EK-DMF32-TD-001

DMF32 Multi-Function Communications Interface Technical Description

digital

EK-DMF32-TD-001

DMF32 Multi-Function Communications Interface Technical Description

Prepared by Educational Services
of
Digital Equipment Corporation

Printed in U.S.A.

The manuscript for this book was created on a DIGITAL Word Processing System and, via a translation program, was automatically typeset on DIGITAL's DECset Integrated Publishing System. Book production was done by Educational Services Development and Publishing in South Lawrence, MA.

CONTENTS

FIGURES

TABLES

CHAPTER 1
INTRODUCTION

1.1 SCOPE

This manual describes the DMF32 hardware operation in detail. Besides the detailed hardware description, a higher-level flow description of the microinstruction fields and the microprogram are also provided. The microinstruction fields control the hardware, while the microprogram determines the functionality of the DMF32 module, as seen by the system software. Additionally, a detailed description of the power-up self test is included.

For a reader to obtain full benefit from this manual, the reader should have an understanding of the following:

- DMF32 functionality (described in the *DMF32 User's Guide*)
- DIGITAL logic
- Microprogramming
- UNIBUS concepts
- Program Array Logic (PAL)
- 2901 ALU
- 2911 microprogram sequencer
- 2661-3 Universal Asynchronous Receiver Transmitter (UART)
- 2652 Multi-Protocol Communications Controller (MPCC).

1.2 DMF32 DOCUMENTATION

Table 1-1 lists other DMF32 documentation.

Table 1-1 DMF32 Documentation

Document Title	Document Number
DMF32 User's Guide	EK-DMF32-UG-001
DMF32 Field Service Print Set	MP01271
DMF32 Maintenance Advisory	to be supplied

For information concerning microfiche libraries, contact:

Digital Equipment Corporation
Micropublishing Group, BU/D2
12 Crosby Drive
Bedford, MA 01730

Manuals in hard copy may be obtained from:

Digital Equipment Corporation
Accessories and Supplies Group
Cotton Road
Nashua, NH 03060

Attention:	Documentation Products
Telephone:	1-800-258-1710

1.3 PHYSICAL DESCRIPTION

The DMF32 consists of a single hex-size peripheral controller module, an 8.25-inch × 4-inch distribution panel, and three 40-pin shielded BC06R flat cables. The three BC06R cables connect the single hex M8396 module to the distribution panel via standard Berg connectors. The M8396 module can be electrically installed in any VAX family hex-height SPC slot. Refer to Figure 1-1 for the DMF32 components.

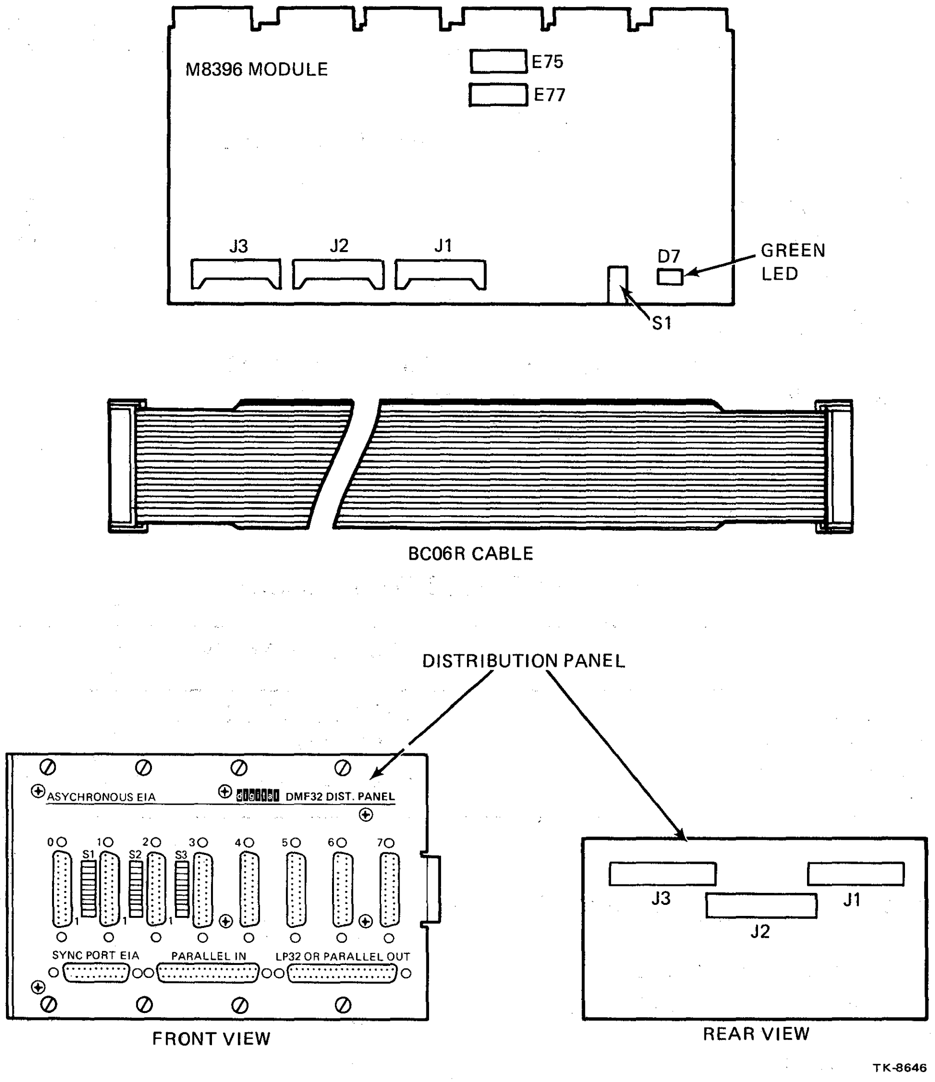


Figure 1-1 DMF32 Components

The M8396 module has one toggle switch, two 10-position dip switch packs, and a green LED. The S1 toggle switch is used to single step microword instructions. The two dip-switch packs are used for DMF32 parameters. Dip-switch pack E75 determines the DMF32 UNIBUS address (switches 1 through 8) and controls the DMA power-up self test (switches 9 and 10). Dip-switch pack E77 controls UNIBUS INIT (switch 1), determines device priority (switches 2 through 9), and enables the single step mode (switch 10). Refer to the *DMF32 User's Guide* for the switch setting configurations. The green LED indicates if the M8396 module is operational. A parity error extinguishes the green LED.

