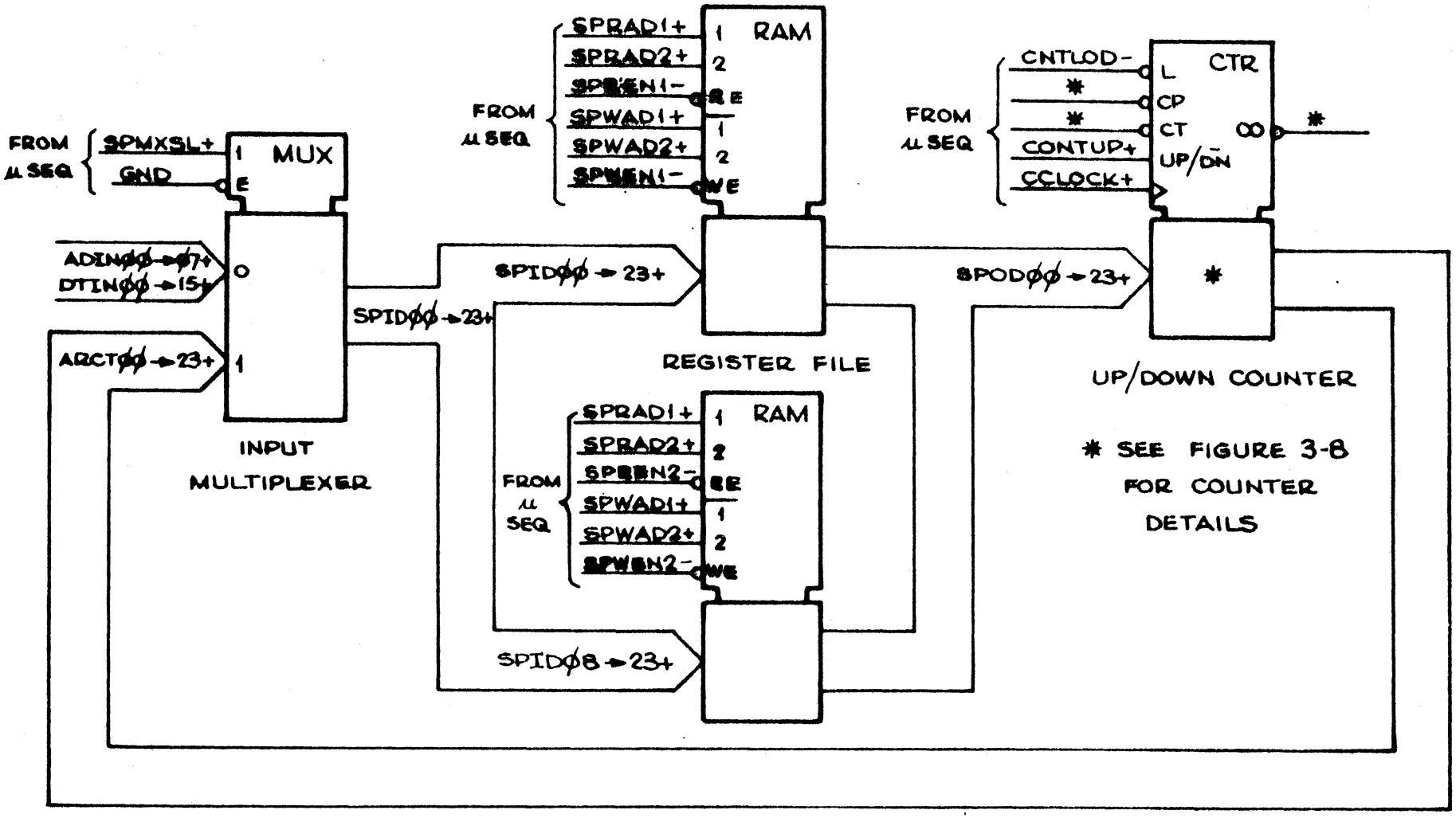


FIGURE 3-7 SCRATCHPAD REGISTER FILE AND ADDRESS/RANGE COUNTER

The register file outputs are fed to the data and address multiplexers for access to the megabus. They are also fed to a 24-bit up-down counter to allow the range and address registers to be decremented and incremented respectively during DMA transfers. This is discussed in the following sections. The register file is loaded via a 2-input multiplexer, the input being selected by SPMXSL+ (a microsequencer output). When SPMXSL+ is low, data from the megabus holding registers is selected (ADIN00-07 and DTIN00-15). ADIN00-07 is only relevant to loading the DMA address registers. When SPMXSL+ is high, the counter outputs are selected so that the modified address or range can be written back.

3.5 UP-DOWN COUNTER (FIGURE 3-8)

The up-down counter is composed of six 74S169 devices, and is used for address/range modification. For the most part operation is straight forward, with a degree of look-ahead carry being implemented. The counter is controlled by four signals as follows:-

a) CCLOCK+ is the clock, being derived from SCLOCK+ under control of CONTEN- (Counter Enable). The counter is loaded, incremented or decremented on the rising edge of CCLOCK+.

b) CNTLOD-, when low causes the counter to be parallel loaded on the next CCLOCK with the output of the scratchpad register file.

c) CONTUP+, when high, sets the counter in UP-counting mode, and when low sets it to DOWN-counting mode.

d) ARC023- is the carry input to the LS stage, and determines whether a count operation takes place or not.

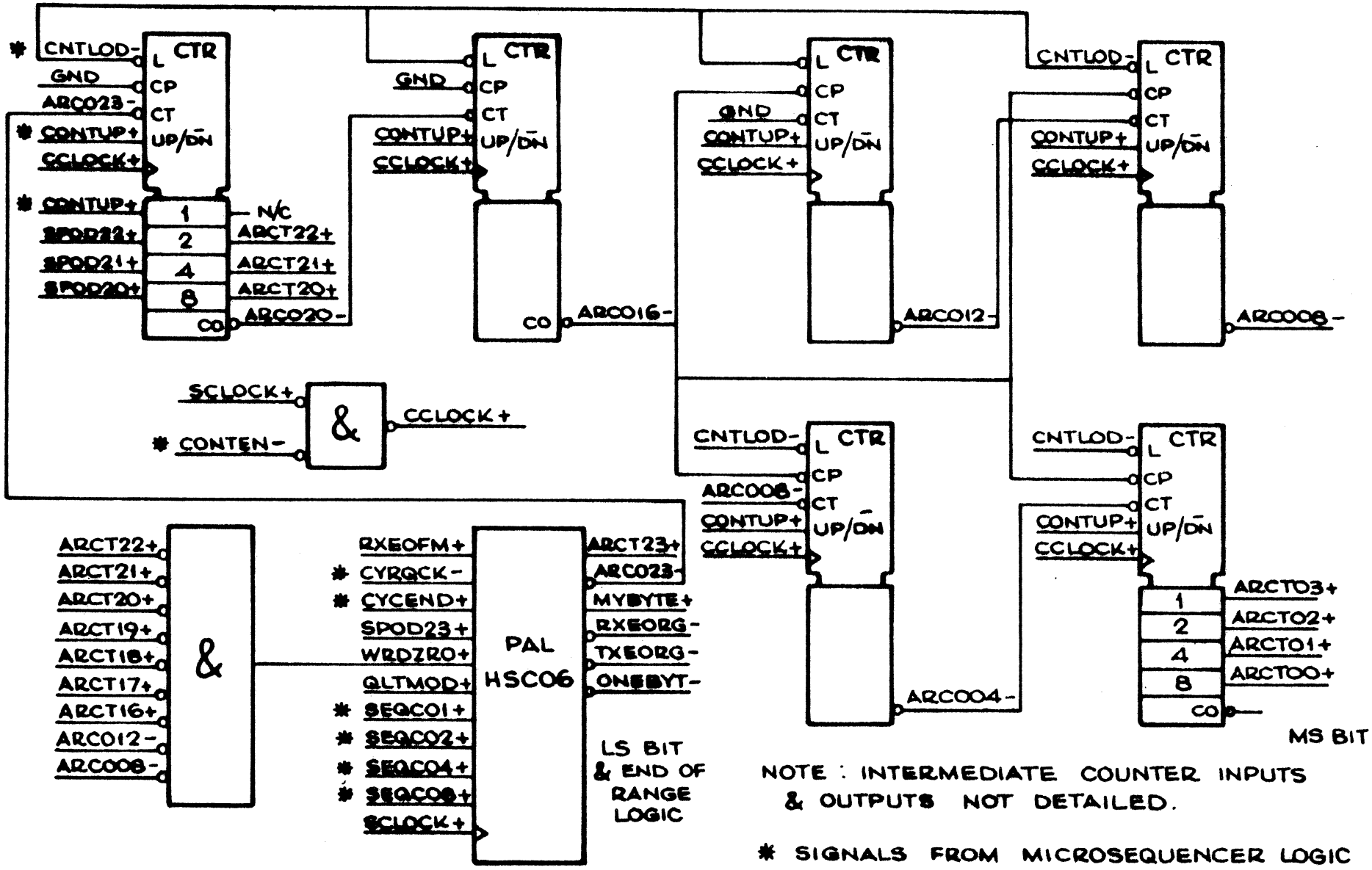


FIGURE 3-8 UP/DOWN COUNTER

Because it is necessary to be able to increment or decrement by 1 or 2 counts (byte or word transfer) the counter is modified in the following way.

1) The LS bit (ARCT23+) is calculated elsewhere in a PAL device HSC06

2) when the counter is loaded CONTUP+ replaces the scratchpad LS bit as input to the counter. Thus the LS bit is forced to zero for down-counting or 1 for up-counting. This is necessary to propagate the carry input (ARC023-) which is also produced by the PAL device.

PAL HSC06 is responsible for a number of functions associated with the up-down counter.

i) It serves as the L.S. bit of the counter, producing ARCT23+, and its carry ARC023-

ii) It detects end of range, via WRDZR0+ (which is true when bits ARCT08- through ARCT22- are zero, when down-counting); setting RXEORG- or TXEORG- low as appropriate.

iii) It decides whether a one or two byte transfer is to take place, setting MYBYTE+ high for byte transfers. For a single-byte last read for the TX channel it sets ONEBYT- low to signal to the TX FIFO logic that only one byte is present.

This is discussed in more detail in the MICROSEQUENCER description.

3.6 MICROSEQUENCER LOGIC

3.6.1 Overview

This logic is designed to handle all the megabus-related operations in an orderly manner, resolving conflict between various possibly co-existent requests. The bus-operations are listed below in order of priority.

1) Programmed I/O - write or read

2) Receive Channel DMA transfer (memory write)

3) Transmit Channel DMA transfer (memory read)

4) Receive Channel Interrupt

5) Transmit Channel Interrupt

These operations are performed by a sequential state machine having 16 possible main states of which 15 are used. Table 3-8 summarises the use of the various states. State zero is the idle state, which is entered at power-up, and to which all other states eventually return. Those states which perform Megabus cycles (2,5,8,9 and D), are further divided into three sub-states as follows:-

Sub-state a) Before requesting bus-cycle

b) Waiting for bus-cycle completion (i.e. ACK or NAK)

c) After bus-cycle completion

The micro-sequencer is logically composed of a number of sub-units, illustrated by Figure 3-9. General operation is as follows. At the end of each operation (or every SCLOCK+ cycle if in state zero) the requests for the operations 1) to 5) above are examined and the highest priority request chosen. The sequence counter is set to the entry point for that service. The requested operation is performed, which may require one or more timing states, after which requests are again examined to decide the next operation. If none is true, then the idle state is re-entered. The following subsections describe the components in more detail.

TABLE 3-8 MICROSEQUENCE SUMMARY

	SEQ No.	ACTIONS
	0	IDLE STATE - WAIT FOR REQUESTS
	1	I/O OUTPUT: LOAD APPROPRIATE REGISTER FILE WITH DATA FROM MEGABUS (VIA DTINXX). RETURN
	2	I/O INPUT: READ APPROPRIATE REGISTER FILE OR OTHER SOURCE: SETUP MUX's PERFORM S.H.B.C. RETURN
RX DMA SEQ	3	RX DMA: LOAD RX RANGE INTO COUNTER FROM REGISTER FILE
RX DMA SEQ	4	RX DMA: READ RX ADDRESS, DECIDE IF WORD OR BYTE TRANSFER
RX DMA SEQ	5	RX DMA: DECREMENT RANGE, REQUEST MEMORY WRITE LOAD RANGE TO REG FILE. LOAD RX ADDRESS INTO COUNTER.
RX DMA SEQ	6	RX DMA: INCREMENT RX ADDRESS
RX DMA SEQ	7	RX DMA: LOAD RX ADDRESS INTO REGISTER FILE RETURN TO IDLE OR START NEW SEQUENCE
	8	RX INTERRUPT: SET UP MUX's POST INTERRUPT RETURN TO IDLE
	9	TX INTERRUPT: SET UP MUX's POST INTERRUPT RETURN TO IDLE
	A	NOT USED
TX DMA SEQ	B	TX DMA: LOAD TX RANGE INTO COUNTER FROM REGISTER FILE
TX DMA SEQ	C	TX DMA: READ TX ADDRESS, DECIDE IF WORD OR BYTE TRANSFER
TX DMA SEQ	D	TX DMA: DECREMENT RANGE, REQUEST MEMORY READ; LOAD RANGE TO REG. FILE LOAD ADDRESS TO COUNTER
TX DMA SEQ	E	TX DMA: INCREMENT TX ADDRESS
TX DMA SEQ	F	TX DMA: LOAD TX ADDRESS INTO REGISTER FILE RETURN TO IDLE OR START NEW SEQUENCE

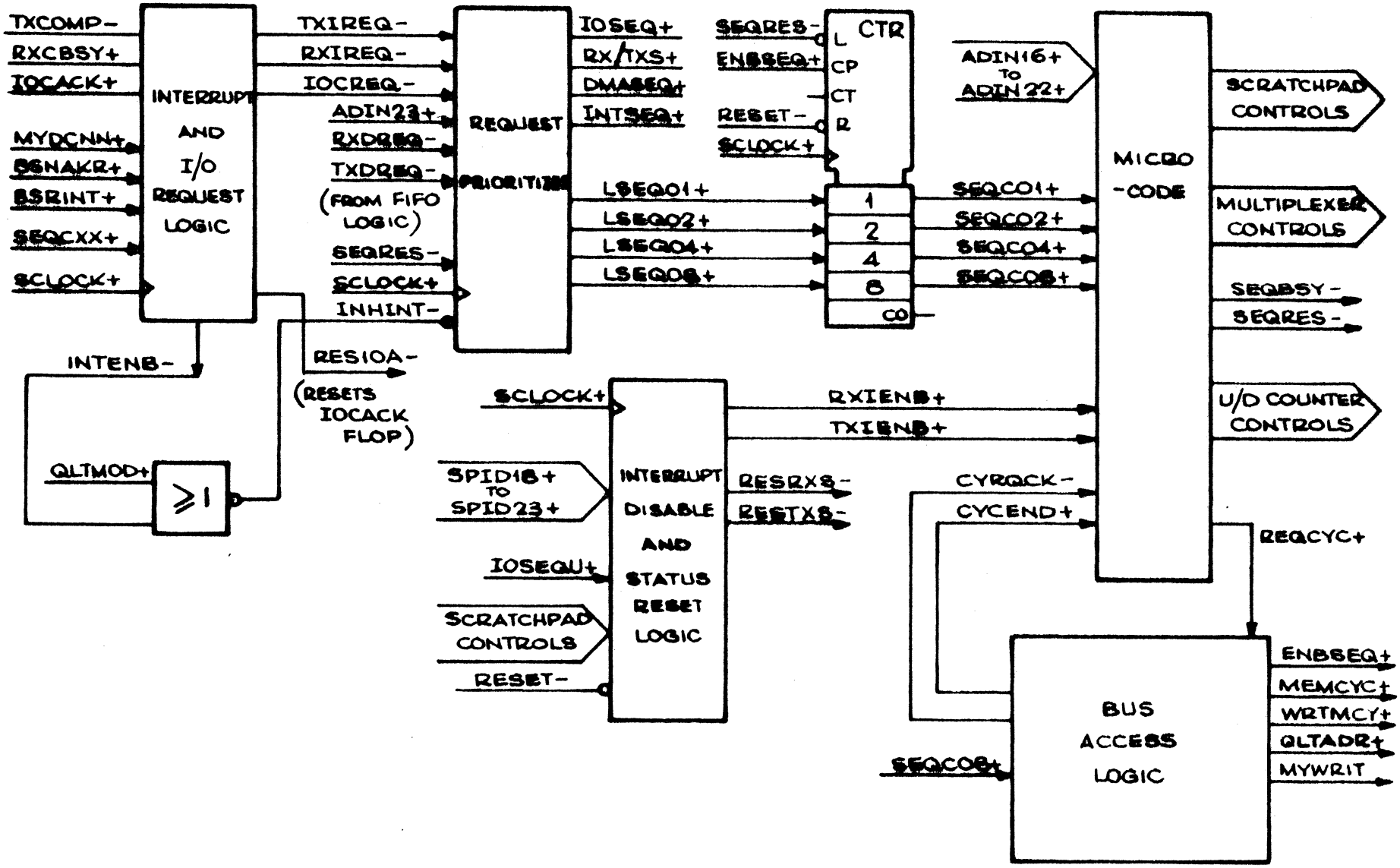


FIGURE 3-9 MICROSEQUENCER BLOCK DIAGRAM

3.6.2 Interrupt and I/O Request Logic

(Fig.3.10)

A PAL device (HSC07) is programmed to generate interrupt and I/O requests to the request prioritizer as follows:-

3.6.2.1 I/O Requests

When an incoming I/O bus cycle is acknowledged, the IOCACK flop is set. This is an input to HSC07, which sets IOCREQ- low on the next SCLOCK pulse to request an I/O sequence. HSC07 monitors the sequence counter SEQCxx and when it detects that state 1 or 2 has been entered (I/O service states) IOCREQ- is reset high, and RESIOA- pulsed low for one SCLOCK period to reset IOCACK. RESIOA is also pulsed at QLT time (via QLTMOD) to initialise IOCACK.

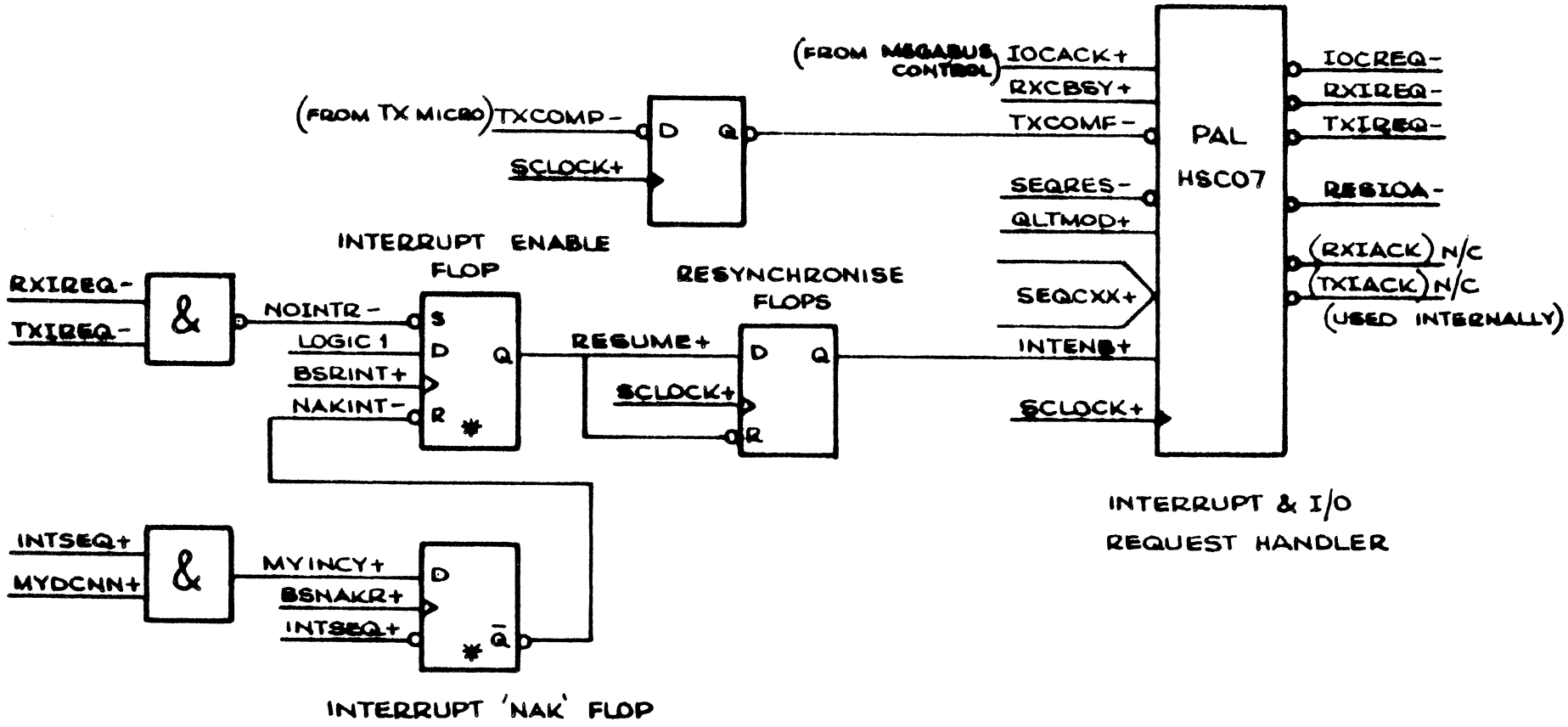
3.6.2.2 Receive Interrupt Requests

RXCBSY+ (Receive Channel Busy) is true while waiting for and transferring a received message. At termination RXCBSY+ falls. HSC07 then sets RXIREQ- low on the following SCLOCK. HSC07 monitors the sequence counter SEQCxx and when state 8 is reached (indicating receive interrupt service) it waits for SEQRES- to go low showing that the bus cycle has been completed. If INTENB+ is true then RXIREQ- is reset high and an internal flop RXIACK is set which prevents further RX interrupts until RXCBSY+ has gone true and false again (i.e. another RX channel message has terminated).