The DMF32 distribution panel has three 10-position dip-switch packs and eleven Cinch connectors. The three 10-position switch packs are used for modem interface configurations and DMF32 device selection. Refer to the *DMF32 User's Guide* for the switch setting configurations. The distribution panel has the following connectors:

- Eight 25-pin Cinch connectors for eight RS232-C asynchronous lines
- One 25-pin Cinch connector for one RS232-C synchronous line
- Two 37-pin "D" type connectors for either a line printer or a special user's device.

1.4 FUNCTIONAL DESCRIPTION

The DMF32 is a multifunction module which incorporates an eight-line asynchronous multiplexer, a synchronous controller, and a parallel interface (line printer or user device). These three devices are hardware dependent, since all three devices are supported by the same 2901 microprocessor. Functionally, the three devices are independent. This functional independence is achieved by partitioning the CSR and vector space to independent microprocesses which allow simultaneous control by up to three independent software drivers.

1.4.1 Asynchronous Multiplexer

The asynchronous multiplexer, an enhanced version of the DZ11-A, supports eight transmit and eight receive lines. Each pair of lines (one transmit and one receive) can be programmed to operate at one time of 16 baud rates ranging from 50 bps to 19.2 Kbps. Both line 0 and line 1 have split speed capability and full modem control. The asynchronous multiplexer also supports the echo function.

Transmission can be selected for DMA or SILO operation. In SILO mode, each line transmits characters from its own 32-character buffer. These buffers are loaded under host software control. In the DMA mode, a transmit line transmits a character from the main memory location that is specified by a buffer address and character count.

All eight lines share a 48-character receive silo. There is a programmable silo timeout period for the receive silo.

An interrupt can be generated under one of the following conditions:

- Sixteen characters have entered the silo.
- The silo has been non-empty for a time greater than a programmable timeout period. This timeout period can be set to zero.

The asynchronous lines are connected either to data terminal equipment (DTE) or data communication equipment (DCE) via standard EIA RS-232-C 25-pin connectors.

1.4.2 Synchronous Interface
The synchronous interface is a single-line DMA communications device that has full modem control (EIA RS-232-C/CCITT-V.24). The DMA transfers are double buffered; that is, both the transmitter and receiver have two sets of byte count and buffer address registers.

The synchronous interface supports various bit-oriented protocols (e.g., SDLC and HDLC) and byte-oriented protocols (e.g., DDCMP). The synchronous line can frame the messages, generate and check CRC, and DMA these messages to and from host memory. The host-level software performs all message acknowledgments and higher-level network functions.

Running the GEN BYTE protocol (general byte-oriented synchronous) allows the synchronous interface to implement any byte-oriented protocol. The GEN BYTE protocol uses a straight transfer of data between main memory and synchronous interface. The host-level software handles the protocol-specific functions.

The synchronous interface has full modem control. The modem lines conform to EIA RS-232-C/CCITT-V.24 specifications for speeds up to 19.2 Kbps. The synchronous interface is connected to data terminal equipment (DTE) or data communications equipment (DCE) via a standard 25-pin Cinch connector.

When using the DMF32 crystal-controlled baud rate generator, the synchronous line can transmit at one of sixteen different programmable speeds. With external clocking, any transmit or receive bit rate up to 19.2 Kbps can be used.

1.4.3 Line Printer Controller
The line printer controller interfaces with the LP32 family of printers (LP25, LP26, and LP07). This DMA device can optionally perform the following low-level formatting functions:

- Tab expansion
- Auto carriage return insertion
- Auto line wrap
- Auto form feed to multiple-line feed conversion
- DAFVU support.

1.4.4 Parallel Interface
The parallel interface is an enhanced version of the DR11-C. This device is not only functionally compatible with the DR11-C, but also supports the SILO mode (half duplex) and the double-buffered DMA mode (half duplex). In the DR11-C functionality mode, the parallel interface performs similarly to a DR11-C. After a UNIBUS INIT, the parallel interface emulates a DR11-C. To operate in either SILO or DMA mode, the software device driver must set mode bits in a parallel interface's miscellaneous register.

1.5 POWER-UP SELF TEST
The DMF32 executes a power-up self test upon power-up or UNIBUS INIT. The self test is performed before the operating variables of the devices are initialized and device operation is initiated. Upon successful completion of the self test, AA (hex) is loaded into DMF32 CSR1 bits ⟨15:8⟩. The self test checks the following hardware elements and ascertains that:

- 2901 ALU can perform computations correctly.
- 2901 A and B registers can be addressed properly.
- Condition codes can be set properly.
- The local store RAM is operational.
- The micro PC stack functions correctly to four levels of subroutine call.
- The UNIBUS slave trap hardware functions correctly.
- The UNIBUS master I/O addressing and data transfers function correctly.

1.6 CONTROL STATUS REGISTER (CSR) AND VECTOR ASSIGNMENTS

The floating control status registers for the four devices (asynchronous multiplexer, synchronous interface, line printer controller, and parallel interface) reside in a contiguous block of sixteen words in UNIBUS I/O address space. The addresses of these CSRs are calculated from a base address. This base or starting address is determined by the switch settings of switch pack E75 (switches 1 through 8) on the M8396 module. Refer to Figure 1-2 for the CSR address map.