When no interrupts are pending (RXIREQ- and TXIREQ- high) RESUME+ is held high, and so INTENB+ is initially high. This state persists when a request is generated, but if the resulting interrupt bus cycle receives a NAK response (priority level too low) NAKINT- pulses low, resetting RESUME+ and INTENB+. This prevents HSC07 from setting RXIACK, and RXIREQ- remains low, so that the interrupt is still pending. At some time the CPU will pulse BSRINT+, inviting controllers to resubmit their interrupts. When this happens RESUME+ is clocked high, setting INTENB+ on the next SCLOCK. If the subsequent interrupt cycle is acknowledged, then RXIREQ- is reset high, otherwise the process repeats. The INTENB+ flop is necessary to guarantee adequate set-up time for the HSC07 PAL, since BSRINT is asynchronous to SCLOCK.

3.6.2.3 Transmit Interrupt Request

TXCOMP- is driven low by the TX micro at the termination of a transmit frame. This signal is reclocked by SCLOCK to produce TXCOMF- to guarantee adequate set-up time for HSC07. Operation from hereon is similar to the receive channel with TXIREQ- being generated. When state 9 is reached (TX interrupt service) and SEQRES- is low, TXIREQ- is reset and TXIACK set, provided INTENB+ is true. If not, the request remains pending until eventually, following a BSRINT pulse it is finally ACKnowledged, by the mechanism described in 3.6.2.2.



* IMPLEMENTATION IS EQUIVALENT BUT NOT EXACTLY AS SHOWN

FIGURE 3-10 INTERRUPT AND I/O REQUEST LOGIC

3.6.3 Request Prioritizer

This function is performed by a PAL device HSC02.

Its main inputs are the five request signals:-

1) IOCREQ- I/O cycle request from HSC07

2) RXDREQ- Receive Data request from RX FIFO logic

3) TXDREQ- Transmit Data request from TX FIFO logic

4) RXIREQ- Receive Interrupt request from HSC07

5) TXIREQ- Transmit Interrupt request from HSC07

In addition, ADIN23+ is used to distinguish between input and output I/O requests; and INHINT- is used to prevent interrupt sequences from being performed while disabled, which would otherwise waste both sequencer- and bus- cycles.

HSC02 provides two sets of outputs. The LSEQ0x outputs give the start sequence number for the current highest-priority request service routine. If there are no requests then these lines are zeros. LSEQ0x are parallel loaded into the sequence counter at the end of a sequence (or state zero) by SEQRES- going low. At the same time HSC02 sets its second set of outputs to appropriate values to indicate the request being serviced.

These outputs are summarised in table 3-9

TABLE 3-9 PRIORITIZER OUTPUTS

Priority (1=high)	Request	LSEQ0x Binary (Hex)	IOSEQU+	DMASEQ+	INTSEQ+	RX/TXS+
1	{(ADIN23=0	0001(1)	1	0	0	1
	{(ADIN23=1	0010(2)	1	0	0	0
2	RXDREQ	0011(3)	0	1	0	1
3	TXDREQ	1011(B)	0	1	0	0
4	RXIREQ	1000(8)	0	0	1	1
5	TXIREQ	1001(9)	0	0	1	0
6	NONE	0000(0)	0	0	0	0

The action of the prioritizer differs slightly depending on the sequence which is terminating. At the end of DMA sequences (DMASEQ+=1), any pending request will be serviced immediately without going to the idle state. However, for I/O and Interrupt Sequences the prioritizer forces an idle state cycle before servicing any other request. This to ensure that REQCYC+ is allowed to reset. (See Bus Access Logic description, Section 3.6.7).

3.6.4 Sequence Counter

This four bit counter determines the current state of the microsequencer. It is reset at power-up or initialize time to return to idle state. At the end of an operation it is loaded with a new starting value from the request prioritizer by action of SEQRES-. It normally increments by one for every SCLOCK pulse (125ns period) but when megabus accesses are performed, it is held in its current state by ENBSEQ+ being low, until the access is complete (see bus access logic description, section 3.6.7).

3.6.5 Interrupt Disable and Status Logic

This logic, composed of PAL HSC18 and an external NOR gate, monitors the scratch pad control signals output from the microsequencer, and performs the following actions.

1) When the RX address register is being written by an output instruction (i.e. IOLD) or when RESET- is low, it drives RESRXS- low to clear the receive channel status register.

2) When the TX address register is being written by an output instruction (IOLD) or when RESET- is low, it drives RESTXS- low to clear the transmit channel status.

3) When the RX interrupt control register is being written (I/O FC=03) and the interrupt level being written is zero (inhibit) (SPID18+ to 23+ = 0) OR RESET is low, it resets RXIENB+. RXIENB+ is set if a non-zero level is written.

4) Similarly TXIENB+ is set or reset according to the level being written to the TX interrupt control register and reset when RESET- is low.

RXIENB+ and TXIENB+ are condition inputs to the micro-code logic.

3.6.6 Microcode Logic

This logic consists of three PAL devices which form the microprogram of the microsequencer. These decode the sequence counter outputs SEQCOX and for I/O operations ADIN 16+ to 22+ (containing the I/O Function code and sub-channel number) to produce the following groups of outputs

1) Scratchpad Controls:-

SPRAD1, SPRAD2	-	Read address
SPWAD1, SPWAD2	-	Write address
SPREN1, SPREN2	-	Read enables *
SPWEN1, SPWEN2	-	Write enables

* (Note SPREN2 = $\overline{\text{SPREN1}}$)

2) Multiplexer Controls

SPMXSL	-	Scratchpad MUX select
DMSEL1, DMSEL2		Data MUX selects
AUXMUX, LOWMUX		Data MUX enables
ADRSEL		Address MUX select
ADLOEN		Address MUX enable

3) Up/Down Counter Controls

CONTEN	Counter Clock Enable
CNTLOD	Counter Load
CONTUP	Count UP/DOWN Control

4) Sequence Controls

SEQBSY	Sequence Busy (i.e. not state 0)
SEQRES	Sequence Reset (i.e. end of operation)
REQCYC	Request Bus Cycle

Some of these outputs are further decoded or combined in the Bus Access Logic. Although the main states of the microprogram are defined by the SEQCOX signals, states which perform Megabus accesses are further divided into three sub-states (a,b,c) by use of CYRQCK and CYCEND from the Bus Access Logic.

Substate	CYRQCK	CYCEND
a) Cycle Not yet requested	False	False
b) Waiting for cyclke complete	True	False
c) Cycle complete	True	True

The microprogram is detailed in Table 3-10.

Blanks in the output fields of Table 3-10 are functionally "don't care" but will usually be forced to logic 1 level.

TABLE 3-10a

MICRO-CODE FOR MICROSEQUENCER (I/O)

	TSTATE	1								2a,b,								2c
	*FUNC. CODE	03	43	09	49	0D	4D	02	42	08	48	0A	4A	0C	4C	**	26 66	x
I	SEQC01+	1		1		1		0		0		0		0		0	0	0
	SEQC02+	0		0		0		1		1		1		1		1	1	1
	SEQC04+	0		0		0		0		0		0		0		0	0	0
N	SEQC08+	0		0		0		0		0		0		0		0	0	0
P	CYRQCK-							X		X		X		X		X	X	0
	CYCEND+							0		0		0		0		0	0	1
U	ADIN16+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	^
T	ADIN17+	0	1	0	1	0	1	0	1	0	1	0	1	0	1	X	X	
	ADIN18+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
S	ADIN19+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	A
	ADIN20+	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	0	S
	ADIN21+	0	0	0	0	1	1	0	0	0	0	0	0	1	1	X	1	
	ADIN22+	1	1	0	0	0	0	1	1	0	0	1	1	0	0	X	1	
	SPRAD1+							0	1	0	1	0	1	0	1			2a,
	SPRAD2+							1	1	0	0	0	0	0	0			
	SPREN1-							1	1	0	0	0	0	1	1			2b
O	SPWAD1+	0	1	0	1	0	1											
U	SPWAD2+	1	1	0	0	0	0											
	SPWEN1-	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
T	SPWEN2-	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	
P	SPMXSL+	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	
	ADRSEL+							1	1	1	1	1	1	1	1	1	1	
U	ADLOEN-							0	0	0	0	0	0	0	0	0	0	
T	DMSEL1+							1	1	1	1	1	1	1	1	0	1	
	DMSEL2+							0	0	0	0	1	1	0	0	1	1	
S	AUXMUX-							1	1	1	1	0	0	1	1	1	1	
	LOWMUX-							0	0	0	0	1	1	0	0	0	0	
	CONTEN-	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	CNTLOD-																	
	CONTUP+																	\/
	SEQBSY-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	SEQRES-	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0
	REQCYC+	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

* - FUNCTION CODE includes direction bit
e.g. 43 = Function code 03 on TX channel

** - Status Reads -18,1A,1C,58,5A or 5C

TABLE 3-10b

MICRO-CODE FOR MICROSEQUENCER (DMA)

	TSTATE	RX DMA							TX DMA						
		3	4	5a	5b	5c	6	7	B	C	Da	Db	Dc	E	F
	SEQC01+	1	0	1	1	1	0	1	1	0	1	1	1	0	1
I	SEQC02+	1	0	0	0	0	1	1	1	0	0	0	0	1	1
N	SEQC04+	0	1	1	1	1	1	1	0	1	1	1	1	1	1
P	SEQC08+	0	0	0	0	0	0	0	1	1	1	1	1	1	1
U															
T	CYRQCK-	1	1	1	0	0	1	1	1	1	1	0	0	1	1
S	CYCEND+	0	0	0	0	1	0	0	0	0	0	0	1	0	0
	SPRAD1+	0	0	0	0	0	0	0	1	1	1	1	1	1	1
	SPRAD2+	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	SPREN1-	1	0	0	0	0	0	0	1	0	0	0	0	0	0
O	SPWAD1+	1	0	0	0	0	0	0	1	1	1	1	1	1	1
	SPWAD2+	1	0	0	0	0	0	0	1	0	0	0	0	0	0
U	SPWEN1-	1	1	1	1	1	1	0	1	1	1	1	1	1	0
	SPWEN2-	1	1	0	0	0	1	1	1	1	0	0	0	1	1
T															
	SPMXSL+	1	1	1	1	1	1	1	1	1	1	1	1	1	1
P	ADRSEL+	1	0	0	0	0	0	0	1	0	0	0	0	0	0
	ADLOEN-	1	0	0	0	0	0	0	1	0	0	0	0	0	0
U															
	DMSEL1+	1	0	0	0	0	0	0	1	0	0	0	0	0	0
T	DMSEL2+	1	0	0	0	0	0	0	1	0	0	0	0	0	0
	AUXMUX-	1	1	1	1	1	1	1	1	0	0	0	0	0	0
S	LOWMUX-	1	0	0	0	0	0	0	1	1	1	1	1	1	1
	CONTEN-	0	1	0	1	0	0	1	0	1	0	1	0	0	1
	CNTLOD-	0	1	1	1	0	1	1	0	1	1	1	0	1	1
	CONTUP+	0	0	0	0	0	1	1	0	0	0	0	0	1	1
	SEQBSY-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	SEQRES-	1	1	1	1	1	1	0	1	1	1	1	1	1	0
	REQCYC+	0	0	1	1	1	0	0	0	0	1	1	1	0	0

Table 3-10c

MICRO-CODE FOR MICROSEQUENCER (INTERRUPTS AND IDLE)

	TSTATE	RX INTERRUPT INHIBITED	RX INTERRUPT ENABLED		TX INTERRUPT INHIBITED	TX INTERRUPT ENABLED		IDLE
		8	8a,b	8c	9	9a,b	9c	0
	SEQC01+	0	0		1	1		0
I	SEQC02+	0	0		0	0		0
N	SEQC04+	0	0		0	0		0
P	SEQC08+	1	1		1	1		0
U								
T	CYRQCK-	1	X	0	1	X	0	1
S	CYCEND+	0	0	1	0	0	1	0
	RXIENB+	0	1		X	X		X
	TXIENB+	X	X		0	1		X
	SPRAD1+	0	0	0	1	1	1	
	SPRAD2+	1	1	1	1	1	1	
	SPREN1-	1	1	1	1	1	1	
	SPWAD1+							
O	SPWAD2+							
	SPWEN1-	1	1	1	1	1	1	1
U	SPWEN2-	1	1	1	1	1	1	1
T	SPMXSL+							
	ADRSEL+	0	0	0	0	0	0	
P	ADLOEN-	1	1	1	1	1	1	
U	DMSEL1+	1	1	1	1	1	1	
	DMSEL2+	0	0	0	0	0	0	
T	AUXMUX-	0	0	0	0	0	0	
	LOWMUX-	0	0	0	0	0	0	
S								
	CONTEN-	1	1	1	1	1	1	1
	CNTLOD-							
	CONTUP+							
	SEQBSY-	0	0	0	0	0	0	1
	SEQRES-	0	1	0	0	1	0	0
	REQCYC+	0	1	1	0	1	1	0

3.6.7 Bus Access Logic (Figure 3-11)

This logic falls naturally into two parts. At the top of the diagram a collection of gates interpret the outputs of the microcode to generate the bus-qualifier signals and some memory-access related status signals. The lower part of the diagram is concerned with the bus-cycle request.

3.6.7.1 Bus Qualifier and Memory Access Logic

ADRSEL+ is true during HSC response cycles only and is therefore used to drive BSSHBC (Bus Second Half Bus Cycle). ADRSEL+ and ADLOEN- are both low for HSC memory reference cycles, and then generate MYMREF which drives BSMREF. Similarly, memory writes are indicated by LOWMUX- and ADRSEL+ being both low, generating MYWRIT+ and hence BSWRIT. When ADLOEN- and ADRSEL+ are both high QLTADR is asserted, allowing the HSC to address itself for QLT purposes.

MYMREF+ AND MYDCNN+ produce MMDCNN+ (My Memory Data Cycle Now). This is gated with SEQC08+ and SEQC08- to generate WRTMCY and SETMCY which indicate that a write or read memory first half bus cycle is in progress. These are used by the status logic to set the appropriate status flop in the event of an error (e.g. Non-existent memory). When SETMCY is true and an ACKnowledge is received, MEMCYC+ is set to indicate to the function validator that a following SHBC addressed to the HSC is expected.

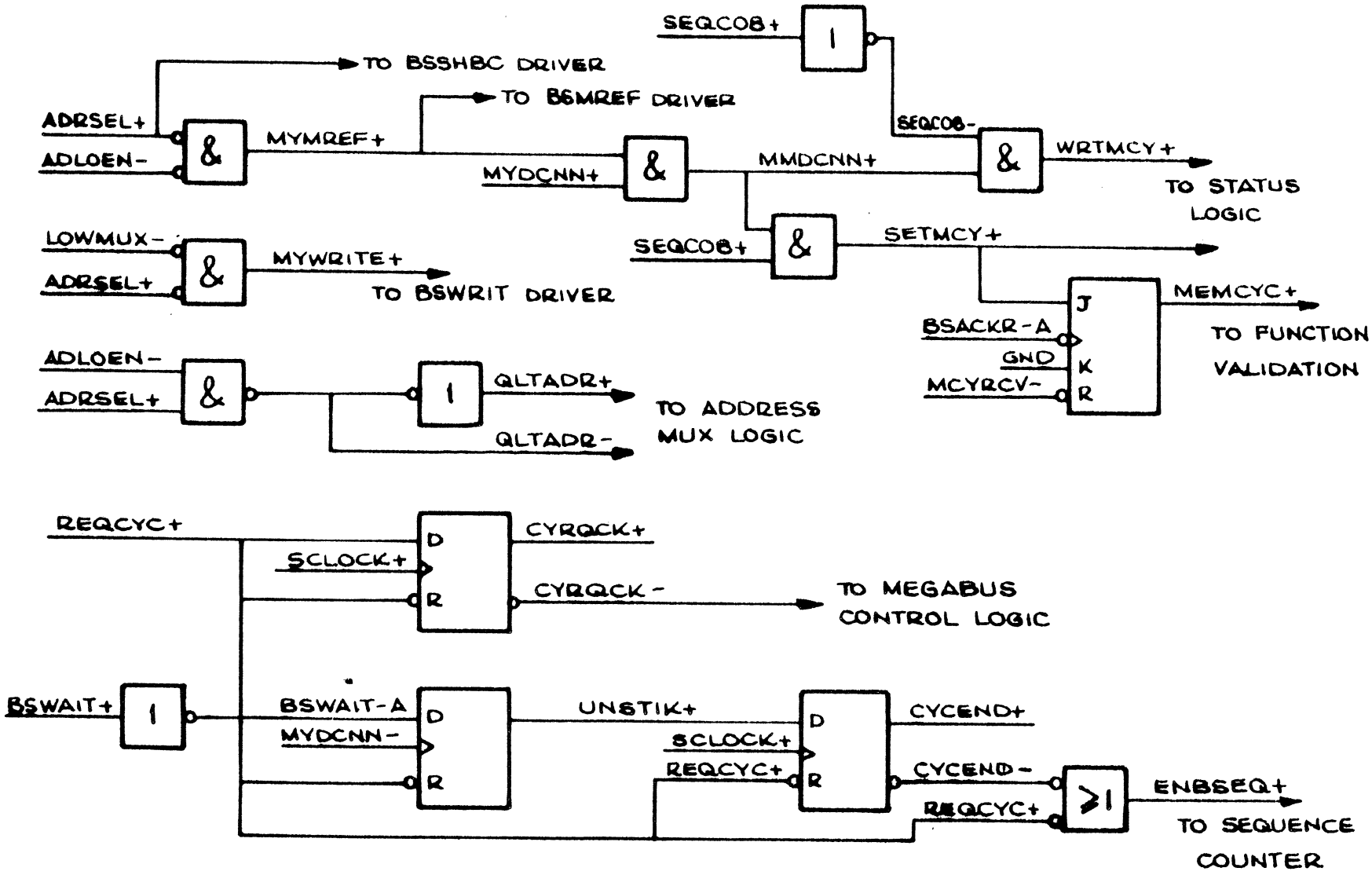


FIGURE 3-11 BUS ACCESS LOGIC

3.6.7.2 Bus cycle request logic

REQCYC+ goes high in those sequence states where a bus-cycle is required (2,5,8,9,D). This removes the reset from CYRQCK, UNSTIK and CYCEND and allows CYRQCK+ to set on the next SCLOCK but forces ENBSEQ+ low preventing the sequence counter from incrementing. CYRQCK setting takes the microsequence from substate a) to substate b) and causes the Megabus Control Logic to request a bus cycle. When a cycle is obtained, the back edge of MYDCNN clocks the UNSTIK flop. If the bus response was WAIT, UNSTIK will remain false because its D input will be low. The megabus control logic will retry until an ACK or NAK response is obtained, when UNSTIK is set. This allows CYCEND+ to set on the next SCLOCK, taking the microsequence to substate c). CYCEND- forces ENBSEQ high allowing the counter to increment if required. However, in some states (i.e. 2,8,9) CYCEND causes the microcode to output SEQRES- low, which loads the sequence counter with its next start point. In any case REQCYC+ will go low in the next state, resetting CYRQCK, UNSTIK and CYCEND.

3.7 TRANSMIT CHANNEL FIFO AND STATUS LOGIC

3.7.1 Word to byte Conversion and Tx FIFO

(Figure 3-12)

This logic forms the data interface between the Megabus at 16 bit data width and the Tx micro with 8 bit data width. During DMA, data is received from main memory and loaded into two byte-wide registers by MCYRCV+. The tri-state outputs of the two registers are commoned so that the left or right-hand byte may be output under control of EVNBYT. There is also a change of signal numbering convention at this point, since the microprocessor subsystems number their data bits the opposite way to DPS6 bit numbering. Table 3-11 shows the conversion.

TABLE 3-11 WORD TO BYTE CONVERSION

	Input Byte		Output Byte
	EVNBYT=0	EVNBYT=1	
MS			
	BSDT08+	BSDT00+	TXFD7+
	BSDT09+	BSDT01+	TXFD6+
	BSDT10+	BSDT02+	TXDS5+
	BSDT11+	BSDT03+	TXFD4+
	BSDT12+	BSDT04+	TXFD3+
	BSDT13+	BSDT05+	TXFD2+
	BSDT14+	BSDT06+	TXFD1+
LS	BSDT15+	BSDT07+	TXFD0+

The transmit channel FIFO is composed of two 16 word deep by 5 bit wide FIFOs, used in parallel. 8 of the 10 bits are used for data, being loaded with TXFDX on the rising edge of TFICLK+. A ninth bit is loaded with LASBYT+, which is used to flag the last byte of the transfer, so that the TX micro can add the CRC and close the frame.

Each FIFO produces an "Input Ready" signal when it has space for more data. The "Input Ready" signals from the two FIFOs are ANDed to produce TFIRDY-, used by the control logic. The FIFO reset, output enable and output clock are under control of the TX micro, which is described later.