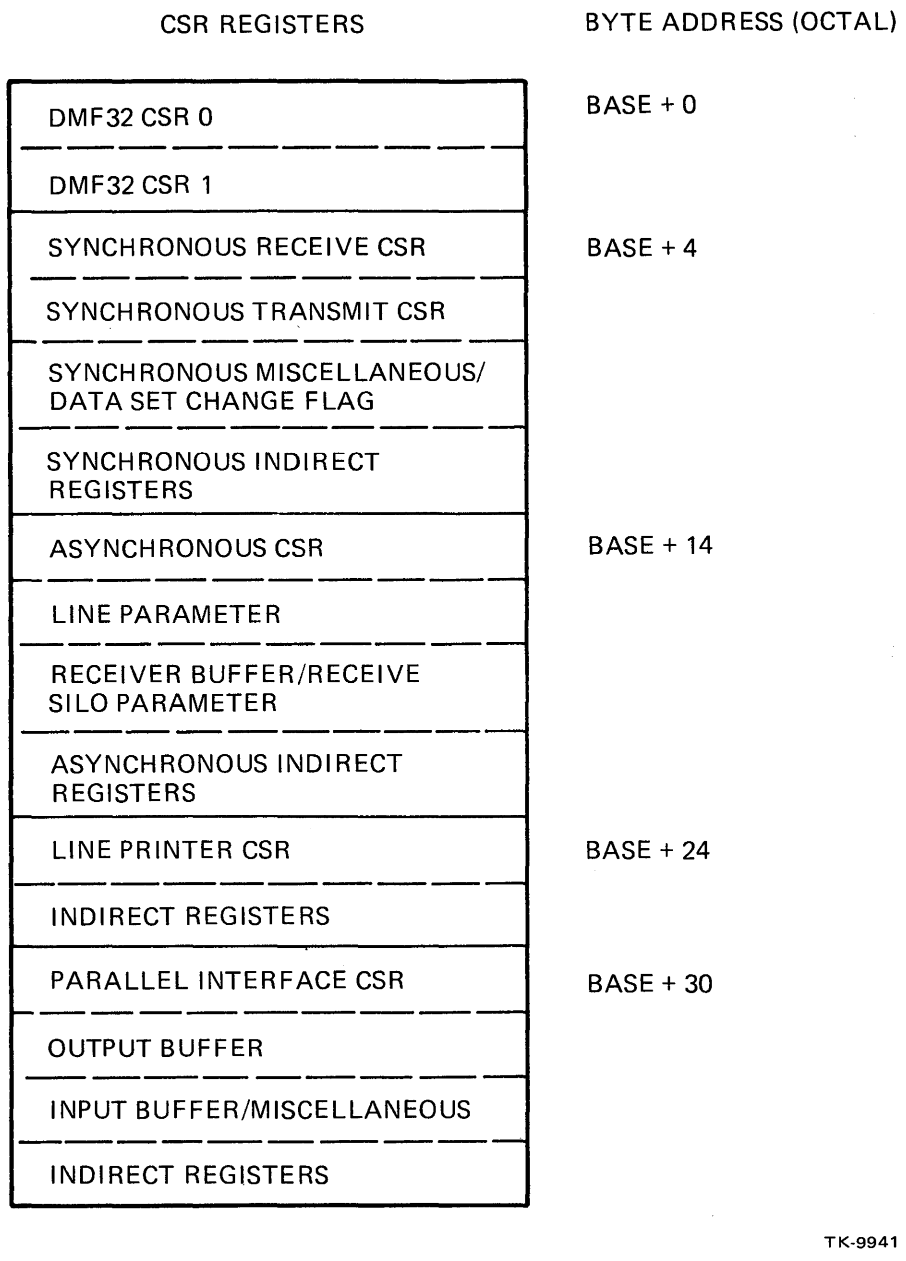


Figure 1-2 CRS Address Map

The DMF32's CSRs are only word accessed, except for the register used to access an asynchronous line's transmit silo and the parallel interface output buffer when operating in the DR-11-C functional mode. Word access means that the instruction which operates on the register is interpreted by the DMF32 as a DATO (data out) rather than a DATOB (data out byte) UNIBUS cycle. Because the DMF32 ignores the least significant UNIBUS address bit on these word-accessed-only registers, a DATOB operation on these registers will be performed as a DATO.

1.6.1 DMF32 CSR 0

At auto-configure time, the operating system uses CSR 0. CSR 0 bits ⟨15:12⟩ contains a four-bit device code that indicates to the operating system which DMF32 devices are available. Refer to Table 1-2 for the device code configurations. Also at auto-configure time, the operating system loads the value of the first vector into CSR 0 bits ⟨7:0⟩. Because there are no switches on the M8396 module for the interrupt vector values, the value of this first vector (VECTOR[0]⟨9:2⟩), which is loaded by the operating system, is used to calculate the value of the other seven interrupt vectors.

Table 1-2 CSR 0 Bits ⟨15:12⟩ Device Codes

CSR 0 Bit (bit = 1)	Device Available
15	synchronous interface
14	asynchronous multiplexer
13	line printer
12	parallel interface

The DMF32's available devices are determined by the switch settings of switch pack 3 (switches 4 and 5) on the DMF32's distribution panel. The DMF32 reads these switches to determine which devices are available. The read value is reflected in CSR 0 bits ⟨15:12⟩. Since the microcode reads these switches only once after power-up, the switches should be set for the selected devices before power-up. Refer to Table 1-3 for switch pack 3 (switches 4 and 5) switch settings.

Table 1-3 DMF32 Device Selection

Switch Pack 3		
Switch 5	**Switch 4**	**Devices Available**
ON	ON	asynchronous
ON	OFF	asynchronous, line printer
OFF	ON	asynchronous, synchronous, parallel interface
OFF	OFF	asynchronous, synchronous, line printer

CSR 0 bits ⟨15:12⟩ may be changed to another valid configuration by writing CSR 0 bits ⟨15:12⟩. For example, a diagnostic program might want to change from parallel interface to line printer or from line printer to parallel interface functionality without human intervention. This may be done by executing a WRITE WORD (e.g., MOVW) instruction to CSR 0 bits ⟨15:12⟩. However, this WRITE WORD instruction will write over the base interrupt vector, that occupies the low byte of CSR 0. To load the interrupt vector (CSR 0 low byte) without affecting the CSR 0 high byte (i.e., device available bits), a BYTE output instruction (e.g., MOVB) should be executed. This MOVB instruction will load the low byte of CSR 0, regardless if the high or low byte is addressed.