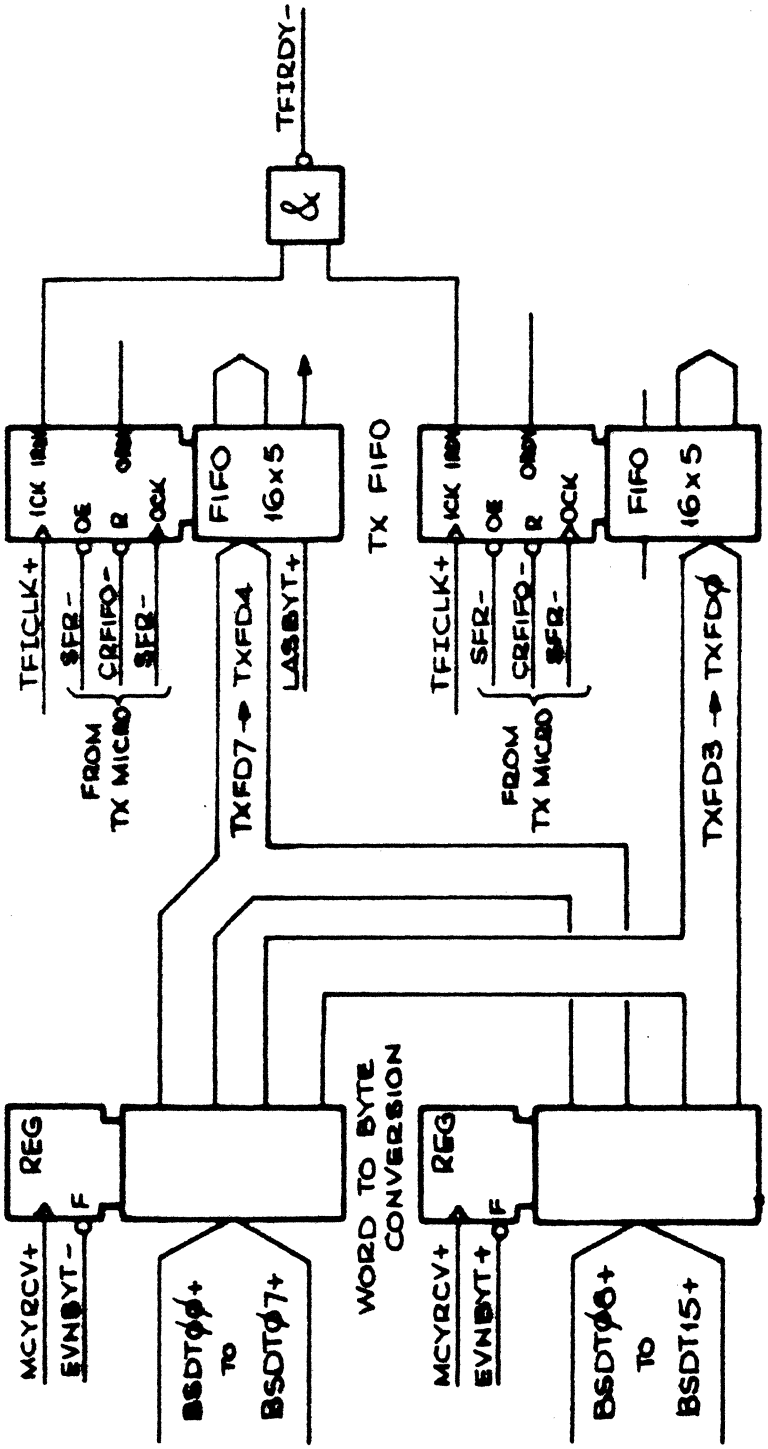


FIGURE 3-12 WORD TO BYTE CONVERSION & TX FIFO

3.7.2 TX FIFO Control (Figure 3-13)

The transmit channel FIFO is controlled by two PAL devices HSC09 and HSC19. HSC09 monitors the scratchpad control signals SPWRTx and SPWADx, and the IOSEQU signal from the request prioritizer. When a TX channel IOLD is issued and ACKnowledged by the HSC, HSC09 detects when data is written to the TX DMA address register, and copies SPID23 (which at this time carries the LS address bit) into the EVNBYT flop.

When the range is written HSC09 sets TXCBSY+ to start the transmit DMA process.

At this time TIRDYF- (TFIRDY- resynchronised to SCLOCK by HSC19) is low indicating an empty FIFO and TXRFUL+ is also low, indicating no data in the registers.

This causes HSC09 to drive TXDREQ- low to request a memory read DMA transfer. As soon as the microsequencer addresses the TX DMA range register of the scratchpad to perform the transfer, this is sensed by HSC09 which sets TDRACK- low and TXDREQ- high to avoid duplicating the request. TDRACK- is an interlock flop.

When the memory responds with the data, MCYRCV sets TXRLOD which via HSC09 sets TXRFUL on the next SCLOCK.

HSC09 now drives TFICLK+ low to load the FIFO and toggles EVNBYT on the next SCLOCK to select the appropriate byte of data. HSC19 raises TIRDYF- at this time, making TFICLK+ rise on the following clock. TFIRDY- may or may not go high depending on the state of the FIFO, but if it does go high, TIRDYF- will fall again when TFIRDY- next goes low, indicating room in the FIFO for the next byte, leading to the next pulsing of TFICLK and toggling of EVNBYT.

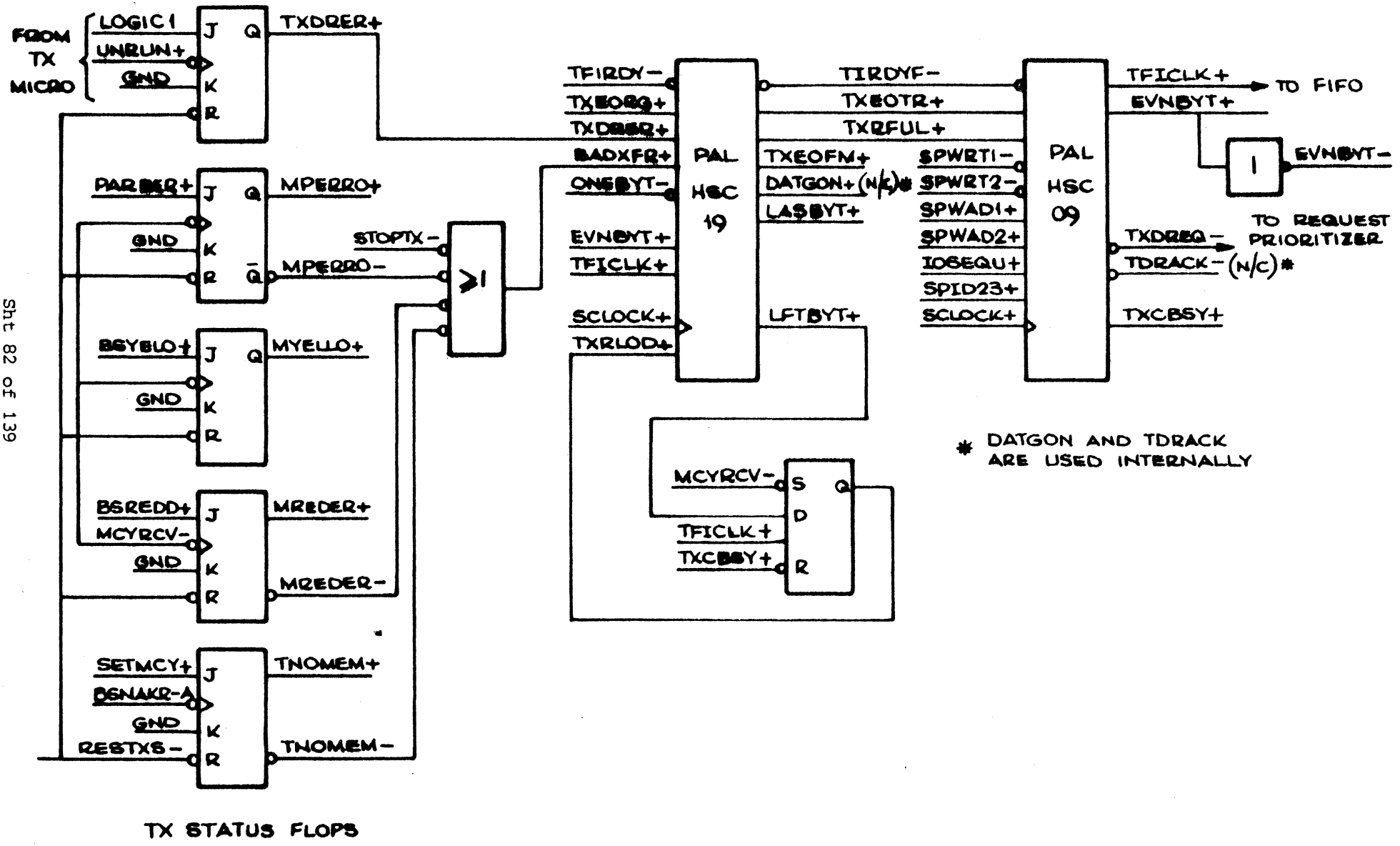


FIGURE 3-13 TX STATUS & FIFO CONTROL LOGIC

Each time TFICLK pulses the value of LFTBYT is copied to TXRLOD. LFTBYT+ is the AND of EVNBYT+ and ONEBYT-. Normally ONEBYT- will be high, so that every time a right-hand byte is pushed into the FIFO, TXRLOD+ is reset, causing HSC09 to ask for another DMA transfer. This continues until a termination condition occurs. This may be due to one of the following.

a) End of Range (TXEORG)

b) Data Rate Error (TXDRER)

c) Parity Error (MPERRO)

d) Memory Red Error (MREDER)

e) Unavailable Resource (TNOMEM)

Any of these cause TXEOTR to be asserted (via HSC19) following the reset of TXRLOD+.

HSC09 then resets TXCBSY and issues no more data requests.

In the case of end of range termination, this may be on an odd-byte boundary, in which case ONEBYT- will be low forcing LFTBYT+ low, so that TXRLOD+ resets after the first byte of the word is transferred. This prevents an unwanted byte from being pushed into the FIFO.

As the last byte is loaded into the FIFO, HSC19 asserts LASBYT+, which is also a FIFO input, to flag to the TX micro that this is the last byte of the frame.

The timing is illustrated in Figure 3-14.

3.7.3 TX Status Logic (Figure 3-13)

The TX status logic is composed of five JK flops.

These are all reset at IOLD time by RESTXS-, and maybe set by various error conditions during the transfer. Except for MYELLO, any of these flops setting will cause termination of the transfer.

3.7.3.1 TXDRER

TXDRER (Transmit Data Rate Error) is set by a pulse on UNRUN+ from the transmit micro-system to signify that no data was available when required for transmission.

3.7.3.2 MPERRO

MPERRO (Memory Parity Error) is set if a bus parity error is detected when DMA data is received from the memory.

(PARRER true at MCYRCV time).

3.7.3.3 MYELLO

MYELLO (Memory Yellow) is set if BSYELLO is true on receipt of DMA data from memory, and indicates that a successful error correction was performed. This is non-fatal.

3.7.3.4 MREDER

MREDER (Memory Red Error) is set if BSREDD is true on recipt of DMA data from memory, and indicates that an uncorrectable error was detected.

3.7.3.5 TNOMEM

TNOMEM (TX No Memory) is set by BSNAKR being received in response to a DMA memory read request, indicating either bad parity was detected by the memory on the first half cycle, or that the memory address was outside the range of available memory.

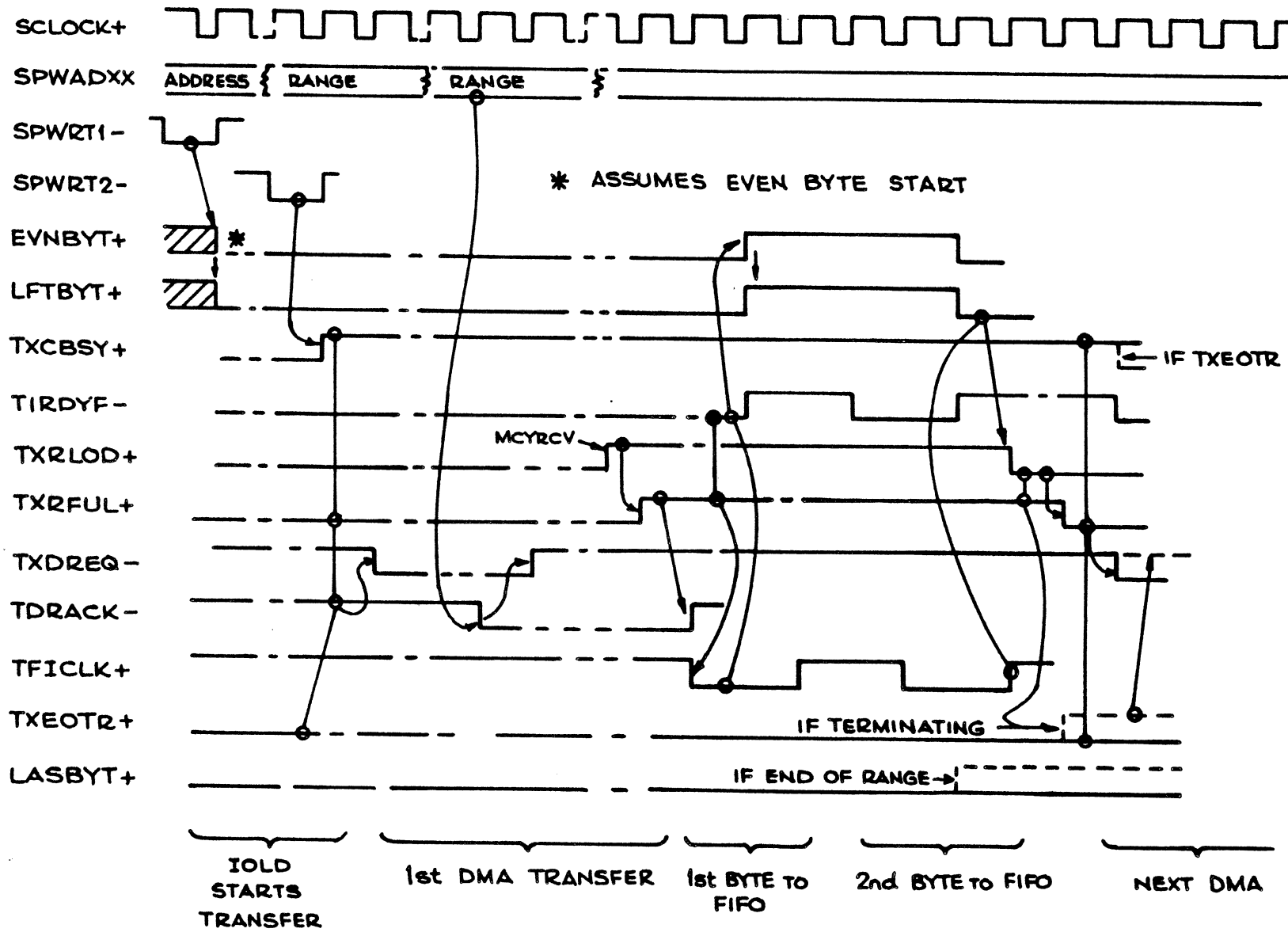


FIGURE 3-14 TX FIFO CONTROL TIMING

3.8 RECEIVE CHANNEL FIFO AND STATUS LOGIC

3.8.1 RX FIFO and Byte to Word Conversion

(Figure 3-15)

This logic forms the interface between the RX micro with 8 bit data width and the Megabus with 16 bit data width.

During reception of an incoming frame, the receive micro feeds the data into the RX FIFO. This FIFO consists of two 16 word deep by 5 bit wide FIFO chips used in parallel. 8 bits are used for data, outputting RFI0 to RFI7. A ninth bit is used to flag the end of message. When this bit (RFISL+) is set, the accompanying data bits contain status information.

Data output from the FIFO is alternately loaded into 2 eight-bit registers under control of LFTCLK and RITCLK. The outputs of these two registers form the data word for transfer to main memory.

At this point there is a change of signal numbering, since the micros number their bits the opposite way to DPS6. Table 3-10 shows their equivalence.

Each FIFO chip asserts an output ready signal when data is available at its outputs. The output readies from the two FIFO chips are combined to generate RFORDY- which is used by the RX FIFO control logic.

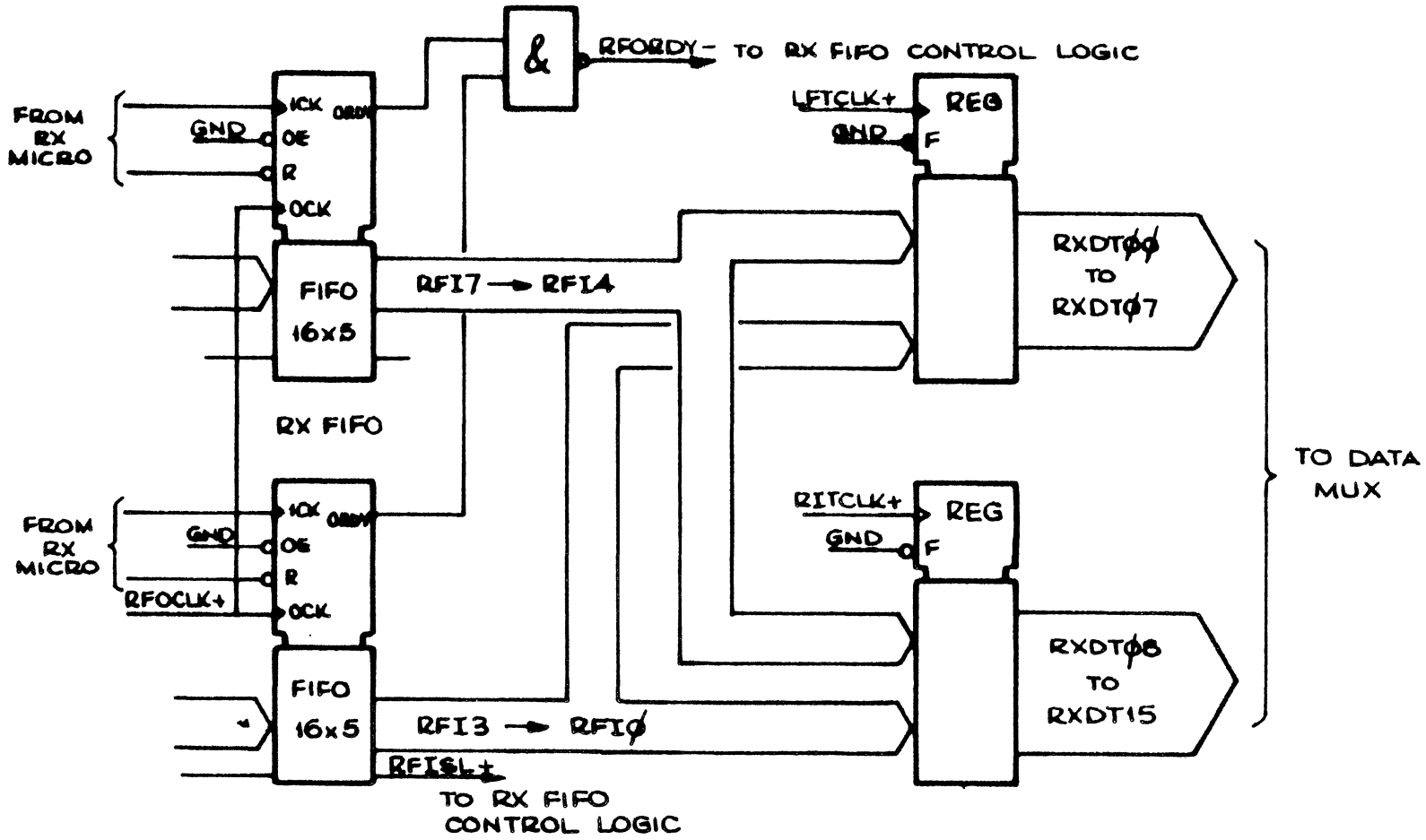


FIGURE 3-15 RX FIFO AND BYTE TO WORD CONVERSION

TABLE 3-10 BYTE TO WORD CONVERSION

	Input Byte	Output Byte LFTCLK	Output Byte RITCLK
MS	RFI7+	RXDT00+	RXDT08+
	RFI6+	RXDT01+	RXDT09+
	RFI5+	RXDT02+	RXDT10+
	RFI4+	RXDT03+	RXDT11+
	RFI3+	RXDT04+	RXDT12+
	RFI2+	RXDT05+	RXDT13+
	RFI1+	RXDT06+	RXDT14+
LS	RFI0+	RXDT07+	RXDT15+

3.8.2 RX FIFO Control

(Figures 3-16 and 3-17)

3.8.2.1 Starting the Transfer

The receive channel FIFO is controlled by two PAL devices HSC09 and HSC08. Both PAL's monitor the scratchpad control signals SPWRTx and SPWADx. When an RX channel IOLD is issued and ACKnowledged by the HSC, HSC08 detects when data is written to the RX DMA address register and copies SPID23 (which at this time carries the LS address bit) into the RODBYT- flop. When the range is written HSC09 sets RXCBSY+ to start the receive DMA process.

At this time RFORDY- is high, since the FIFO is empty. This signal is resynchronised to SCLOCK time to give RORDYF-, used internally by HSC08. When some data is put into the FIFO by the RX micro, RFORDY- and therefore RORDYF- go low.