The operating system loads CSR 0 bits ⟨15:12⟩ with the value of the first vector VECTOR[0] ⟨9:2⟩. The DMF32 calculates the other seven interrupt vectors form the value of VECTOR[0] ⟨9:2⟩. The DMF32 assumes that the other seven vectors are contiguous to and of greater value than VECTOR[0] ⟨9:2⟩. Refer to Table 1-4 for the vector values.

Table 1-4 DMF32 Floating Interrupt Vectors

Vector	Function	Vector Value (octal)
VECTOR[0]⟨9:0⟩	synchronous interface receive	base(VECTOR[0]⟨9:0⟩)
VECTOR[1]⟨9:0⟩	synchronous interface transmit	base + 4
VECTOR[2]⟨9:0⟩	parallel interface Vector A	base + 10
VECTOR[3]⟨9:0⟩	parallel interface Vector B	base + 14
VECTOR[4]⟨9:0⟩	asynchronous multiplexer receive	base + 20
VECTOR[5]⟨9:0⟩	asynchronous multiplexer transmit	base + 24
VECTOR[6]⟨9:0⟩	line printer controller	base + 30
VECTOR[7]⟨9:0⟩	unused	base + 34

1.6.2 DMF32 CSR 1

The DMF32 CSR 1 is used for diagnostic purposes. There are five different purposes that CSR 1 can be used for. CSR 1 can be used in conjunction with the line printer maintenance mode. When the line printer is in maintenance mode, data is transferred to CSR 1 bits ⟨7:0⟩ instead of being transferred to the line printer. Reading CSR 1 will automatically clear the CSR 1 Bits ⟨7:0⟩. The other four uses of CSR 1 are listed in Table 1-5. The contents of CSR 1 bits ⟨15:8⟩ denotes the function of the register.

Table 1-5 CSR 1 Functions

CSR 1 Bits⟨15:8⟩ Contents(hex)	Diagnostic Function
55	Forces a parity error. A parity error extinguishes the green LED on the M8396 module and also inhibits microcode execution. In this state, DMF32 registers cannot be accessed. To restart execution, UNIBUS signals DC LO or INIT must be asserted.
AA	Starts execution at location 0000. Location 0000 is where execution begins after a UNIBUS DC LO or INIT. This feature allows program-controlled initiation of the power-up self test.
2A	CSR 1 high byte contains the microcode REV level. To read the REV level, 2A (hex) must be written to CSR 1 bits ⟨15:8⟩, then a read of CSR 1 bits ⟨15:8⟩ will obtain the REV level. The REV level is stored in BCD, and, thus, there are two digits in the byte.
AA	The self-test is successfully complete.

1.7 POWER SUPPLY REQUIREMENTS

The DMF32's power requirements are as follows:

- 8.0 amperes @ +5 Vdc
- 0.5 amperes @ +15 Vdc
- 0.5 amperes @ −15 Vdc

CHAPTER 2
DMF32 HARDWARE OVERVIEW

2.1 INTRODUCTION

The M8396 module is an intelligent peripheral controller that interfaces a CPU to terminals, peripheral devices, and other computers. These devices are interfaced to the M8396 module via the following four ports:

1. One UNIBUS interface that can operate as a bus master (for DMA and interrupts) or as a slave.
2. Eight asynchronous lines (EIA/RS232-C voltage levels). Two of the lines have both split baud rate capability and modem control.
3. One asynchronous line (EIA/RS232-C voltage levels) with modem control.
4. One parallel interface (TTL) whose hardware is shared for use as both a general purpose interface, and a parallel line printer interface.

The module's intelligence is contained in a 4K × 36-bit microprogram, that is stored in nine 4K × 4 bipolar PROMs. The microprogram instructs the hardware to perform various functions. The microcode, which makes up the microprogram, and the hardware that it controls defines the functionality of the M8396 module.

The M8396 uses seven programmed array logic devices (PALs) to control various hardware functions. Five of these PALs are used as finite state machines (FSMs). A FSM is a circuit that contains both combinational circuitry and memory elements and which sequences through various states. The FSM's present state is stored in the memory elements. The next state is a function of the present state and the FSM's inputs.

The hardware functions of the PALs are listed in Table 2-1.

Table 2-1 PALs' Hardware Functions

PAL	Controlled Hardware Function
Trap FSM (E110)	traps
Interrupt Control FSM (E103)	BR cycles
Master Control FSM (E78)	UNIBUS NPR cycles
Slave I/O FSM (E102)	UNIBUS slave handshaking protocol
Slow Read/Write Control FSM (E73)	slow read and write cycles
Shift Control (E133)	shifts and rotates of the arithmetic logic unit (ALU)
Microsequence Control (E132)	next address for the microsequencer

2.2 DMF32 MAJOR COMPONENTS

The major components of the DMF32 are shown in Figure 2-1. Figure 2-1 is provided with cross-reference numbers to locate the major components.

1 **Microcontrol Store PROM**
The microcontrol store consists of nine 4K × 4 PROMs that contain a 4K × 36 microprogram. This microprogram provides the intelligence for the DMF32. The microprogram instructs the hardware to perform various functions.

2 **Microword Registers and Parity Checkers**
The microword registers store the microword after the microword is read out of the microcontrol store. When the microword is read, it is checked for parity. A parity error disables the master clock and extinguishes the green LED on the M8396 module.

3 **2901 Bit-Slice Microprocessors (ALU)**
Two 2901's are cascaded together to form an eight-bit ALU with 16 dual-ported eight-bit working registers. This ALU performs arithmetic (e.g., ADD, SUB) and logical (e.g., AND, OR, XOR) operations on data. Bits in the microinstruction control the operation of the ALU and its associated circuitry. The condition code bits emanating from the ALU are clocked into the condition code registers (E131) at the end of each microinstruction if a specific bit in the microinstruction is set.

4 **Micro Shift Control**
The micro shift control determines the shift operation for the 2901 bit-slice microprocessors. It determines the direction of the shift or rotation and also specifies the end round conditions for a single- or double-precision shift or rotation.

5 **2911 Bit-Slice Microsequencers**
Three 2911 bit-slice microsequencers are cascaded together to provide 12-bit addressing capability and a stack depth of four. This microsequencer forms the microprogram's next microword address. The next microaddress can be formed by using one of the following methods:

A. Sequential
The next microaddress is the current microaddress + 1.

B. Unconditional Jump
The next microaddress is specified by bits in the current microinstruction.

C. Stack
Next microaddress is popped from the stack. This method is used for returns from subroutines.

D. Unconditional Jump and Push Stack
The stack is pushed with the next sequential microaddress (i.e., current microaddress + 1). This method is used for subroutine entry.

E. Conditional Jump
If the specified condition is true, the next microaddress is specified by bits in the current microinstruction. If the specified condition is false, the next microaddress is the current microaddress + 1. The particular condition to be tested is specified by bits in the microinstruction.

F. Traps
A trap is a form of microprogram interrupt. A trap request is generated to enable the microcode to service a UNIBUS request. The trap address is a function of the DMF32 register being accessed and whether the access is either a DATO or a DATI cycle.

6 **Microsequence Control**
The microsequence control PAL (E132) determines the data source for the next microinstruction address. The microsequence control PAL controls the 2911 bit-slice microsequencers (E106, 118, 119), while the microsequence control PAL is controlled by bits in the microword.

7 **ALU Condition Code Register**
The ALU condition code register PAL (E131) is clocked with the ALU condition codes at the end of each cycle, if a specific bit in the microinstruction is set. The ALU condition codes can indicate the following: the result of an ALU operation is zero; the sign of the most significant bit is negative; and there is a carry-out bit from the ALU.

8 **Trap control**
A trap is a form of microprogram interrupt. When a CPU accesses the DMF32 via the UNIBUS (DATO and DATI), a certain microsubroutine obtains the requested data (DATI) and transfers the data to the UNIBUS, or accepts the data from the UNIBUS (DATO). The microsubroutine must respond very quickly to the UNIBUS request. A trap is the automatic vectoring of the microprogram to a specific subroutine to satisfy the UNIBUS request.

There is only one trap level. The occurrence of a trap is totally transparent to the code that is being trapped. When the trap code is executing, another trap cannot occur. A non-trap code can disable traps so certain functions that must not be trapped can be performed.

The trap control PAL (E110) contains programmed logic that controls the trapping. This PAL interfaces with the microsequencers (E106, 118, 119), two address multiplexers (E115, 116), part of the UNIBUS slave circuitry (E102), and bits in the microword to perform the function.

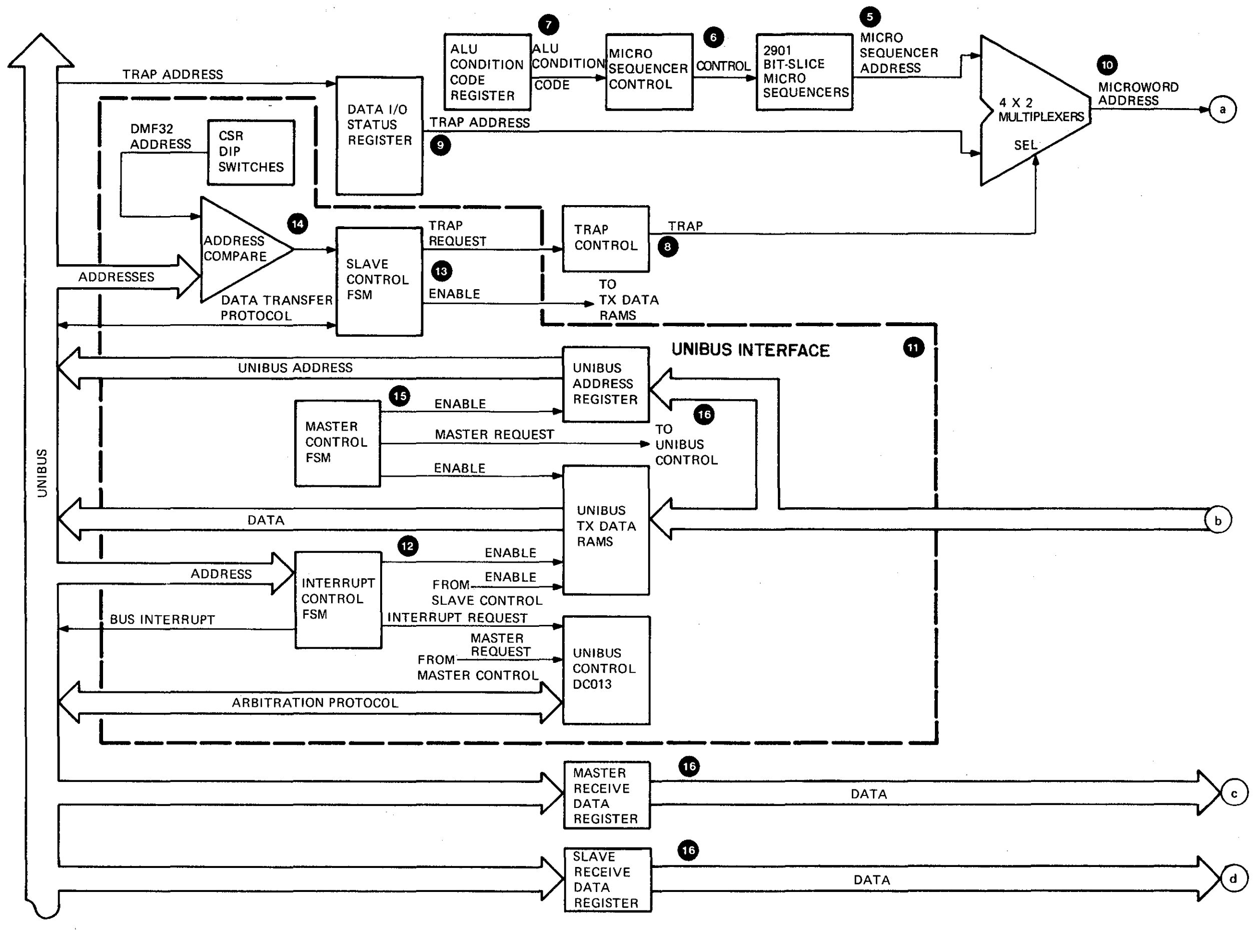


Figure 2-1 DMF32 Hardware Overview (Sheet 1 of 3)