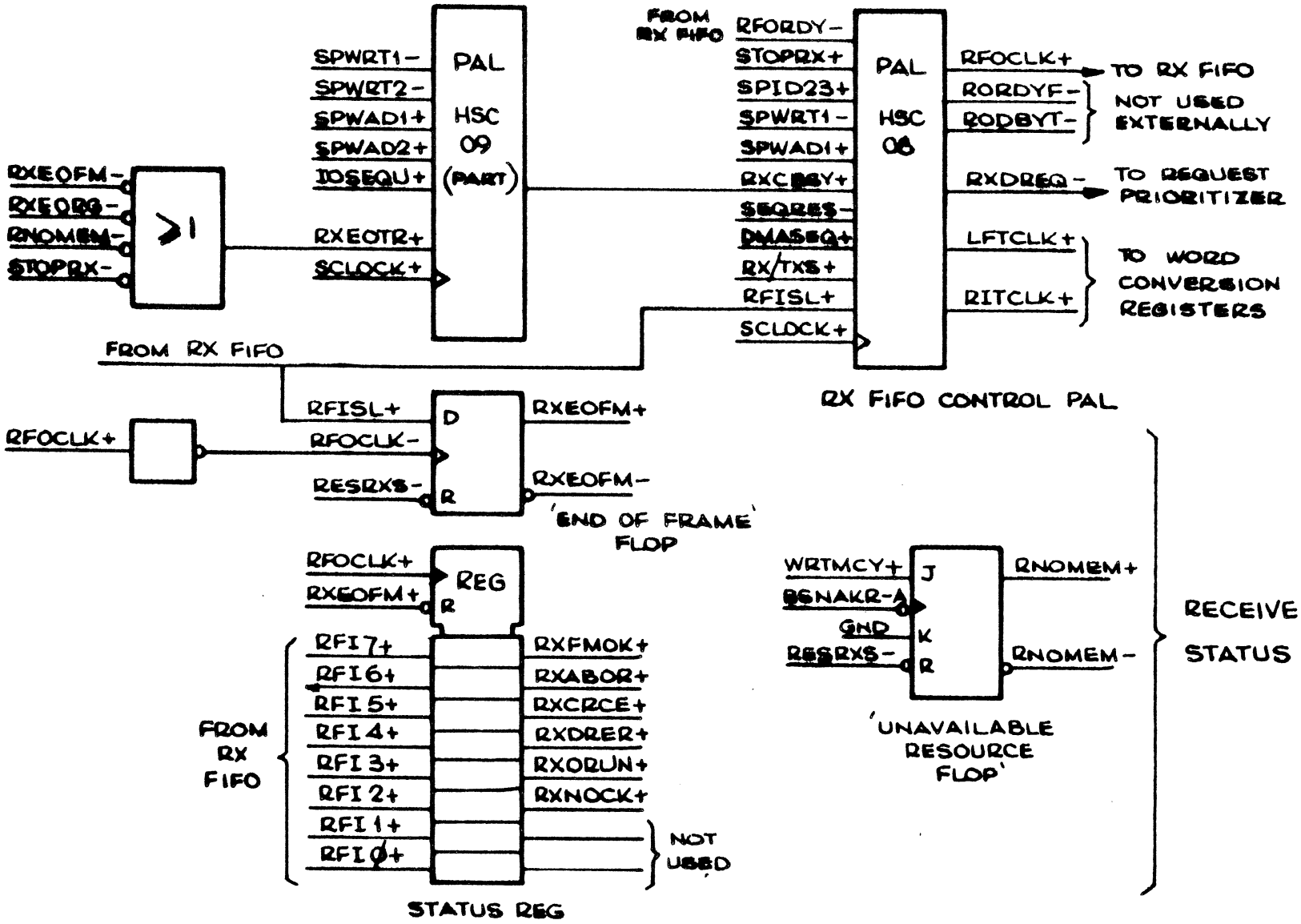


FIGURE 3-16 RX FIFO CONTROL AND RX STATUS

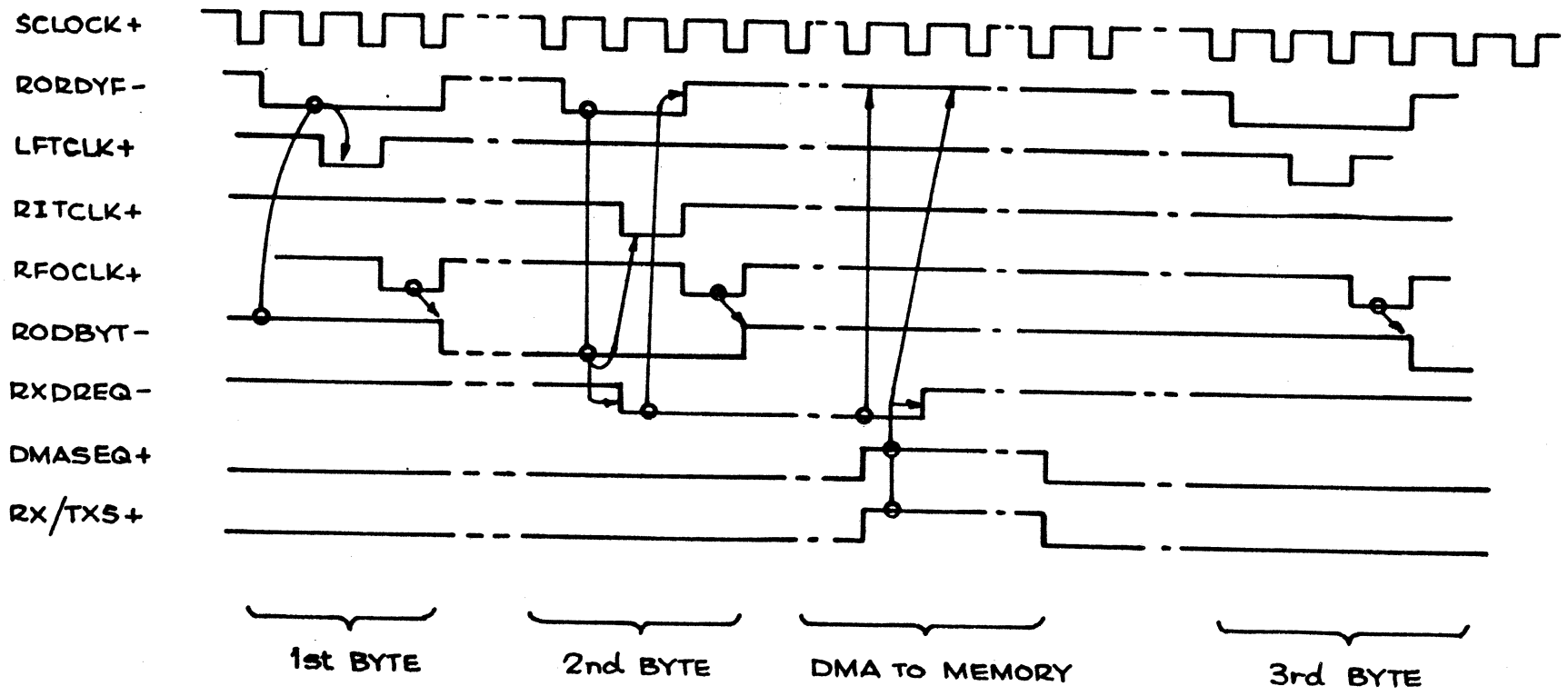


FIGURE 3-17 RX FIFO TIMING

On the next SCLOCK, assuming that RODBYT- was previously loaded high, HSC08 pulses LFTCLK+ low for one cycle to load the FIFO output into the left-byte register. In the next clock period RFOCLK+ pulses low to shift out the used data, causing RFORDY- to go high. RODBYT is toggled at the back edge of every RFOCLK to select alternate byte registers, so that when RFORDY- goes low again, the sequence repeats but with RITCLK+ pulsing instead of LFTCLK+ so loading the right hand byte. A word is now available for DMA transfer, and RXDREQ- is driven low coincident with the leading edge of RITCLK- to request a DMA memory write. RXDREQ- low, forces RORDYF- high, preventing further FIFO accesses. When the DMA write sequence is started (DMASEQ+ and RX/TXS+ high) RXDREQ- is reset high on the next clock, but now the presence of DMASEQ+ and RX/TXS+ serve to keep RORDYF- high. At the end of DMASEQ RORDYF- is allowed to set low when RFORDY- goes low, so continuing the transfer process.

3.8.2.2 Termination of transfer

The receiver DMA transfer is terminated in one of the ways listed below, which are described in more detail in the following sub-sections.

a) End of Range (RXEORG)

b) End of Message (RXEORM)

c) Stop I/O, Stop channel, RESET (STOPRX)

d) Unavailable Resource (RNOMEM)

3.8.2.3 End of Range Termination

In this case termination is forced by the DMA logic of the microsequencer by setting RXEORG. The last operation of the RX FIFO logic will have been to assemble 2 bytes for transfer. For an even byte termination, the DMA logic will transfer both bytes to memory. If range expires on an odd-byte boundary the DMA logic only writes the left hand byte to memory by asserting MYBYTE.

3.8.2.4 End of Message Termination

In this case termination is forced by RFISL going true, indicating that the RX micro termination status byte is present on the FIFO outputs. For an even-byte termination, the FIFO logic will have already assembled the last 2 bytes into a word and asked for DMA transfer (RXDREQ). RFISL true causes HSC08 to output a further pulse on RFOCLK. RXEOFM+ then sets on RFOCLK leading edge, which via RXEOTR resets RXCBSY+ so terminating the transfer.

For an odd-byte termination, the last byte will have been loaded into RXDT00 thru 07. When the status byte, accompanied by RFISL appears on the FIFO output, it is loaded into RXDT08 thru 15 as if it were data. At the same time RXEOFM is set as described above causing termination. Since there are apparently 2 bytes to be transferred to memory, RXDREQ is still set, asking for DMA transfer. The presence of RXEOFM, however, forces the DMA microsequence logic to write only the left byte to memory by asserting MYBYTE.

3.8.2.5 Stop I/O, Stop Channel/Reset Termination

These methods of termination are signalled by STOPRX, which causes HSC09 to reset RXCBSY, at the same time inhibiting HSC08 from generating FIFO clocks or DMA requests. Termination is thus immediate, any bytes in RXDTxx or the FIFO being discarded.

3.8.2.6 Unavailable Resource Termination

This occurs when a DMA write operation to memory receives a NAK response. This may be due to non-existant memory being addressed, or to the memory controller detecting a parity error on the data or module number. The NAK sets the RNOMEM flop, which via RXEOTR resets RXCBSY thus stopping the transfer. Any bytes left in the FIFO or RXDTxx are discarded.

3.8.3 Receive Status Logic

(Figure 3-16)

The receive status logic is composed of the eight flops listed below.

1) RXFMOK - Frame OK

2) RXABOR - Frame was aborted

3) RXCRCE - Frame had a CRC error

4) RXDRER - Data Rate Error
(i.e. DMA did not keep up with message rate)

5) RXORUN - Overrun error
(i.e. Message was longer than buffer)

6) RXNOCK - No Clock (Receive clock from line missing or too slow).

7) RNOMEM - No Memory - NAK received when writing to memory.

8) RXEOFM - End of Frame - RFISL detected.

When the receive channel IOLD is accepted by the HSC, RESRXS- pulses low. This directly resets RNOMEM and RXEOFM, which in turn resets the other 6 flops. These 6 are thus held reset until, at the end of transfer, a status byte is received from the RX micro via the FIFO. When this happens RFISL is asserted which causes HSC08 to pulse RFOCLK, setting RXEOFM on the leading edge of RFOCLK. This removes the reset from flops 1 to 6 above and the status information is loaded into them on the back edge of RFOCLK. When the termination is other than by end of message, this status byte will not appear until after RXCBSY goes false, when the RX micro will clear the FIFO of unwanted data, and load its final status into the FIFO.

RNOMEM monitors the response from the memory to DMA write transfers. When the HSC attempts a write to memory, WRTMCY is true, so that if a NAK response (BSNAKR) is received RNOMEM is set to indicate an "Unavailable resource" error.

3.9 TRANSMIT MICROPROCESSOR SYSTEM

This system provides the means for extracting bytes from the transmit FIFO and serializing them for transmission. It also controls the baud rate generation, and performs quality logic tests.

The Tx microprocessor system is composed of the following parts, as shown on the block diagram, Fig. 3-18.

a) Clock generator
b) 6502B Microprocessor
c) Address Decoder
d) Read-only Memory (EPROM)
e) Read-Write Memory (RAM)
f) Interrupt Control
g) Input Ports
h) Output Ports
i) Pulse Generator Port
j) Baud Rate Generation
k) Communications Controller (2652)
l) I.C.C.U.
m) Transmit FIFO

3.9.1 Clock Generator (Fig. 3-19)

A crystal oscillator produces a 24MHz clock, which is then divided by three to produce F8 and SCLOCK- (for the microsequencer). The 24MHz signal is further divided by PAL HSC10, under control of its various inputs to generate the microprocessor clock CLK0. The PAL provides various clock cycle lengths, depending on the address being accessed, as shown in figure 3-20. When the baud rate is configured to 2MHz, the clock may be suspended awaiting the TXBE signal from the communications controller (2652).

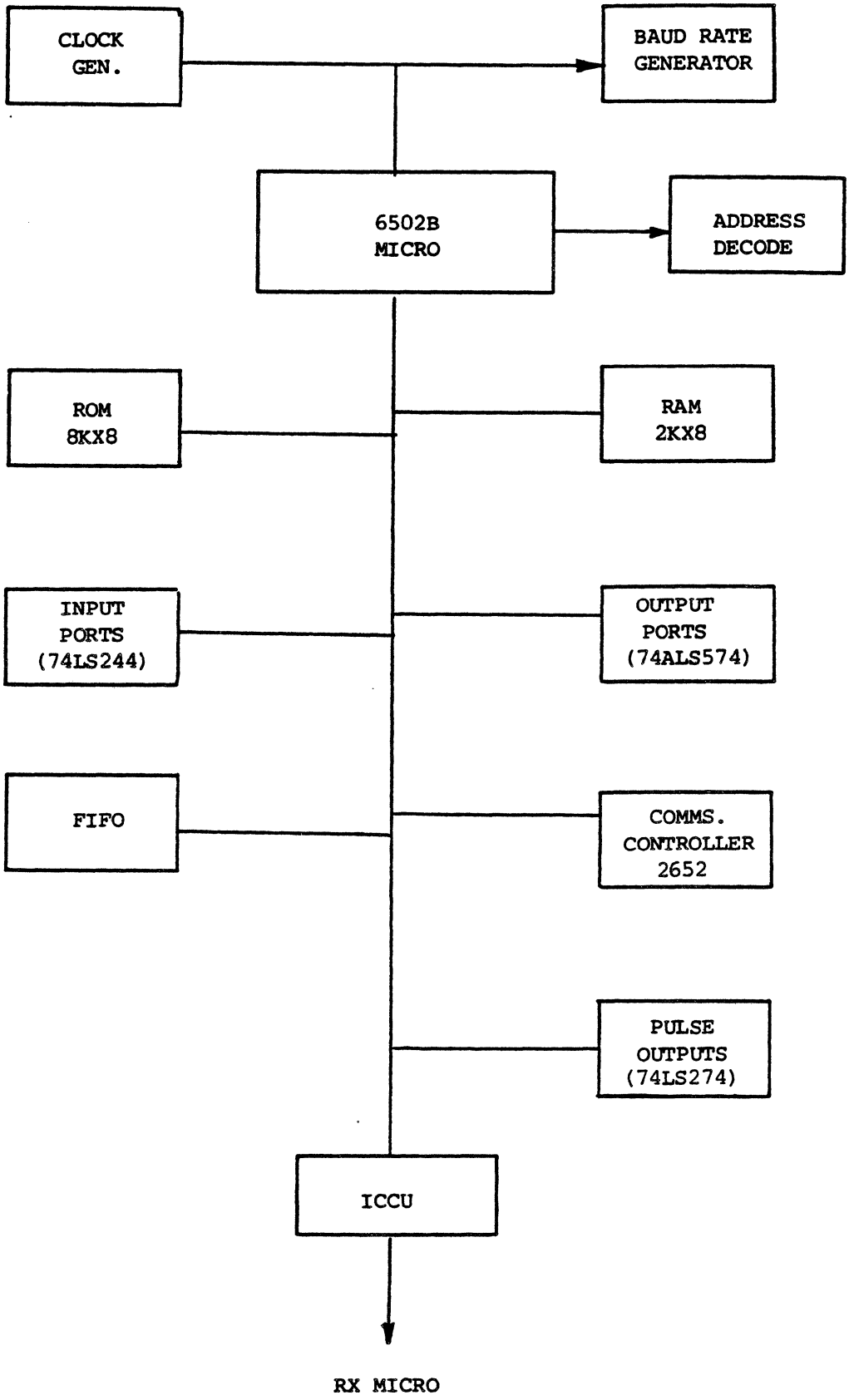


FIGURE 3-18 TX MICROPROCESSOR SYSTEM

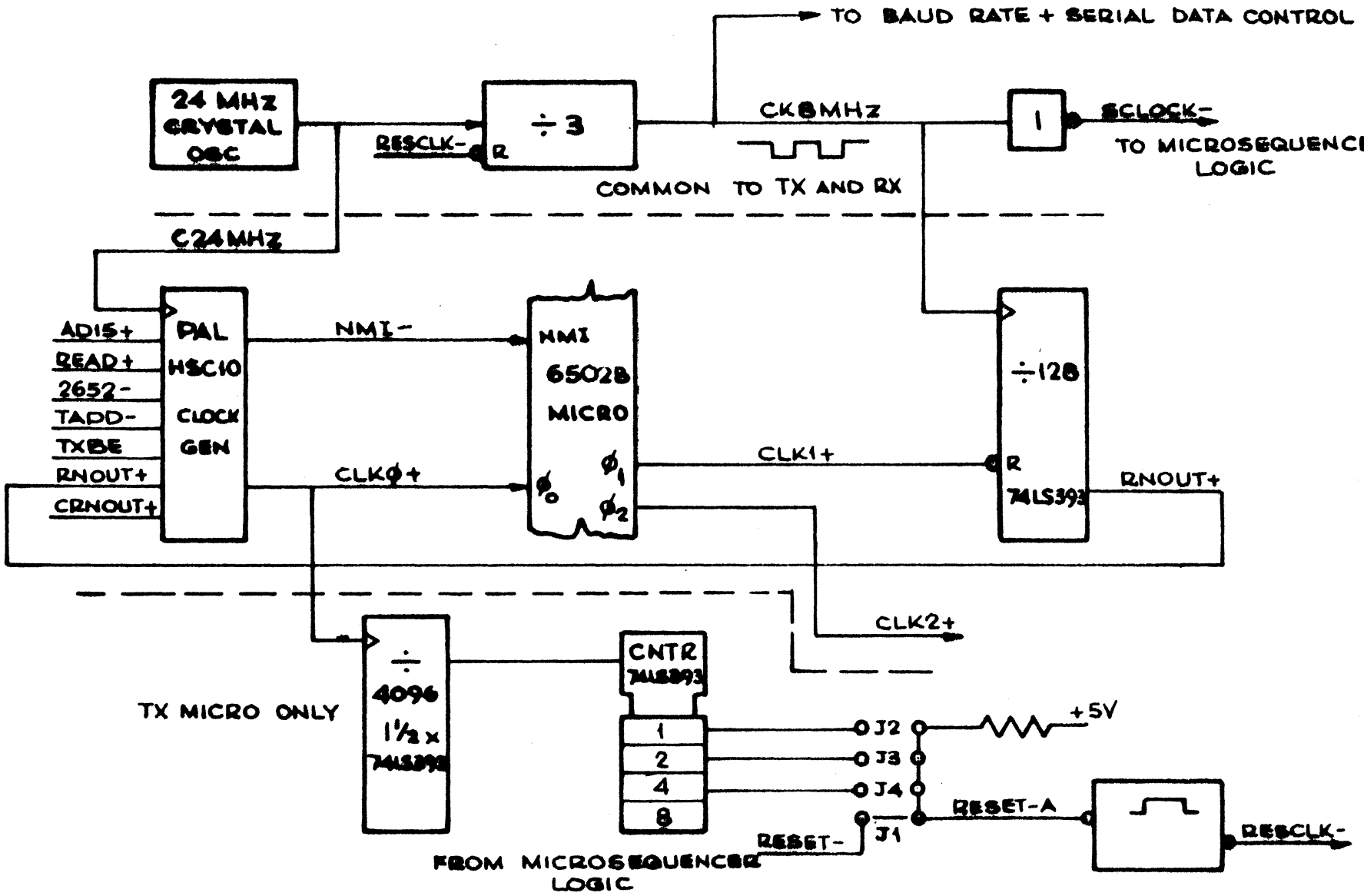
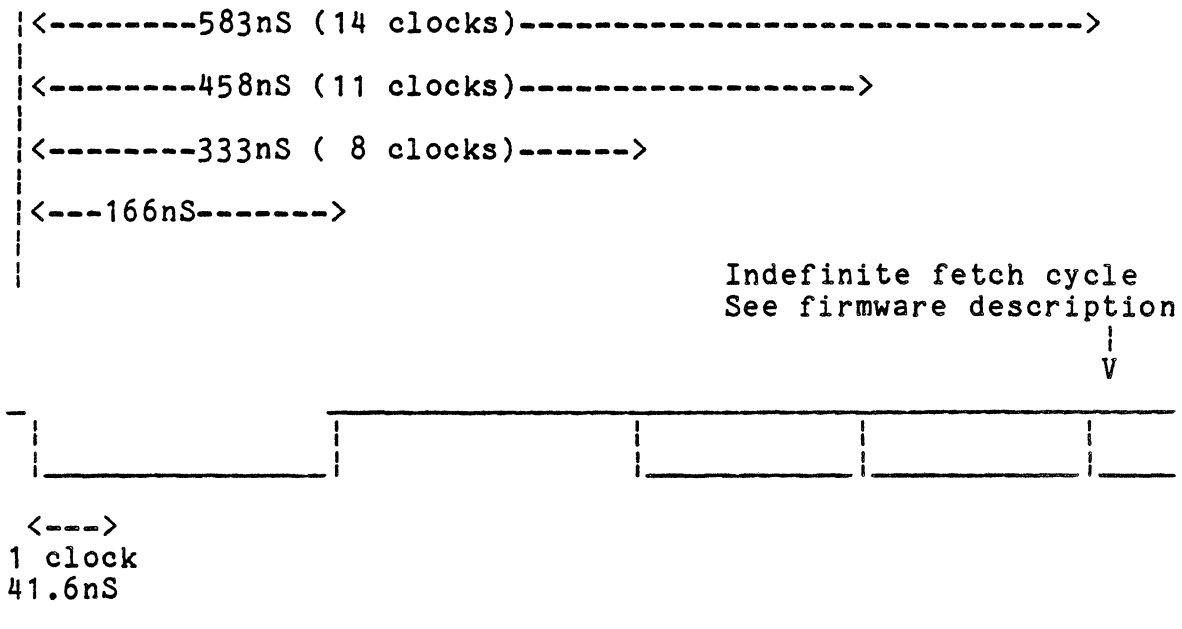


FIGURE 3-19 CLOCK GENERATOR



Indefinite fetch cycle
See firmware description

<---->
1 clock
41.6nS

8 clock cycles for RAM access, FIFO, ICCU, input & output ports & pulse generator port

11 clock cycles for ROM & read from 2652

14 clock cycles for write to 2652

FIGURE 3-20 MICROPROCESSOR CLOCK CYCLES

The 6502B microprocessor generates two other clocks - CLK1+ and CLK2+ from its CLK0 input. CLK2 is used to control peripheral access. CLK1 is used to reset a timeout counter (74LS393), so that if clocks have been suspended for more than 8us, RNOUT+ will go true. This causes HSC10 to restart the clock, and produce a non-maskable interrupt (NMI) to the microprocessor.