9 **Data I/O Status Register**
The data I/O status register contains the lower five bits of the microcontrol store address for the trap routine. The lower five bits of the address consist of the UNIBUS address bits ⟨4:1⟩ and the UNIBUS control bit C1.

10 **4 × 2 Multiplexers**
The 4 × 2 multiplexers determine if the trap address from the trap generator or the address from the 2911 bit-slice microsequencers is used for the next address for the microcontrol store. If a trap address is produced, the 4 × 2 multiplexer forces address bits ⟨7:5⟩ to all ones, and a multiplex in the lowest five bits from the data I/O status register. In addition, the four most significant address bits (which are now tri-stated), are forced to ones by pull-up resistors.

11 **UNIBUS Interface**
The UNIBUS interface controls DMA slave, and interrupt transfers by using PALs. The 2901 microprocessor initiates NPR and interrupt transfers, but are completed by PAL hardware. The UNIBUS interface is divided into the following three main sections:

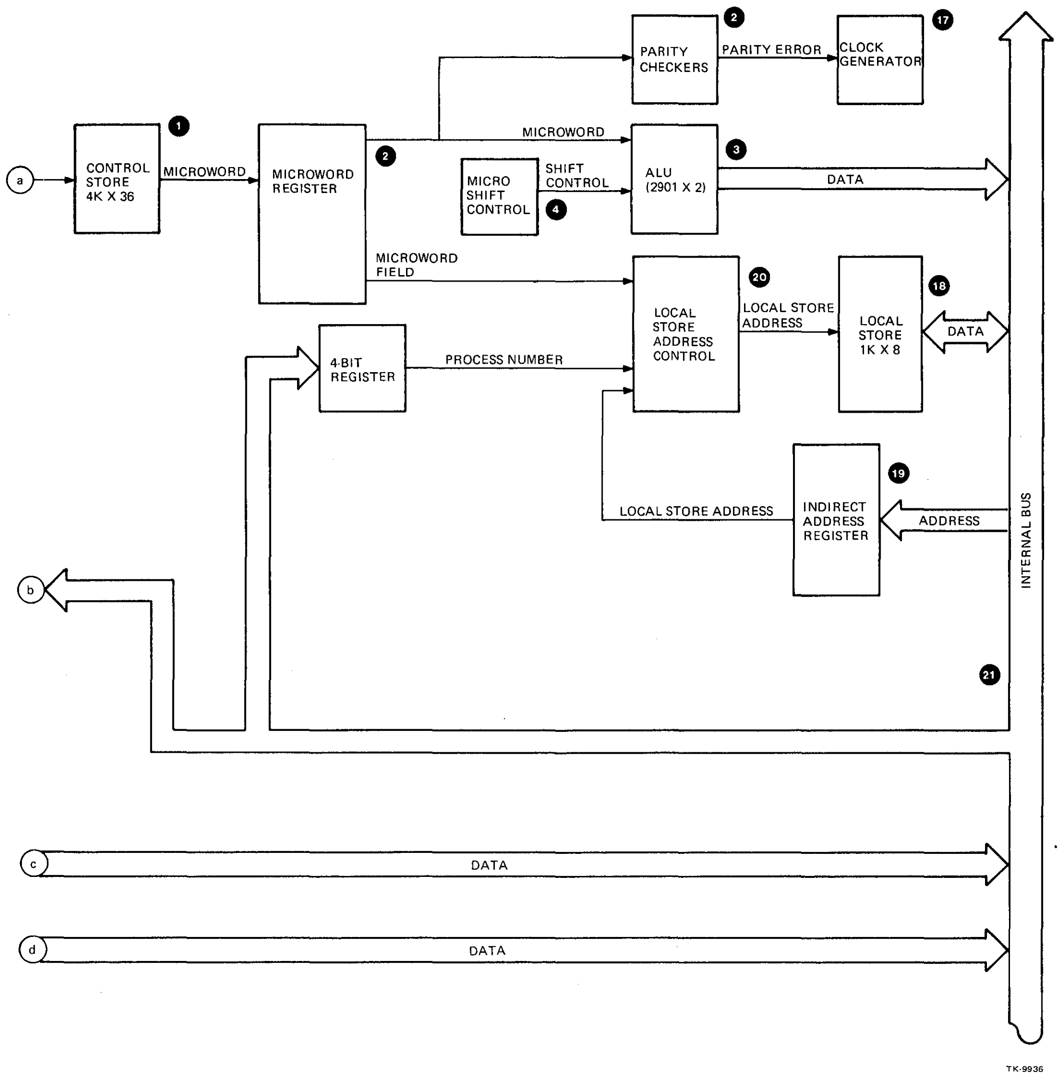


Figure 2-1 DMF32 Hardware Overview (Sheet 2 of 3)

12 **Interrupt Section**
The interrupt circuitry enables the DMF32 to become a bus master and then pass the interrupt vector to the CPU. After the microcode loads the interrupt vector into the interrupt vector register, the microcode sets the BR START bit to initiate the bus cycle. The interrupt PAL E103 controls the bus cycle.

13 **Slave Control Section**
The slave control circuitry allows the DMF32 to respond as a slave device to a UNIBUS request. The slave I/O FSM (E102) controls the slave. The following functions are performed by the slave control circuitry:

- recognizes valid DMF32 register access request (DATO or DATI)
- interfaces with trap circuitry
- controls the handshaking of the UNIBUS signals.