Reset for the microprocessor systems is normally derived from RESET- via jumper J1, generating RESET-A and a short pulse RESCLK- used to reset the divide-by-three stage. For microprocessor/firmware debugging, three alternative resets are available via J2 to J4, which will reset the micro's after 4K, 8K or 16K cycles of CLK0.

3.9.2 Microprocessor

A 6502B microprocessor is used, which is capable of operating at 3MHz, and it is run at this rate whenever possible. However, various peripheral access time constraints make it necessary to vary the clock cycle length as described in 3.9.1. This method is more time-efficient than introducing WAIT states, and was necessitated by the high throughput requirement. For more details of the microprocessor itself refer to manufacturers data sheets (Synertek, Rockwell or CBM).

In this application the microprocessor address space is allocated as shown in Figure 3-21. The peripheral port bit assignments are described in Table 3-12.

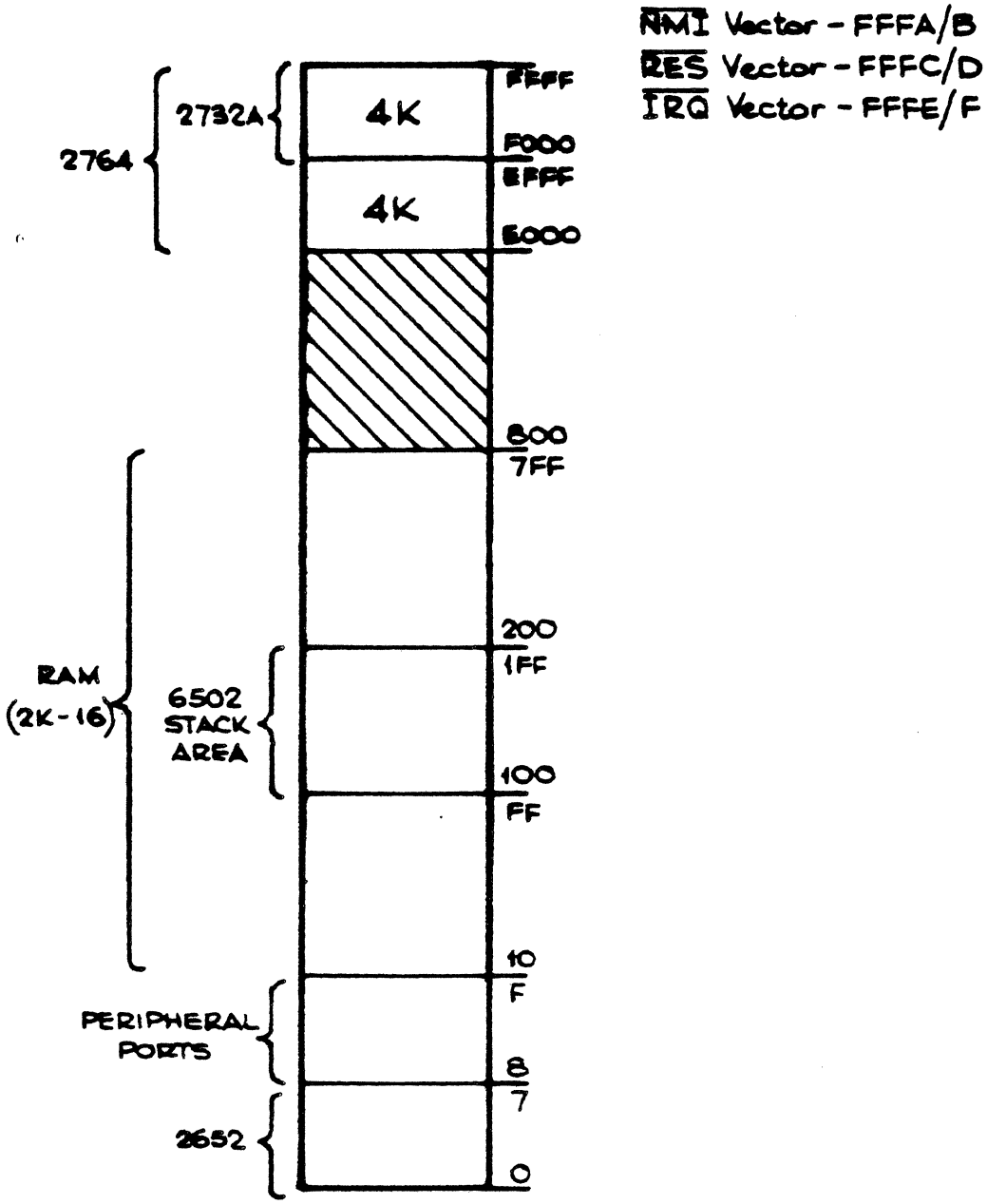


FIGURE 3-21 HSC MICRO MEMORY MAP

TABLE 3-12 Tx MICRO PERIPHERAL PORTS

PORT NO.		BIT NUMBER 7	6	5	4	3	2	1	0
8	R								
	W		TXCOMP-	PAR5	PAR4			DS03-	DS02-
9	R								
	W	BAUD RATE REGISTER							
A	R	TXBE	FOR	TXU		SETDTR	SPECTX	BADXFR	TXACTV
	W			TERRDY	TXE		GEOR		TADDE
B	R	TMM	RMM	EOR	FFE	BAUD RATE SWITCH			
	W								
C	R	MESSAGE FROM RX MICRO							
	W	MESSAGE TO TX MICRO							
D	R								
	W	INTERRUPT MASK REGISTER							
			RMM			BADXFR	TXBE	TXU	FFE
E	R								
	W	PULSED RESETS							
F	R	DATA FROM FIFO							
	W								

R = Read W = Write

3.9.3 Address Decoder (Figure 3.22)

The address decoding is performed primarily by three PAL devices. Address bits 13 and 14 are ignored, while bit 15 is used as a chip-enable for the read-only memory.

PAL HSC13 is the primary decoder. This performs the following actions:-

a) For addresses between 010 and 7FF inclusive makes RAMS- true, selecting read-write memory

b) For addresses between 008 and 00F inclusive makes IOSEL- true, enabling the other two decoders.

c) For addresses 000 to 007 inclusive, makes S2652- true selecting the 2652 Communications Controller

d) For addresses 240 and 248 makes TADD- true, causing the clock generator to suspend CLK0, awaiting TXBE. This function is disabled when TADDE is false. (i.e. for baud rates below 2MHz - see firmware description).

Two identical PAL's (HSC12) are used as read- and write-address decoders. These decode address bits A00 to A02 under control of IOSEL- from HSC13 and the R/W line from the 2652. They are effectively 1 of 8 decoders producing enables corresponding to addresses 008 to 00F of Table 3-11. Not all outputs are currently used.

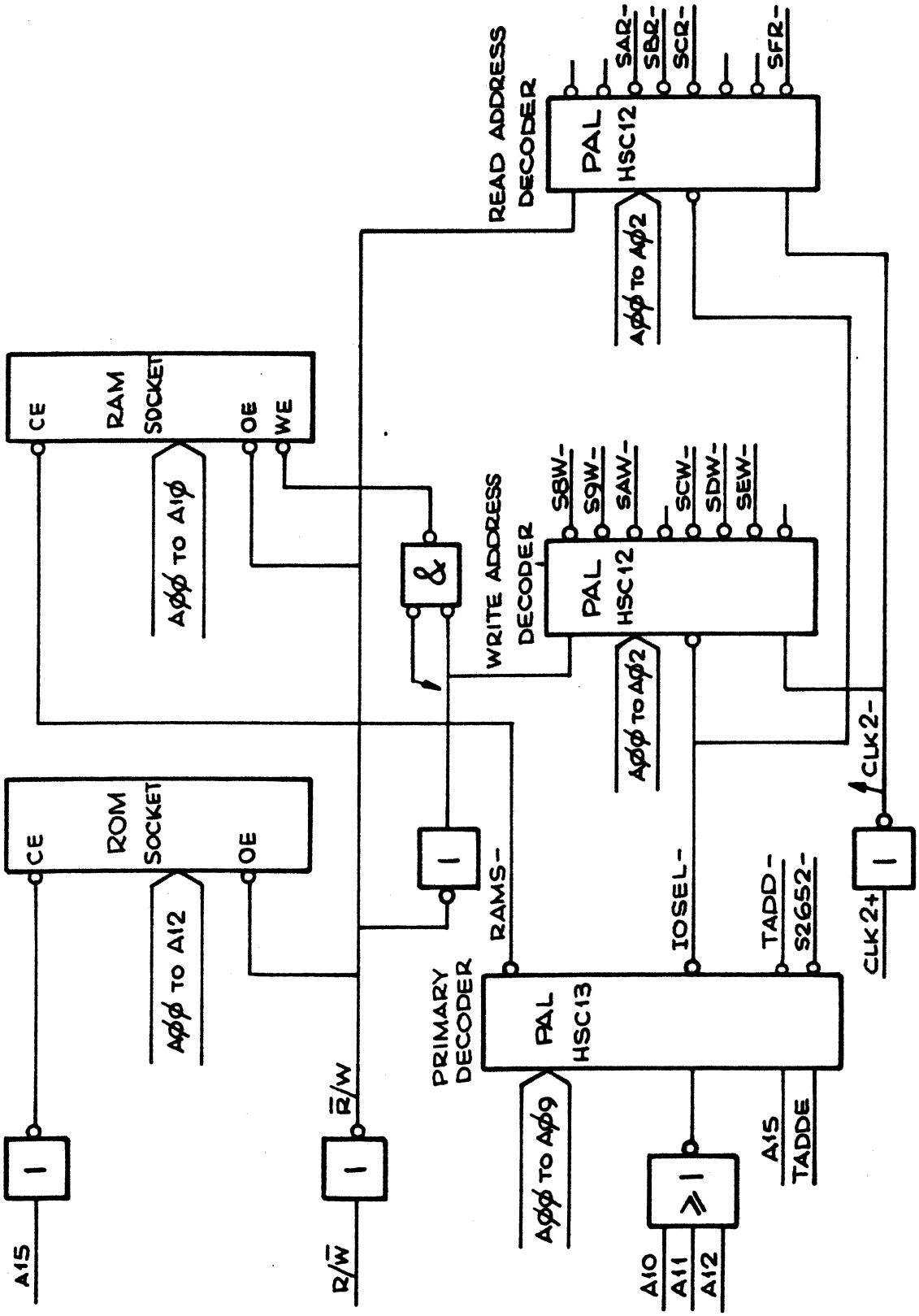


FIGURE 3-22 ADDRESS DECODER

3.9.4 Read-Only Memory (ROM)

A 250-ns access time 2764 EPROM is fitted providing a maximum of 8K bytes of firmware storage. This is enabled by address bit 15, while its tri-state output is controlled by the R/W line from the 6502B.

3.9.5 Read-Write Memory (RAM)

A 100-ns access-time 2016 static ram is fitted providing 2K bytes of working space. Because of its fast access time, some time-critical code is copied to RAM from ROM prior to execution (see firmware description).

The chip is enabled by RAMS-, from the address decoder, while its output is enabled by the R/W line from the 6502B. The write-enable is derived from the R/W line gated with CLK2 from the 6502B.

Provision has been made to allow an 8Kx8 device to be fitted, for possible future enhancements, one cut and one link being required to implement this change.

3.9.6 Interrupt Control (Figure 3-23)

Apart from the non-maskable interrupt (NMI), the transmit microprocessor system has provision for eight external interrupts by means of the logic shown. The 74ALS574 mask register is loaded from the 6502B under control of SDW- and CLK2-. This register drives eight of the inputs of PAL HSC11, the other eight being the interrupts themselves. HSC11 performs an AND-NOR function, gating each interrupt with its mask and ORing the results to generate PIRQ- as input to the 6502B. Thus an interrupt input will interrupt the 6502B provided the corresponding mask bit is true.

For the transmit micro five interrupts are defined, as detailed in table 3-13.

TABLE 3-13

Mask bit no.	Interrupt Signal	Description
0	FFE	Tx FIFO empty signal
1	TXU	Transmitter Underrun - from 2652
2	TXBE	Transmitter Buffer empty - from 2652
3	BADXFR+	Bad transfer - abort required-from u-seq.
4	-	Not used
5	-	Not used
6	RMM	Message from RX micro
7	-	Not used

3.9.7 Input Ports

Two 74LS244 tri-state buffers are used as general input ports to allow the transmit micro to read the baud rate switch and various status bits. These are described in Table 3-14 and 3-15 and are enabled by SAR- and SBR- respectively from the read address decoder.

TABLE 3-14 G.P. INPUT PORT A

BIT NO.	SIGNAL	DESCRIPTION	SOURCE
0	TXACTV	Transmitter Active	2652
1	BADXFR	Bad Transfer - Abort Required	u-seq
2	SPECTX	R.F.U.	"
3	SETDTR	Terminal Ready	"
4	-	Not Used	
5	TXU	Transmitter Underrun	2652
6	FOR	FIFO Output Ready	FIFO
7	TXBE	Transmit Buffer Empty (Ready for next byte)	2652

TABLE 3-15 G.P. INPUT PORT B

BIT NO.	SIGNAL	DESCRIPTION	SOURCE
0	BRC	:	Hex
1	BRB	: > Baud rate Selection	Switch
2	BRA	:	
3	BRD	:	
4	FFE	FIFO empty - micro read FIFO when no data was available	FIFO
5	EOR	End of Range (TERM true)	FIFO
6	RMM	Receive Micro Message available	ICCU
7	TMM	Tx Micro Message not yet read by Rx micro	ICCU

3.9.8 Output Ports

There are 2 general output register ports. These are 74ALS574 devices, which are loaded from the 6502B bus under control of S8W-, and SAW- from the write address decoder. These are described in Table 3-16 and 3-17.

TABLE 3-16 G.P. OUTPUT PORT 8

BIT NO.	SIGNAL	DESCRIPTION
0		Lights LED DS02 when low
1		" " DS03 " "
2	-	Not used
3	-	Not used
4	PAR4	Selects Pulse-width Modulation (2MHz) only
5	PAR5	Enables RS422 drivers
6	TXCOMP-	When low signals transmission complete
7	-	Not used

TABLE 3-17 GP OUTPUT PORT A

BIT NO.	SIGNAL	DESCRIPTION
0	TADDE	Enable indefinite fetch cycle
1	-	Not used
2	GEOR	Enable EOR (End of Range) to set overflow
3		Enable Loopback (for QLT purposes)
4	TXE	Enable Transmitter (2652)
5	TERRDY+	Terminal Ready to RS422 I/F
6		Enable Data and Clock to RS422 I/F
7		Reset 2652

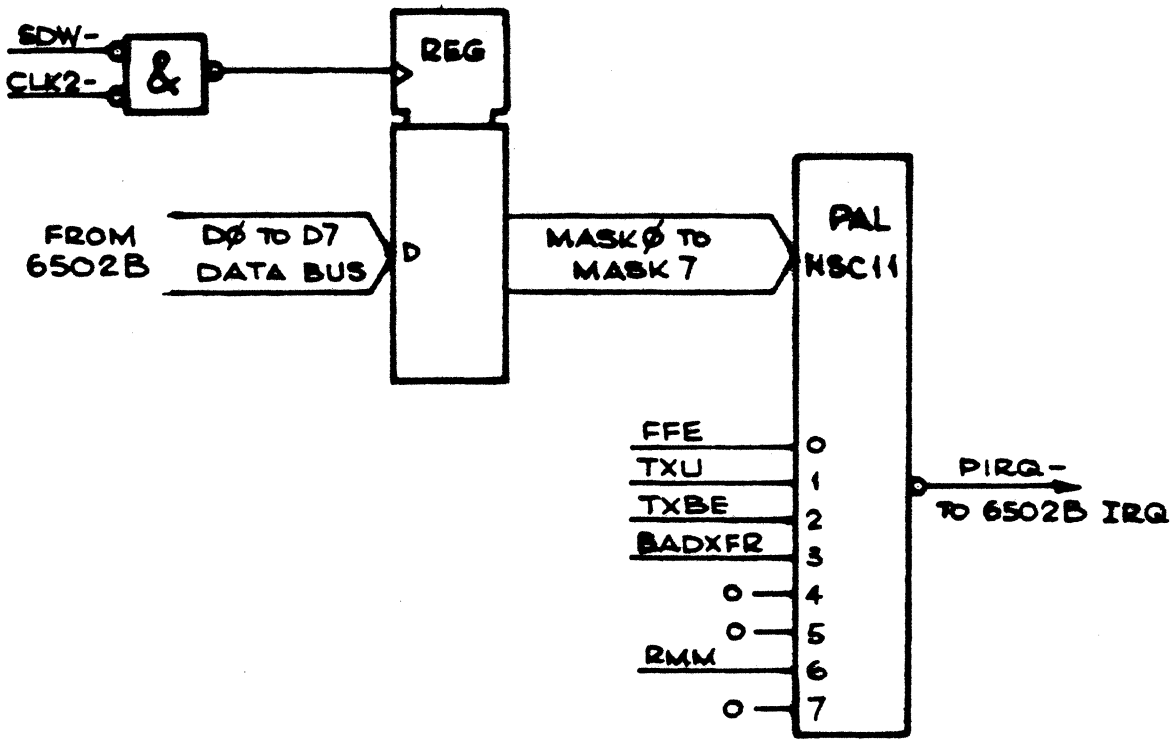


FIGURE 3-23 INTERRUPT CONTROL

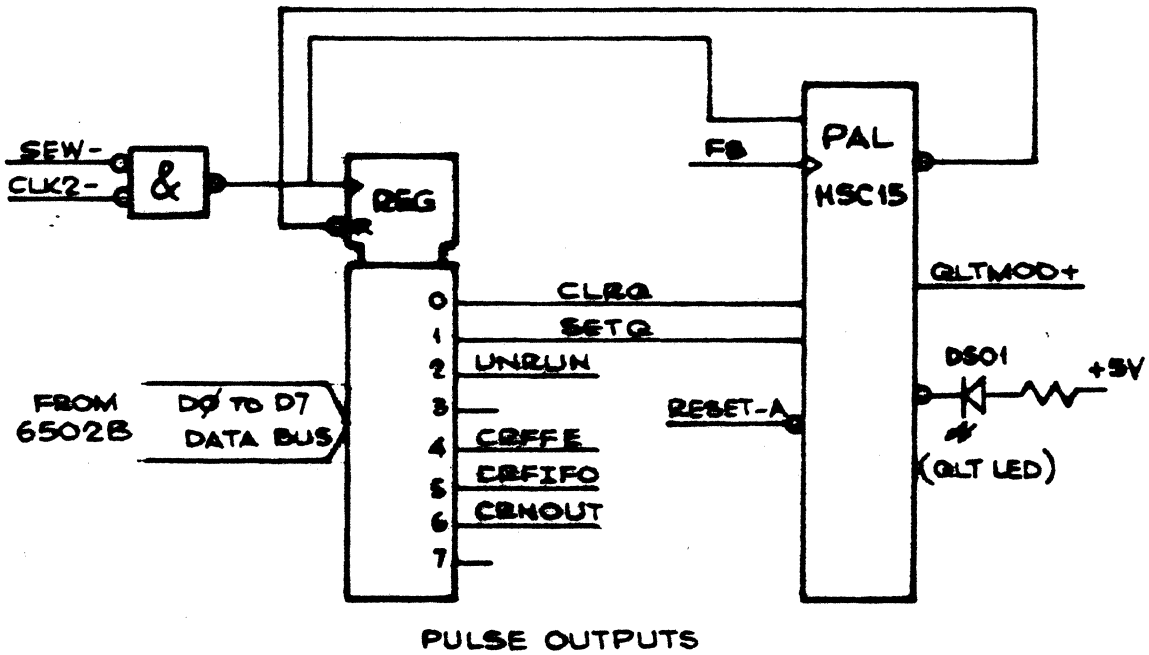


FIGURE 3-24 PULSE OUTPUT PORT

3.9.9 Pulse Output Port (Fig. 3-24)

This port (address E) is a 74LS273 octal register, which is loaded from the 6502B bus under control of SEW- from the write-address decoder. The functions of its various outputs are described in Table 3-18. The clock signal for this register is also fed to PAL HSC15 which generates a reset signal 125 to 250ns after the register is loaded, producing a pulse output facility. Two of the pulse outputs, SETQ and CLRQ are also fed to the HSC15, causing the set and reset respectively of two internal flops which drive QLTMOD+ and the QLT LED DS01. These flops are also set by a low on RESET-A.