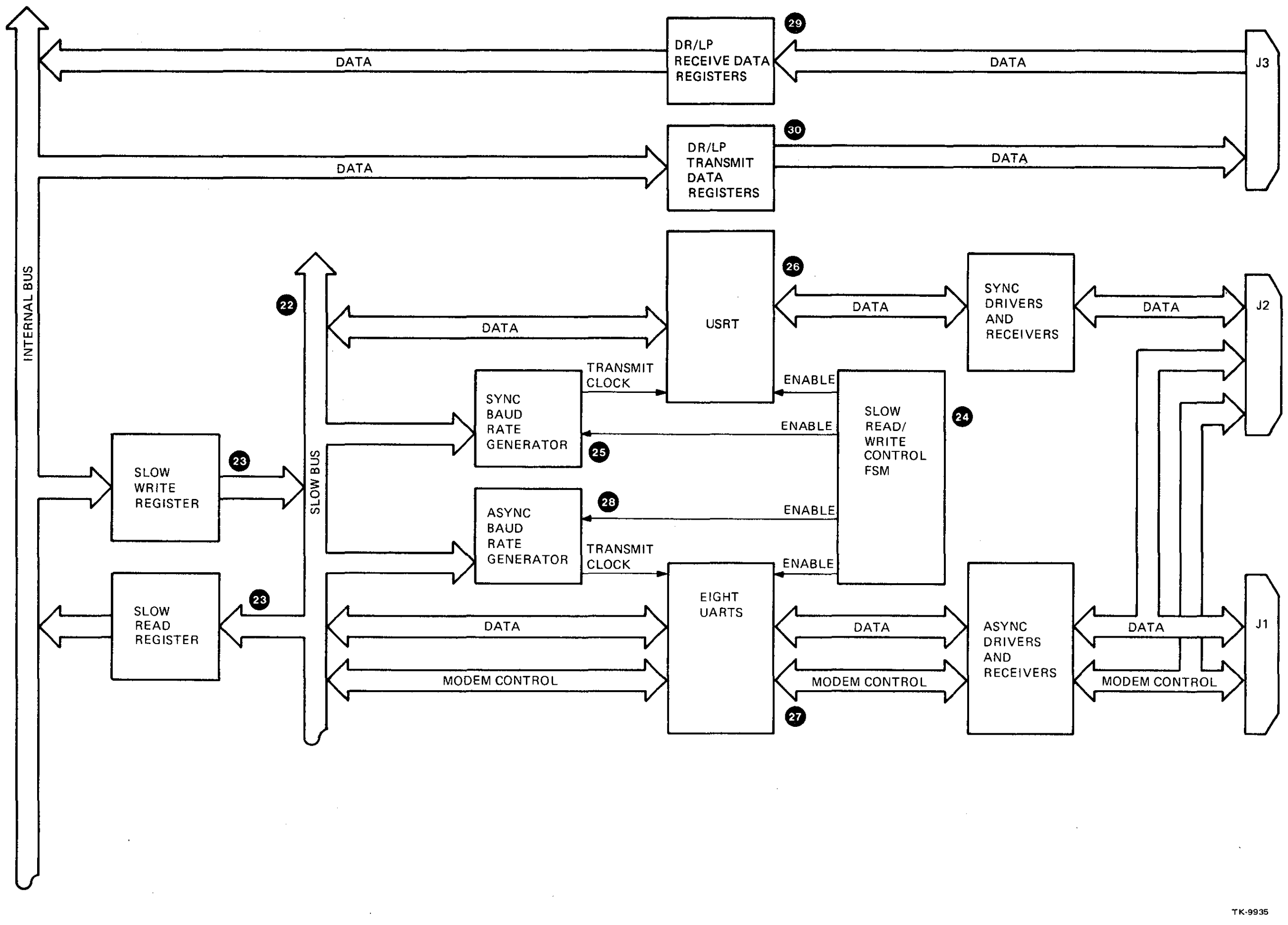


Figure 2-1 DMF32 Hardware Overview (Sheet 3 of 3)

14 **Address Compare**
The address compare determines if the address on the UNIBUS is the valid address for the DMF32. The address on the UNIBUS is compared with the address that is set in the dip switches (E75) on the M8396 module.

15 **Master Control Section**
The master control circuitry enables the DMF32 to become a bus master and perform a DATO or DATI UNIBUS cycle. The microcode loads the UNIBUS address register and data register, and then the microcode sets the NPR START bit to initiate the UNIBUS transaction. The master control FSM (E78) controls the bus cycle.

16 **Receive and Transmit Data/Address Registers**
These registers buffer the data and addresses that are transferred between the UNIBUS and the DMF32's internal bus.

17 **Clock Generator**
This 40.0 MHz master oscillator provides the reference signal for all the clock signals except for the baud rate generators.

18 **Local Store**
The local store is a 1024 byte read/write memory. This memory stores the following:

- images of the 16 DMF32 control status registers
- data buffers (silos)
- microprogram working storage.

The local store is divided into three regions. The first region (512 locations) is indirectly addressable via the indirect address register. The first region contains the silos. The second region (256 locations) is directly addressable within a 16-byte segment. Each segment is indirectly addressable via the process register. This second region contains 16 process context areas of 16 bytes each. The microcode uses eight of the 16 process context areas for storing the state of the eight asynchronous processes. The third region (256 locations) is directly addressable. Bit fields in the microword address the third region.

19 **Indirect Address Register**
The indirect register contains an address that can address one of the first 512 locations of the local store or one of the slow devices (i.e., UART, USART, and BRG).

20 **Local Store Address Control**
The local store address control determines one of three sources for the local store address. The local store address control consists of two dual 4 × 1 multiplexers (E96, 109) and one quad 2 × 1 multiplexer (E107).