TABLE 3-18 PULSE OUTPUT PORT E

BIT NO.	SIGNAL	DESCRIPTION
0	CLRQ	Reset QLTMOD flop and QLT LED
1	SETQ	Set " " " " "
2	UNRUN	Set TXDRER+ flop in TX STATUS
3	-	Not used
4	CLFFE	Reset FIFO empty flop (HSC 16)
5	CRFIFO	Clear Tx FIFO
6	CRNOUT	Clear RUNOUT condition (Clock gen)
7	-	Not used

3.9.10 Baud Rate Generator (Fig. 3-25)

The 8MHz clock F8 is divided by HSC14 producing the 4MHz clock F4. This clocks two 4-bit programmable counters.

The first counter counts up to 15 and then reloads with a value X determined by the MS 4 bits of port 9. Every time the first counter reaches 15 the second counter increments. When the second counter reaches 15, it reloads with a value Y from the LS 4 bits of port 9 when next enabled. The carry output from the second counter is further divided by two in HSC14 to generate the final square-wave clock.

This frequency is thus $\frac{2}{(16-X)(16-Y)}$ MHz

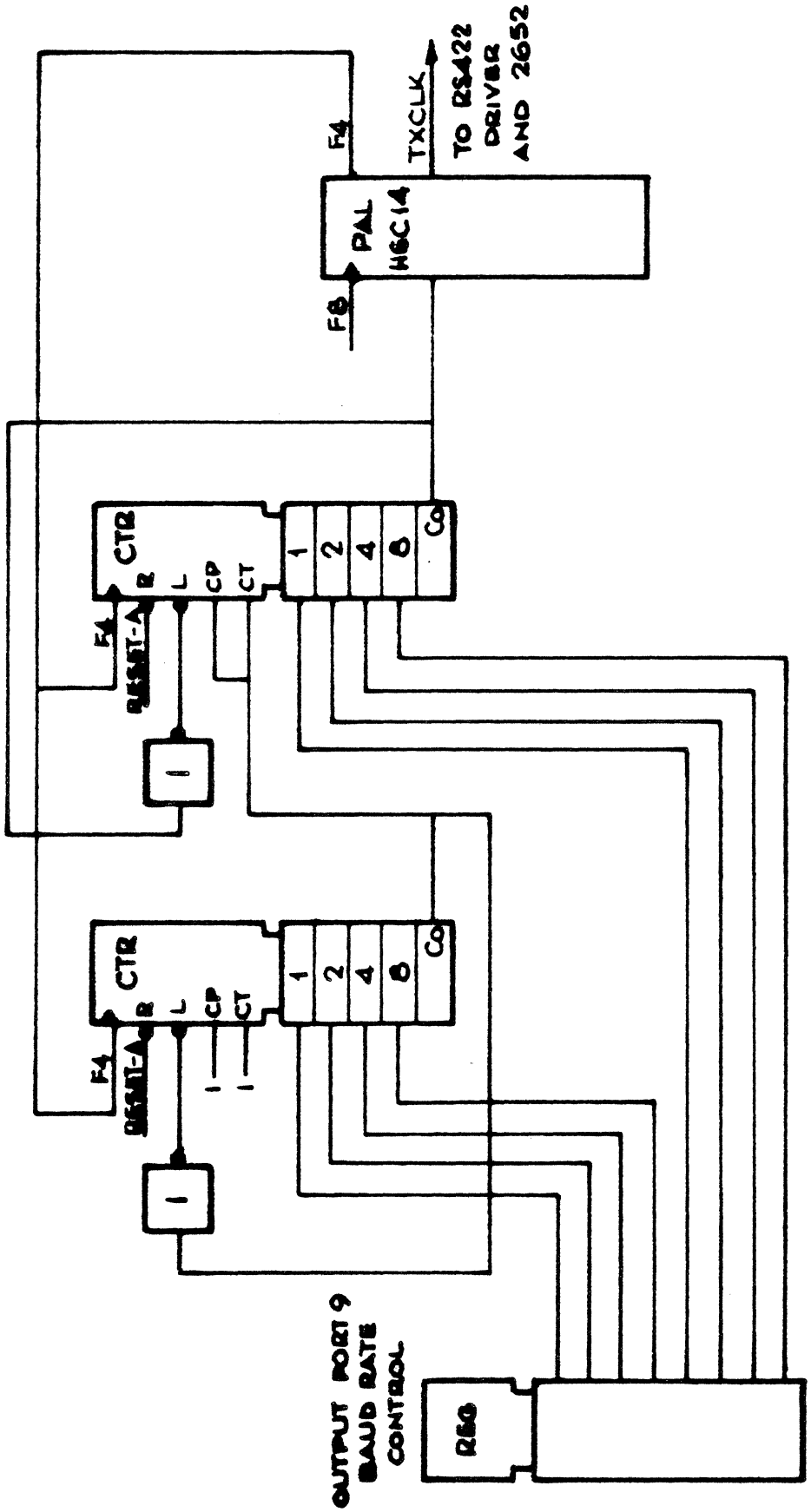


FIGURE 3-25 BAUD RATE GENERATOR

3.9.11 TX Communications Controller

(Figure 3-26)

A 2652 HDLC Communications Controller is used to serialise the data for transmission. This device is responsible for the generation of flags, CRC bytes and when necessary abort sequences. Only the transmitter portion is used.

The device has 8 internal registers, which are addressed by A00 through A02 under control of DBEN and R/W. Since the 2652 is not compatible with the usual 6502B bus timing a converter circuit is used to generate DBEN from the address decode S2652- and CLK2. This works in conjunction with the clock generator (3.9.1.) which produces lengthened clock cycles when the 2652 is selected. In read mode, DBEN follows the lengthened CLK2, since effectively CLK2- appears on the set input and CLK2+ on the reset input of the flop.

In a write operation the data lines must be stable before the rising edge of DBEN. In this case DBEN is set by the rising edge of CLK2+ delayed by 150ns and reset as before on the falling edge of CLK2.

The 2652 is reset by an output from output port A, and transmission is enabled (TXE) by second output.

Three status outputs from the 2652 are read via read port A. These are as follows:-

TXA - Transmitter active (frame in progress)

TXU - Transmitter Underrun (i.e. data has not been loaded in time)

TXBE - Transmit Buffer Empty (i.e. ready for next byte)

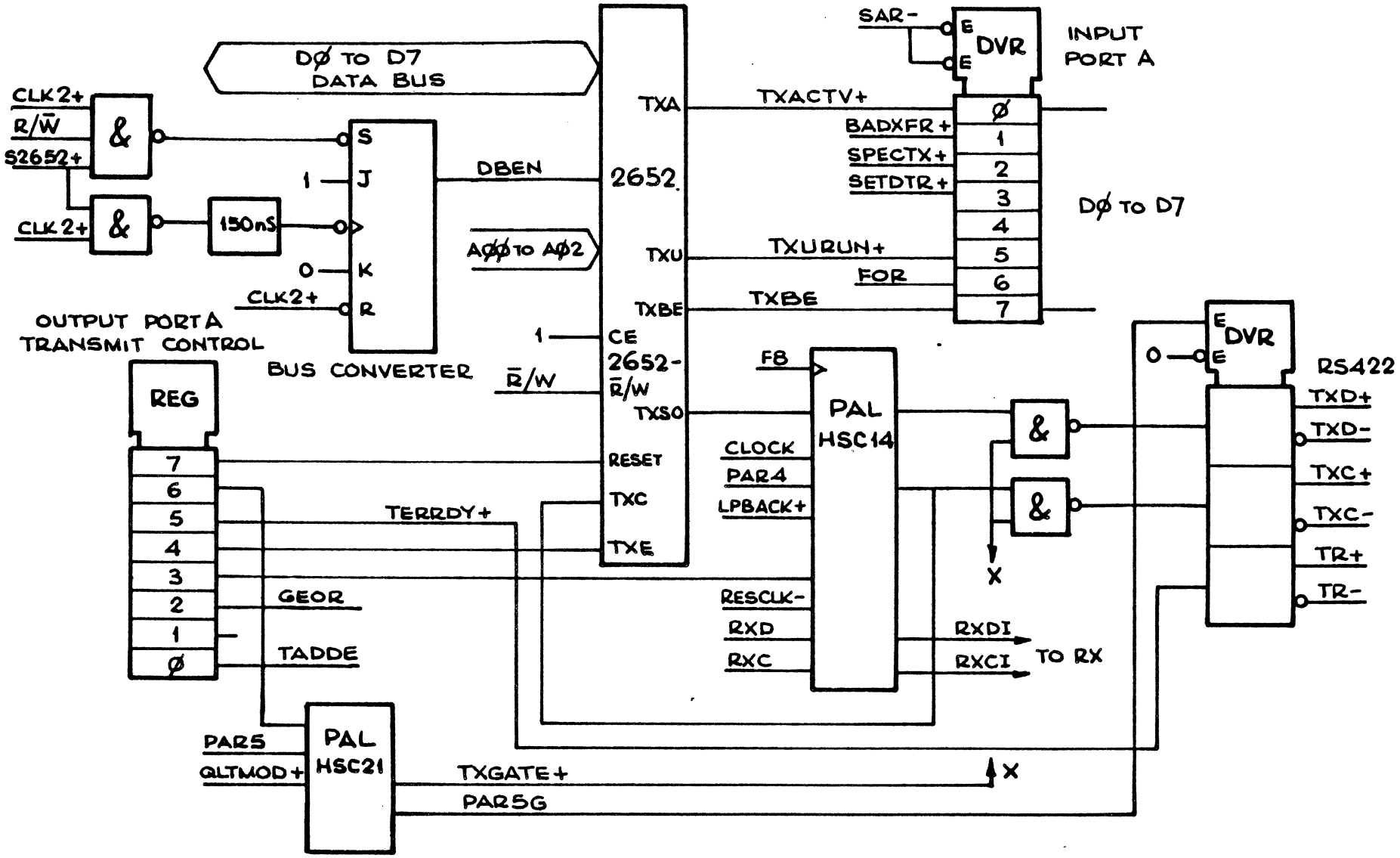


FIGURE 3-26 TX COMMUNICATIONS CONTROLLER

3.9.12 Serial Data Control (Fig. 3-26)

The serial output from the 2652 is routed to PAL HSC14, where it is resynchronised, and if required, converted to pulse width modulation (2MHz only - controlled by an output from O/P port A).

The serial output from HSC14, together with the transmit clock are gated with TXGATE+ and converted to RS422 levels by a 26LS31. A third section of the 26LS31 converts the terminal ready signal (TERRDY+) to RS422.

TXGATE+ is held false during QLT time by HSC21, leaving the data and clock marking. The 26LS31 can be tri-stated by means of PAR5G.

An additional function of HSC14 is to loop back the data and clock during QLT and diagnostic testing. For QLT's this is controlled by an output from O/P port A, while LPBACK+ performs this role under software control.

3.9.13 I.C.C.U. (R/W Port 9) (Fig. 3-27)

This tri-state register provides a data path from the TX micro to the RX micro. It is loaded under control of write decode SCW-, which also sets flop TMM. TMM is an interrupt and status input to the RX micro, signalling that data have been loaded in the ICCU. The RX micro reads the data by driving SCR-B low, enabling the tri-state outputs. This also clocks TMM to the reset state.

A similar register provides the reverse capability from RX micro to TX micro, with flop RMM being set to indicate data being loaded. RMM is an interrupt and status bit to the TX micro. The TX micro reads the ICCU by driving SCR - low, which puts the data on its bus and resets RMM.

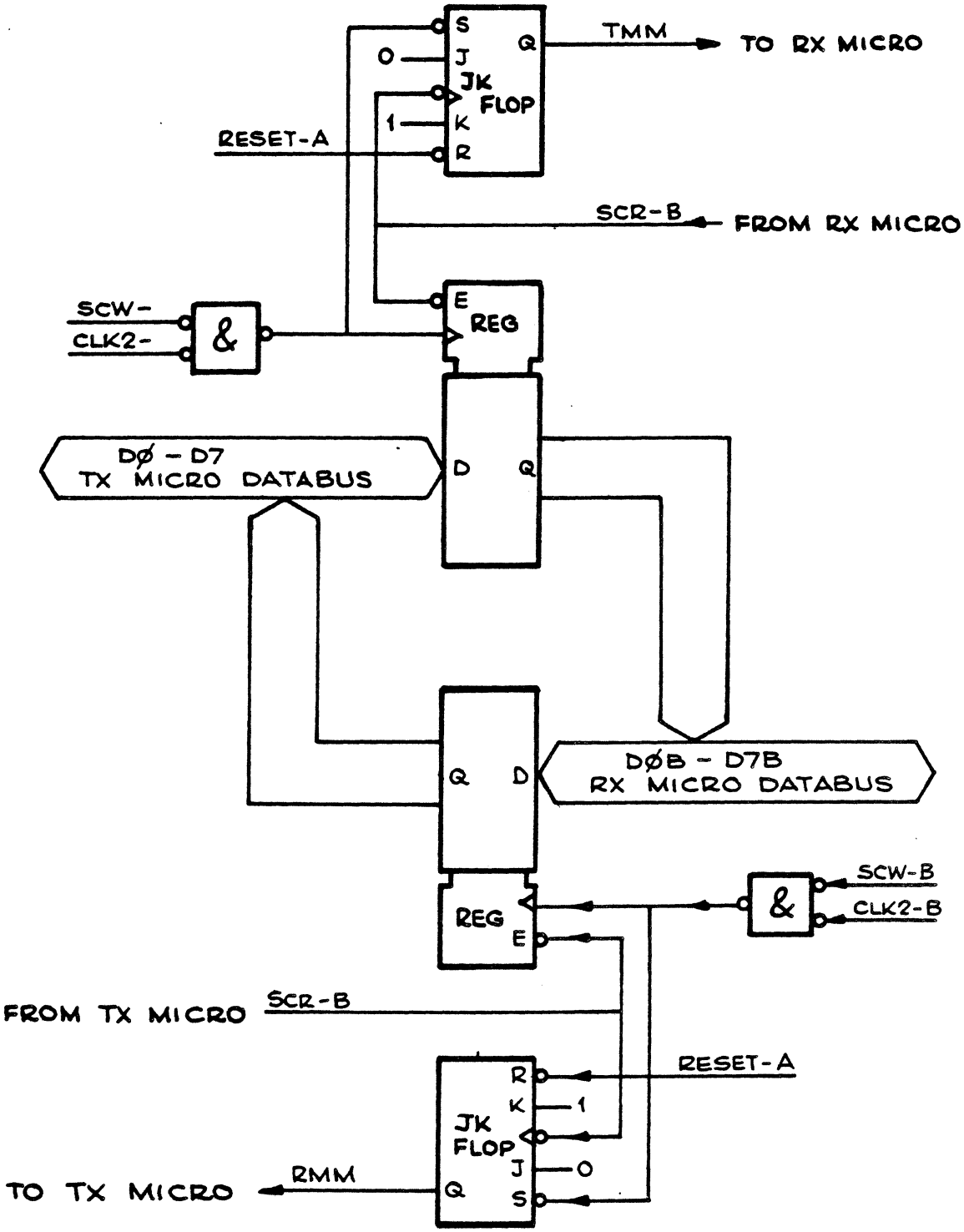


FIGURE 3-27 ICCU

3.9.14 TX FIFO (Fig. 3-28)

This 10 bit wide by 16 word deep FIFO provides the interface between the transmit DMA logic and the TX micro.

When a DMA transfer starts, successive bytes of information are loaded into the FIFO and propagate to its output, resulting in the 'Output Ready' signals going true. HSC-16 ANDs these together, generating FOR, which is a status input to read port A. The TX micro reads a byte from the FIFO by pulsing SFR-low, which enables the FIFO tri-state output. The back edge of SFR- clocks out the byte, allowing the next byte to propagate to the output. The 'Output Ready' signals will go false until this has happened.

In 2MHz operation, the firmware does not have time to check FOR, and assumes that it will be true. Whenever it is ready to accept a byte, PAL HSC 16 monitors Output Readies at SFR- time, and sets FFE (FIFO empty) if data is read when the FIFO is not ready. This signal interrupts the 6502B. The result will be an aborted frame and an under-run condition. Pulse output CRFFE is used to reset FFE.

The last byte of a normal transfer is signalled by TERM being true, which via PAL HSC16 sets EOR true, setting the overflow flag of the 6502B to terminate the firmware loop (see section 4).

If an abnormal termination is forced (BADXFR, FFE or TXU) the FIFO must be cleared, which is performed by CRFIFO.

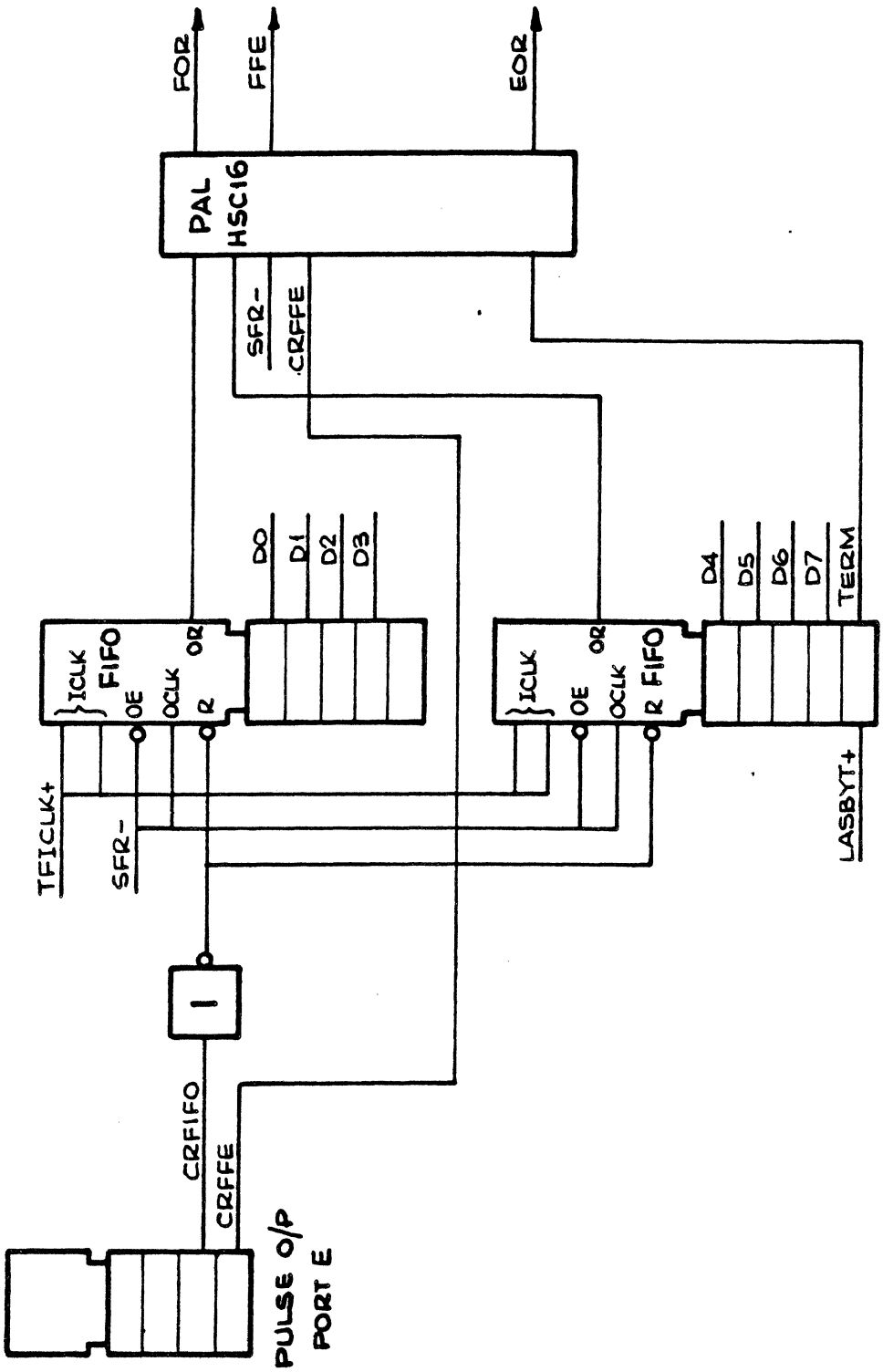


FIGURE 3-28 TX FIFO

3.10 RECEIVER MICROPROCESSOR SYSTEM

This system provides the means for receiving serial data, converting it to parallel byte-wide information which is passed to the receive FIFO.

The RX microprocessor system is composed of the following parts as shown on the block diagram Figure 3-29 .

a) Clock Generator
b) 6502B Microprocessor
c) Address Decoder
d) Read-only Memory (EPROM)
e) Read-write Memory (RAM)
f) Interrupt Control
g) Input Ports
h) Output Ports
i) Pulse Generator Port
j) Communications Controller
k) ICCU
l) Receive FIFO

Since many aspects are similar to the TX microprocessor system, reference will be made to relevant subsections of 3.9 for those similar features.

3.10.1 Clock Generator

This is essentially the same as described in 3.9.1 for the transmit micro, the 24MHz oscillator and 8MHz divider being common. The main difference is that the clock is suspended by RXDAB from the receiver Comms Controller instead of TXBE. The reset logic is common to both systems.

3.10.2 Microprocessor

Like the TX system a 6502B is used as described in 3.9.2. and with the same address space map (Fig. 3.21) However, the peripheral port assignments differ in detail and are shown in Table 3-19.

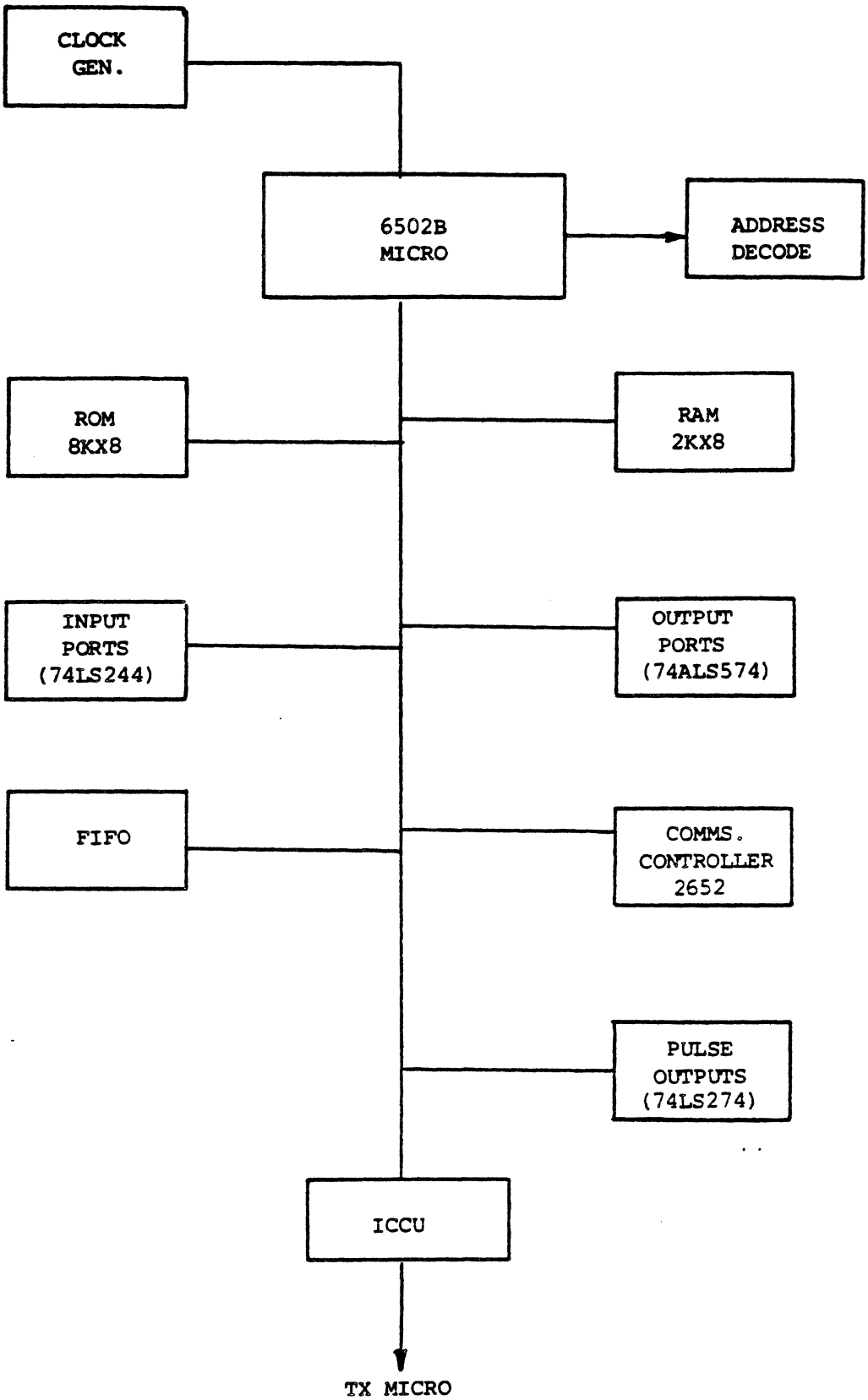


FIGURE 3-29 RX MICROPROCESSOR SYSTEM

TABLE 3-19 RX MICRO PERIPHERAL PORTS

PORT NO.		BIT NUMBER 7	6	5	4	3	2	1	0
8	R								
	W								
9	R								
	W								
A	R	LCON	NCKB	RXCBSY	RXDA	RXSA	S/F	RXACTV+	FIR
	W	RESET 2652	SRFISM	RXE	SFRISL		LED4-	LED5-	TADDEB
B	R	RMM	TMM		FFF	---BAUD RATE SWITCH ---			
	W								
C	R	---------- MESSAGE FROM TX MICRO ----------							
	W	---------- MESSAGE TO TX MICRO ----------							
D	R								
	W	---------- INTERRUPT MASK REGISTER ----------							
		NCKB	TMM	RXSAB	RXDAB	RXCBSY	LCON		FFE
E	R								
	W	---------- PULSED RESETS ----------							
			CRNOVTB	CRFIFB	CRFFF				
F	R								
	W	---------- DATA TO RX FIFO ----------							

R = Read W = Write

3.10.3 Address Decoder

This is essentially as described in 3.9.3 and Figure 3-22.

3.10.4 Read Only Memory (ROM)

This is as described in 3.9.4.

3.10.5 Read-Write Memory (RAM)

This is as described in 3.9.5.

3.10.6 Interrupt Control

This is essentially as described in 3.9.6., except that the interrupt assignments are as in Table 3-20.

TABLE 3-20 RX INTERRUPTS

Mask Bit	Interrupt Signal	Description
0	FFF	FIFO Full
1	-	Not used
2	LCON	Data mode from RS422 I/F
3	RXCBSY+	Receive Busy - Receive DMA started
4	RXDAB	Receiver Data Available - from 2652
5	RXSAB	Receiver Status Available from 2652
6	TMM	TX micro message available
7	NCKB	No - (or slow) receiver clock

3.10.7 Input Ports

Two 74LS244 tristate buffers are used as general input ports to allow the RX micro to read various status bits. These are described in Tables 3-21 and 3-22 and are enabled by SAR-B and SBR-B respectively from the read address decoder.

TABLE 3-21 G.P. INPUT PORT A

BIT NO.	SIGNAL	DESCRIPTION	SOURCE
0	FIR	RX FIFO Input Ready	FIFO
1	RXACTV+	Receiver Active (Frame being received)	2652
2	S/F	Sync/Flag signal from	2652
3	RXSA	Receiver Status Available	2652
4	RXDA	Receiver Data Available	2652
5	RXCBSY+	Receive Busy (DMA in progress)	u-seq
6	NCKB	No (or slow) receiver clock	
7	LCON	Data mode signal from RS422 I/F	

TABLE 3-22 G.P. INPUT PORT B

BIT NO.	SIGNAL	DESCRIPTION	SOURCE
0	BRC	:	
1	BRB	:	
		> Baud Rate Setting	Hex
2	BRA	:	Switch
3	BRD	:	
4	FFF	FIFO Full - micro has attempted to load RX FIFO When not ready	
5	-	Not used	
6	TMM	TX Micro Message available	ICCU
7	RMM	RX Micro Message not yet read by TX	ICCU ICCU

3.10.8 Output Port A

Only one G.P. output port is used on the RX microsystem. This is a 74LS574 and is loaded under control of SAW-B from the write address decoder. The functions of its bits are described in Table 3-23.

TABLE 3-23 G.P. OUTPUT PORT A

BIT NO.	SIGNAL	DESCRIPTION
0	TADDEB	Enables indefinite fetch cycle
1	LED5-	Lights LED DS05 when low
2	LED4-	Lights LED DS04 when low
3	-	Not used
4	SRFISL	Drive status flag bit of RX FIFO
5	RXE	2652 Receive Enable
6	SRFISM	Drives spare bit of RX FIFO (N/U)
7		Reset 2652

3.10.9 Pulse Output Port E

This port operates in the same manner as described in 3.9.9., with HSC15 providing the reset, but is loaded by SEW-B. The assignment of its outputs are shown in table 3.24.

TABLE 3-24 PULSE O/P PORT E

BIT NO.	SIGNAL	DESCRIPTION
0	-	Not used
1	-	Not used
2	-	Not used
3	-	Not used
4	CRFFF	Clear FIFO full Flop
5	CRFIB	Clear RX FIFO
6	CRNOVTB	Clear Runout Condition (Clock Gen)
7	-	Not used

3.10.10 Communications Controller

(Figure 3-30)

A 2652 HDLC Communications Controller is used to receive and assemble serial data. This device is responsible for detection of opening and closing flags, and CRC checking.

Only the receiver portion is used.

The device has 8 internal registers, which are addressed by A00 through A02 under control of DBEN and R/W. Since the 2652 bus timing is not compatible with the usual 6502B bus timing, a converter circuit is used to generate DBEN from the address decode S2652-B and CLK2. This works identically to that described in 3-9-11.

The 2652 is reset by an output from output port A, and reception is enabled by a second output. Four status outputs from the 2652 are read by Input port A, as follows.

RXACTV+	-	Receiver Active (frame in progress)
S/F	-	Sync/Flag indicator
RXSA	-	Receiver Status available
RXRA	-	Receiver Data available

3.10.11 Serial Data Control (Fig. 3-30)

The serial data and clock are received from the line by a 26LS33, which converts from RS422 to TTL levels. A third section of 26LS33 converts the 'Data Mode' signal to LCON which is input to the micro via port A.

The data and clock are input to PAL HSC14. Normally they are output as RXDI and RXCI by HSC14, except when loopback is required.

A 2 I/P multiplexer is used to select the source of receive clock and receive data to be input to the communications controller.

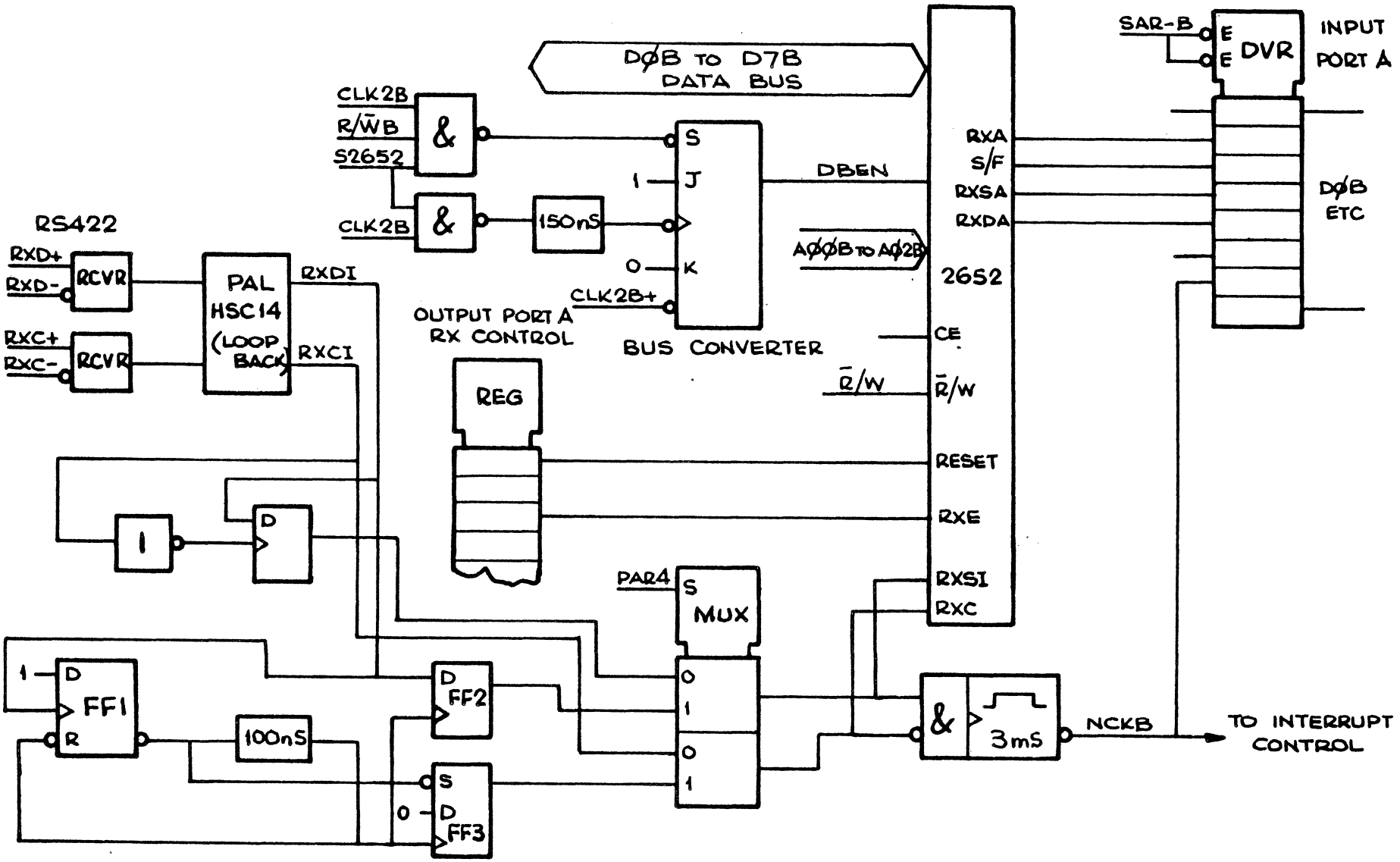


FIGURE 3-30 RX COMMUNICATIONS CONTROLLER

Below 2MHz, (PAR4 false) RXCI is directly routed to the 2652 RXC input, while RXDI after resynchronising to RXCI is routed to the serial data input.

At 2MHz (PAR4 true), when pulse width modulation is used, the clock and data must be recovered from RXDI alone. This performed as follows.

The rising edge of RXDI sets FF1. This causes a low-going edge 100ns later at the output of the delay line, resetting FF1. A positive edge now propagates down the delay line, clocking FF2 100ns later (i.e. approximately 200ns after the rising edge of RXDI). If RXDI is still true, FF2 will set, otherwise it will reset.

Thus FF2 represents the recovered data. Clock is recovered by FF3, which is set via FF1 on the rising edge of RXDI, and reset 200ns later by the rising edge from the delay line.

The clock and data output from the multiplexer are applied to the inputs of a one-shot. The one-shot will be retriggered every time a zero data bit is received with a suitable clock pulse. Thus the Q output (NCKB) will go high if the line has stayed in the marking (or abort) condition, OR if no clock has been received, for a period of about 3ms.

3.10.12 ICCU

This has been described in 3.9.12.

3.10.13 Receive FIFO (Fig. 3-31)

The receive FIFO is the interface between the receive micro system and the receive DMA logic.

When the FIFOs are able to accept data, their "Input Ready" signals are true. These two signals are ANDed by PAL HSC-16 producing FIR which the RX micro can test prior to loading a byte into the FIFO.

This it does by pulsing SFW-B low (from the write-address decoder) which loads the data into a temporary register. The register clock signal is buffered by HSC-16, the buffered signal driving the FIFO input clocks, and loading the data into the FIFO from the holding register. The HSC16 introduces sufficient delay in the clock signal to meet the set-up requirements of the FIFO.

When operating at 2MBps, the RX micro does not have time to test FIR, and so attempts to load the FIFO whenever it has data to send. PAL HSC-16 monitors the clock and ready signals, and in the event of an attempt to load the FIFO when it is not ready, FFF (FIFO full) is set, which interrupts the RX micro. The result will be a receive data rate error. The condition is reset by CRFFF from pulse output port E.

At the end of a frame, or in the event of an error the RX micro prepares a status byte which it loads into the FIFO having first set SRFISL true via output port A. This flags the end of transfer to the microsequencer.

The receive FIFO is cleared by CRFIFB from the pulse output port. This is normally only necessary at RESET time, but if the incoming frame is longer than the buffer, the residual data bytes must be cleared from the FIFO so that the status byte may be passed.

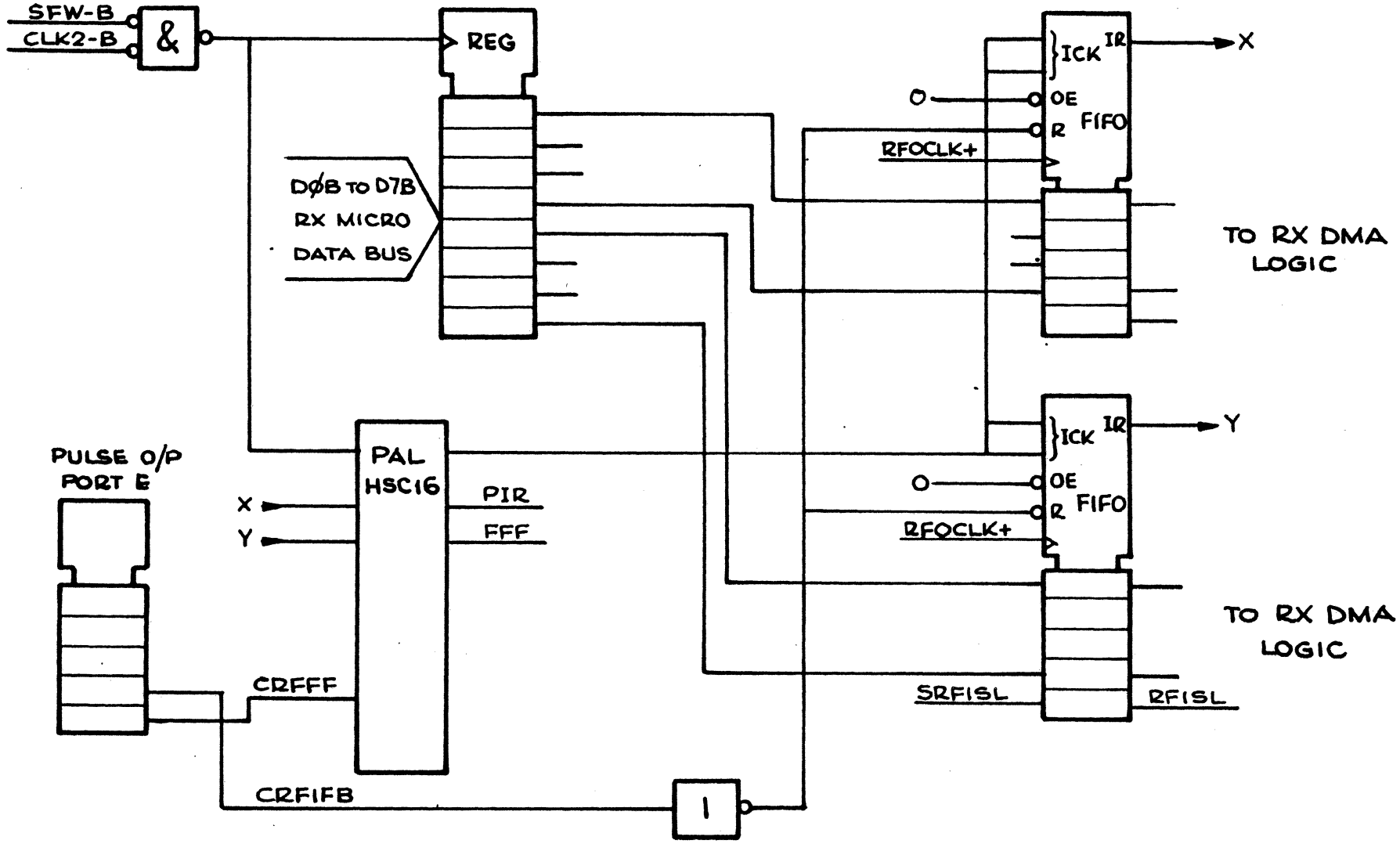


FIGURE 3-31 RX FIFO

4 FIRMWARE

4.1 OVERVIEW

Firmware flow charts are given in figures 4.1. to 4.7. The data handling sections of the firmware are run in RAM because this enables the 6502B to run at full speed. The access time of available EPROM does not permit this.

Subroutines are accessed via a set of jumps starting at location F000. This means that should the start location of a subroutine be changed, only one jump instruction need be modified.

After initialisation, execution starts at location F100.

4.2 INDEFINITE FETCH CYCLE

At Baud rates below 2MB the data is moved between FIFO and 2652 under program control with the program testing the readiness of the FIFO and 2652 via peripheral ports. However, at 2MB this is not possible and another mechanism has to be employed as follows:-

The basic loop for the transmitter is:

```
-->  LDX  0F    -  Load X reg. from FIFO
|    STX  02    -  Output X reg. contents to 2652
|    BVC *-2    -  branch if not end
+--
```

If the FIFO does not have a byte available, then the load instruction will cause a FIFO full error condition.

The whole data handling operation must be matched to the comms. line rate via the 2652, and this is accomplished by the indefinite fetch cycle. The output instruction (STX 02) takes three clock cycles.

1. Fetch op code
2. Fetch address (02)
3. Execute transfer from X reg.

The address is arranged to be in location 240.

When the 6502 attempts to access this location, its clock cycle will be extended until TXBE, the buffer empty signal from the 2652, goes true. When this happens the clock cycle is allowed to complete and the 6502 proceeds to output the data byte. Note that the same technique is used at location 248 to set TEOM at the end of a frame.

The same technique is used for the receiver in its 2MB loop, but location 248 is not used.

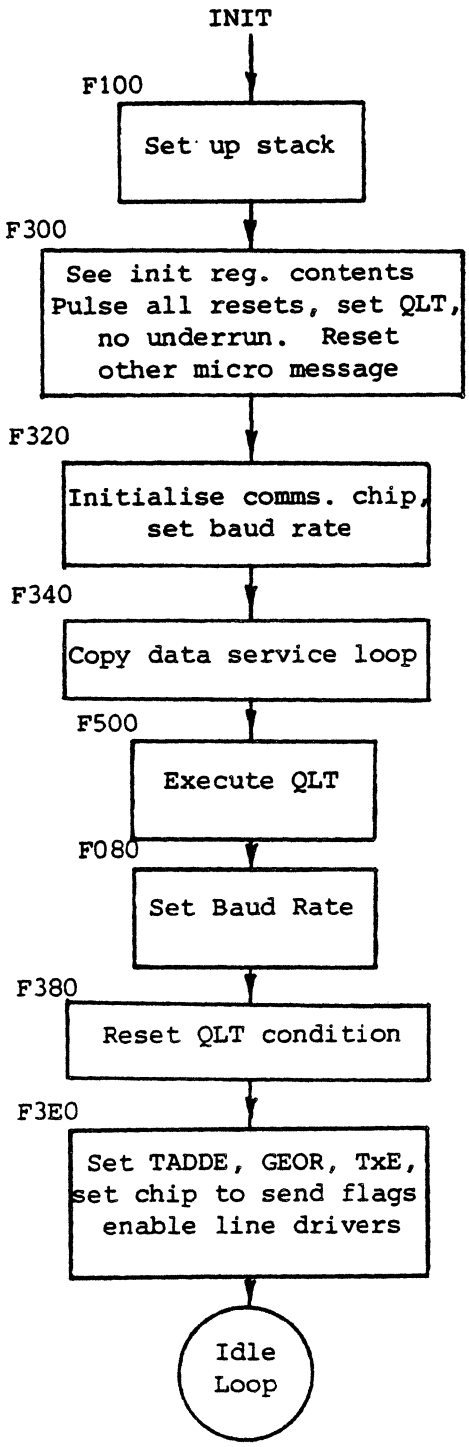


Figure 4-1 TX Initialisation

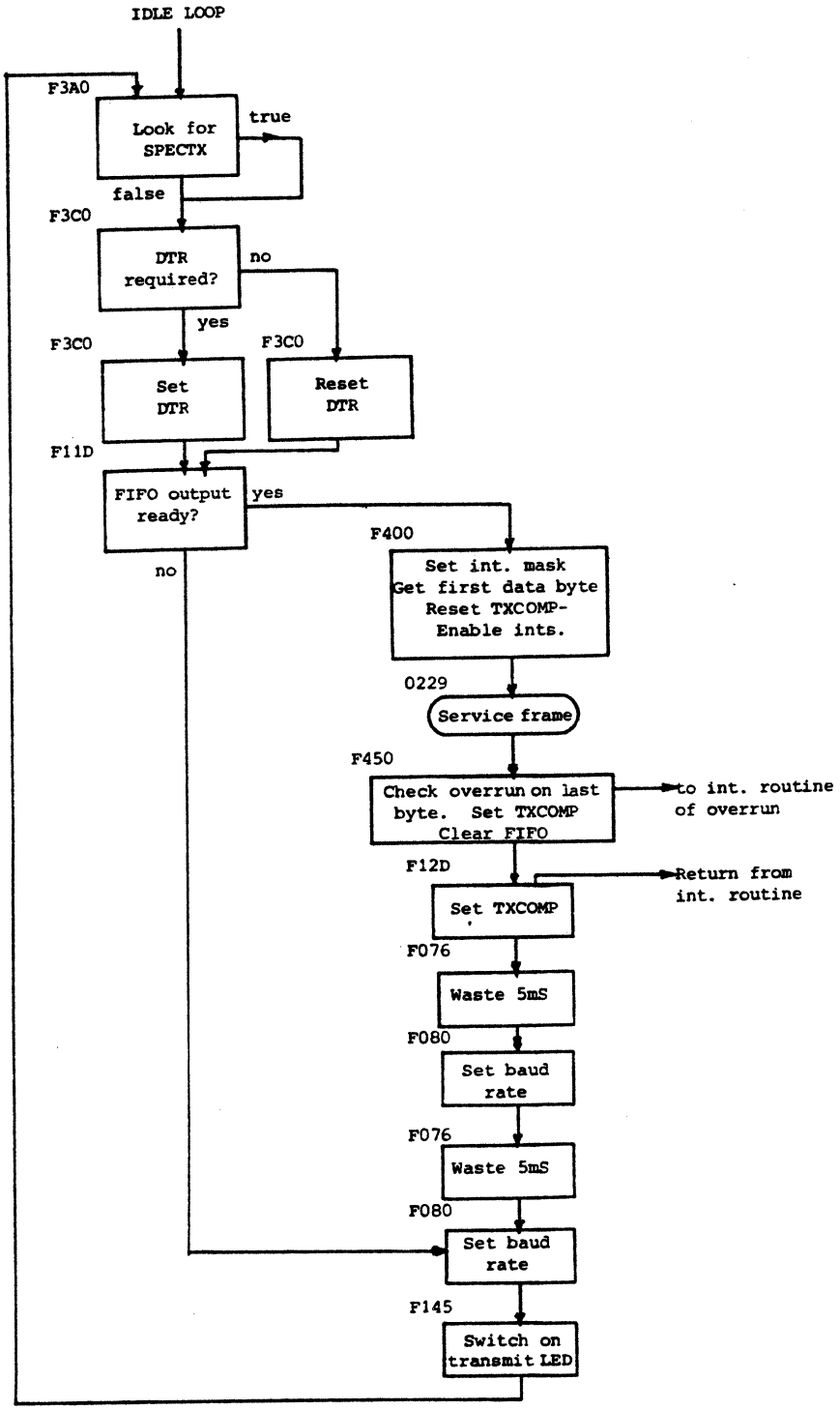


FIGURE 4-2 TX IDLE LOOP

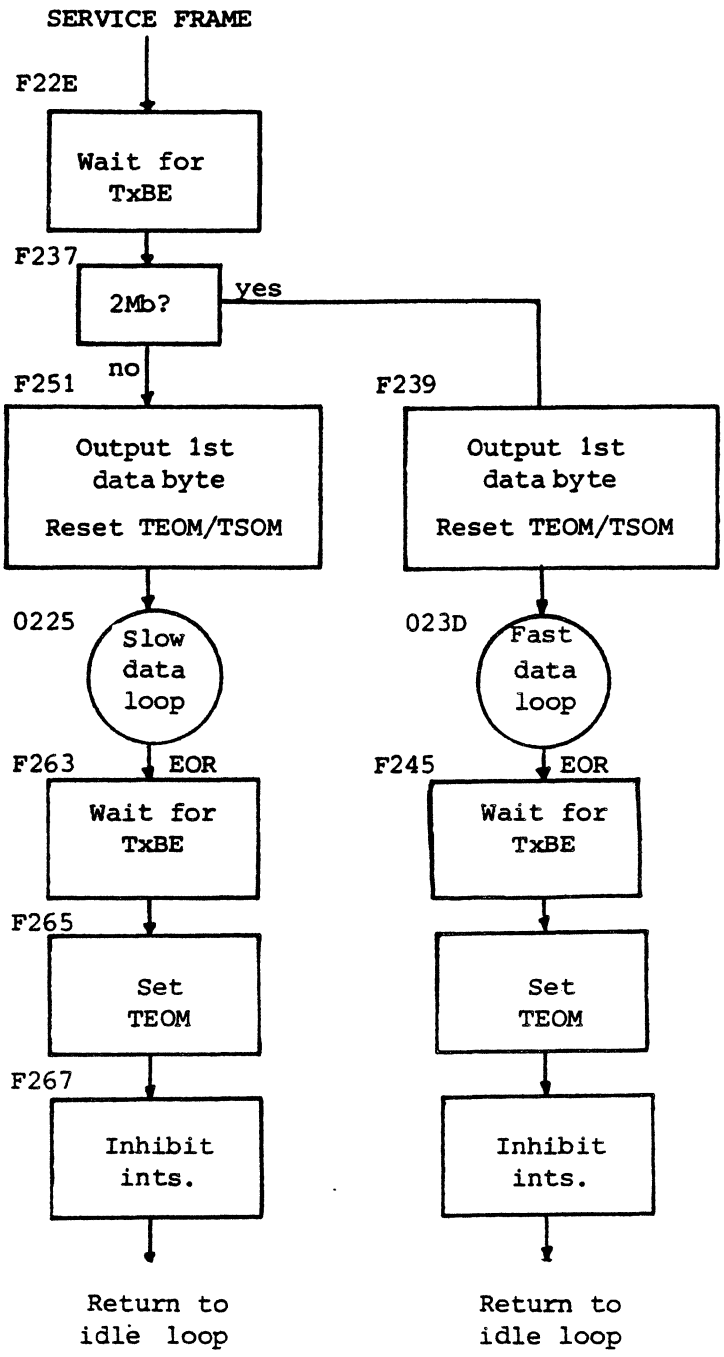


Figure 4-3 TX Data Service

Possible ints:-

BADXFR, TxU, FIFO empty
RNOUT

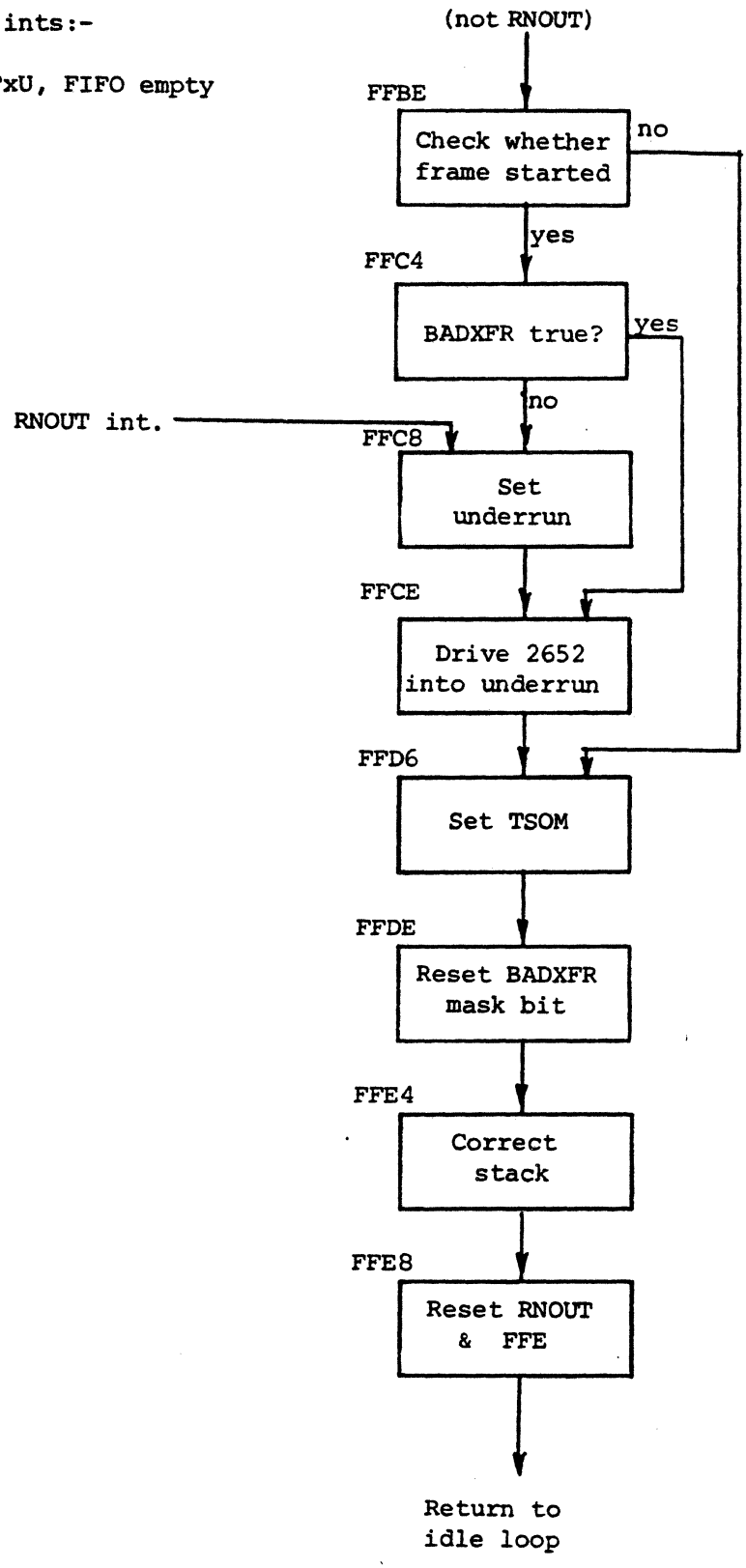
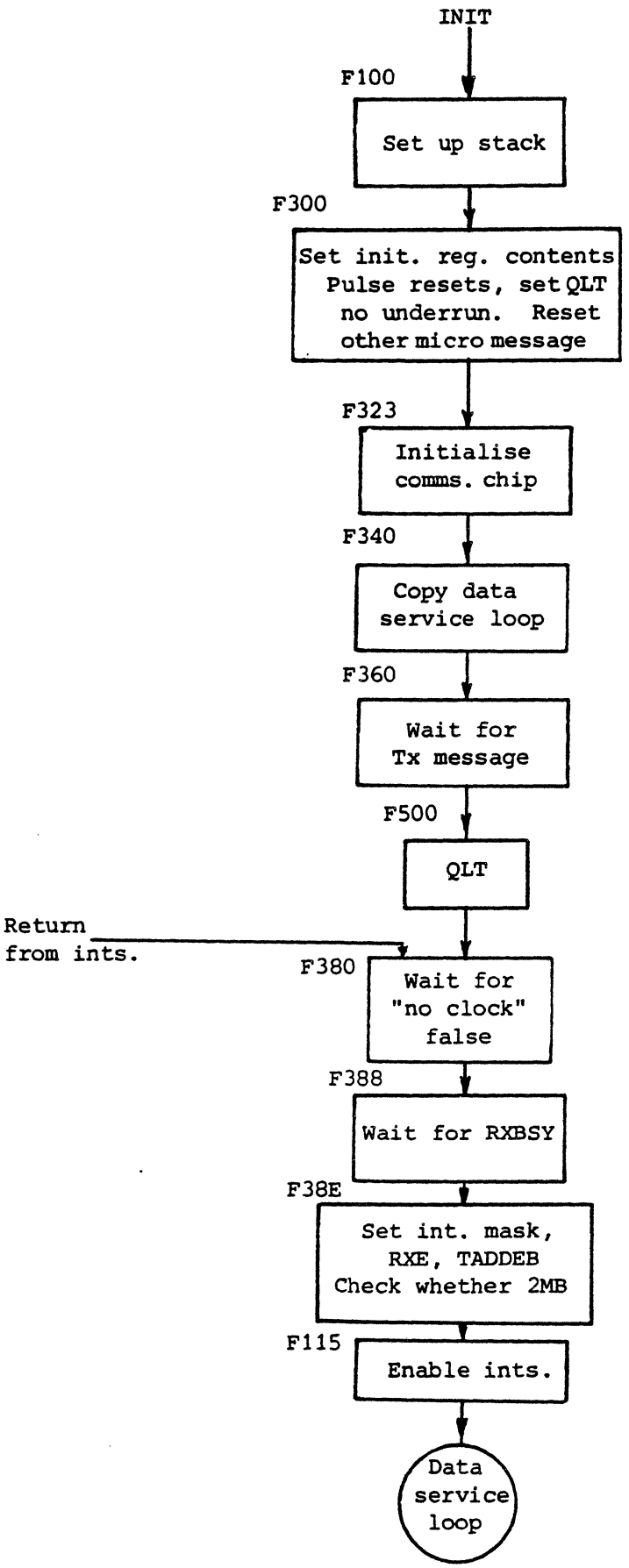


Figure 4-4 TX Interrupt Service



4-5 RX Initialisation

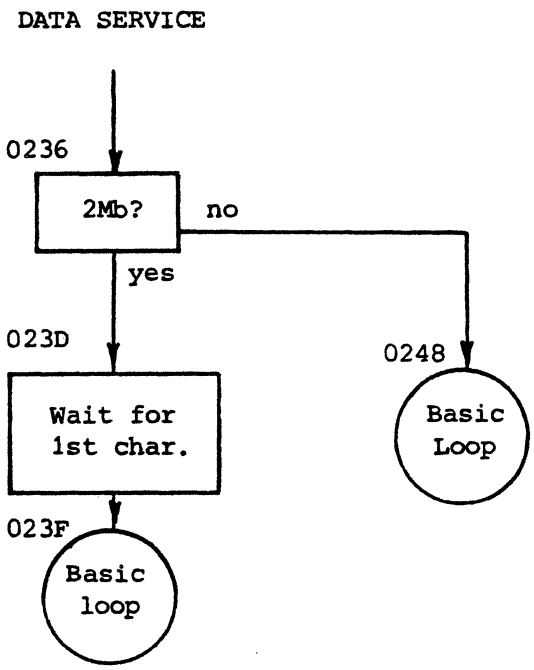


Figure 4-6 RX Data Service

Possible ints:-

RxSA, no clock,
RNOUT, RxBSY, FFF.

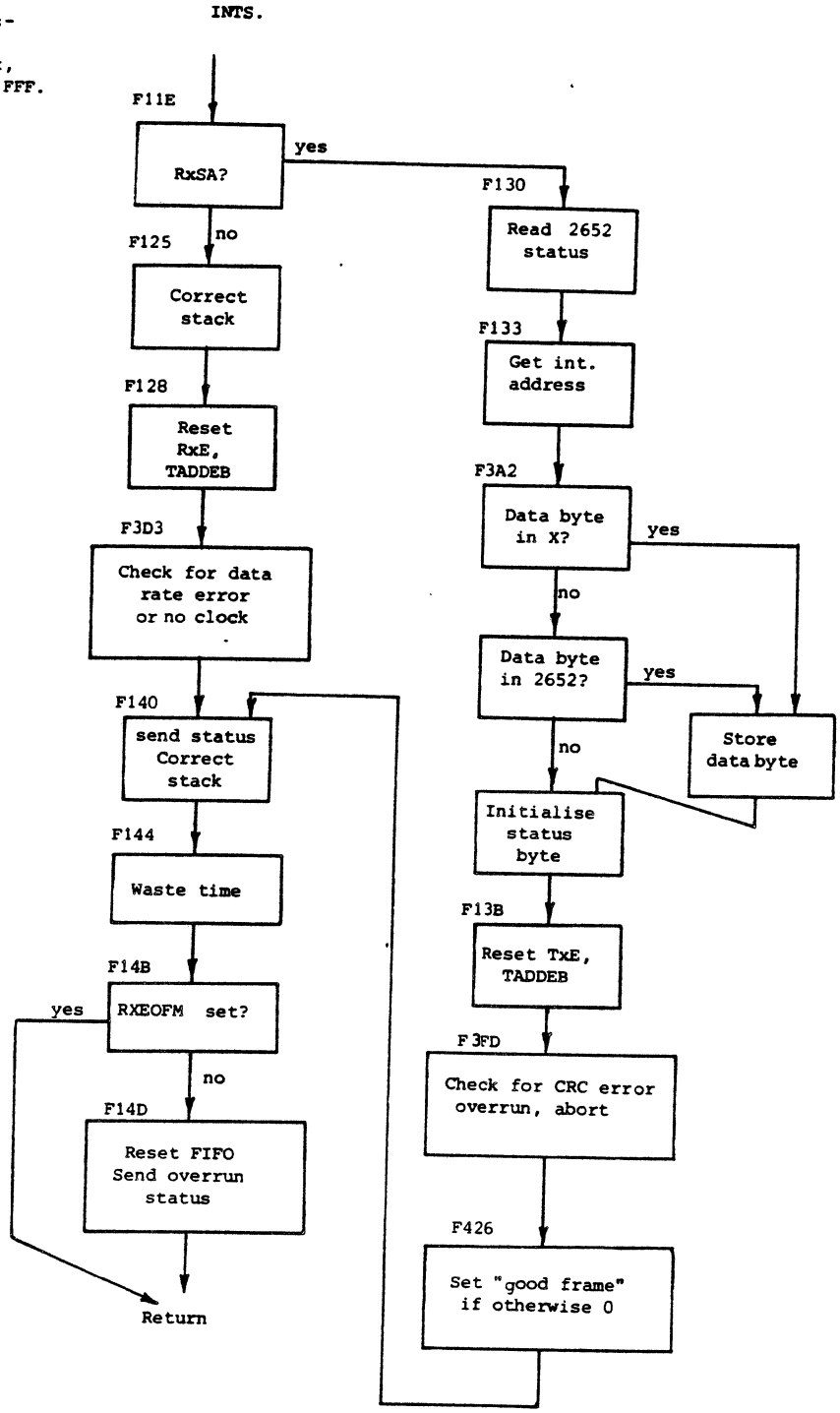


FIGURE 4-7 RX INTERRUPT SERVICE

CUT ALONG LI

HONEYWELL INFORMATION SYSTEMS LIMITED

Technical Publications Remarks Form

TITLE	DPS 6/LEVEL 6 HIGH SPEED COMMUNICATIONS MANUAL	**ORDER NO.**	YM73-00
		DATED	April 1985

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be promptly investigated by appropriate technical personnel and action will be taken as required. If you require a written reply, check here and give complete mailing address below. ☐

FROM: NAME _____ **DATE** _____

TITLE _____

COMPANY _____

ADDRESS _____

fold 3

Postage will be paid by licensee

Do not affix Postage Stamps if posted in Gt. Britain, Channel Islands or N. Ireland

BUSINESS REPLY SERVICE
NW 2787

DOCUMENTATION CENTRE
HONEYWELL INFORMATION SYSTEMS LTD
10 CULLEN WAY
LONDON
NW10 6JZ

2

fold 4 and tuck in

fold 2

fold 1

CUT ALONG LINE

Together, we can find the answers.

Honeywell