21 **Internal Bus**
The internal bus is the medium that is used to transfer data and addresses between the DMF32's internal registers, local store, and the 2901 bit-slice microprocessor.

22 **Slow Bus**
The slow bus interfaces with slow devices that require more than one microinstruction time to perform a read or write cycle. The slow devices are USRT, UARTs, and baud rate generators.

23 **Slow Read/Write Registers**
These registers interface the internal bus and slow bus for slow read and slow write cycles.

24 **Slow Read/Write Control**
The slow read/write control circuitry controls the reading and writing of the MOS devices: UARTs, USRT, and baud rate generators. Since these MOS devices require more than one microinstruction time to be accessed, special circuitry is used to access these devices concurrently with microcode execution. This concurrent operation eliminates the need to stall the clock.

For a slow write cycle, the microcode loads the indirect address register with the address of the device to be written to, and then the microcode loads the slow write register with the desired data which initiates the slow write cycle. While the slow write cycle is in progress, controlled by slow read/write control FSM PAL E73, the microcode can be performing other functions. Similarly, to read data from a MOS device, the microcode performs a special microinstruction to initiate a slow read. Three microinstructions later, the data is in the slow read register, ready for the microcode to read it.

25 **Synchronous Baud Rate Generator**
The synchronous baud rate generator (BRG) provides the synchronous transmitter clock for the USRT.

26 **USRT**
The 2652 multi-protocol communications controller (MPCC) is used for the USRT. The MPCC is a 40-pin monolithic n-channel MOS LSI circuit that can format, transmit, and receive synchronous serial data, while supporting bit-oriented and byte control protocols.

27 **Eight UARTs**
Each one of the eight UARTs is a 2661-3 programmable communications interface (PCI). The UART (PCI) serializes parallel data characters received from the ALU for transmission. Simultaneously, the UARTs can receive serial data and convert the data into parallel character format for input to the ALU.

28 **Asynchronous Baud Rate Generator**
The asynchronous baud rate generator provides the transmitter clocks for both UART 0 and UART 1.

29 **Parallel Port/LP Receive Data Registers**
These registers receive the TTL data from a user device (if present) or receive status from a line printer. Both the user device and the line printer use these receivers.

30 **Parallel Port/LP Transmit Data Registers**
These registers transmit TTL data to either a user device or a line printer. Both the user device and the line printer use these transmitters.

2.3 MICROWORD
The DMF32's intelligence is contained within its 4K × 36 bit microprogram. The microprogram instructs the hardware to perform various functions. Each microword in the microprogram is 36 bits long. Bit 36 is a parity bit. The various microword bit fields are shown in Figure 2-2. Table 2-2 defines the functions of these bit fields.

MIC INST	35	34 33 32	31 30 29	28 27 26	25 24	23 22 21 20 19 18 17 16	15 14 13 12	11 10 9 8	7 6 5	4	3	2	1	0
L.S. RD	EVEN PARITY BIT	ALU INSTRUCTION CODE <8:7:6:5:4:3:2:1:0> ALU DESTINATION CONTROL	ALU FUNCTION CONTROL	ALU SOURCE CONTROL	LOCAL STORE ADDRESSING <SEG> <9:8>	<SECTOR> <BYTE> <7:6:5:4> <3:2:1:0>	2901 REGISTER FILE B ADDRESS <0:1:2:3>	2901 REGISTER FILE A ADDRESS <0:1:2:3>	0:0:0	ALU CARRY IN	COND CODE REG UPDATE ENABLE	NEXT ADDR 0 = PC 1 = RSB	SHIFT/ROTATE SELECT	SHIFT PRECISION SELECT
L.S. RD					<00>	INDIRECT LOW SEGMENT <NOT USED – SEE (3)>			0:0:1					
L.S. RD TX DATA LO (4&5)					<01>	INDIRECT HIGH SEGMENT <NOT USED – SEE (3)>			0:1:0					
L.S. RD TX. DATA HI (4)					<10>	PROCESS SEGMENT <NOT USED> <BYTE> = LATCH (PROC<3:0>)			0:1:1					
L.S. WRT					<11> <9:8>	DIRECT SEGMENT <BYTE ADDRESS> <7:6:5:4:3:2:1:0>			1:0:0					
DER RD/WRT					TRP CTRL	DER FAST REGISTER (1) CMD <1:0> ADDRESS <2:1:0> SELECT <2:1:0>			1:0:1					
COND. JMP. (2)					12 BIT JUMP ADDRESS <9:8:7:6:5:4:3:2:1:0>	BIT 4 = <10>; BIT 5 = <11>			1:1 0	X JMP ADDR <11:10>		0=N 1=N 2=C 3=AXZ 4=CO 5=JSB 6=POPSTK 7=JMP		
COND. JMP. (2)									1:1 1	X				
MIC INST	35	34 33 32	31 30 29	28 27 26	25 24	23 22 21 20 19 18 17 16	15 14 13 12	11 10 9 8	7 6 5	4	3	2	1	0

THESE BITS STORE THE LITERAL (CONSTANTS) FIELD <7 6 5 4 3 2 1 0>

NOTE:

(1) SLOW (CYCLE) DER ACCESSES USE THE INDIRECT ADDRESS REGISTER RATHER THAN THE ADDRESS AND SELECT FIELDS HERE. NOTE THAT SHIFT/ROTATE OR INSTRUCTIONS USING THE 'A' FILE ADDRESS MAY NOT BE USED WHEN READING LITERALS AS THE 'SELECT',' 'A' ADDRESS' AND THE 'PRECISION SELECT' FIELDS ARE USED FOR THE 'LITERAL' FIELD.
(2) THESE INSTRUCTIONS DIFFER ONLY IN THE HALF OF MEMORY IN WHICH THE JUMP DESTINATION IS LOCATED AS THE LOW ORDER BIT OF THE MICROINSTRUCTION DETERMINES ADDR <11>.
(3) LOCAL STORE <7:0> ADDRESS SOURCE DEPENDENT UPON SEGMENT SELECTED BY HIGH ORDER BITS <9:8>.
(4) MWR <17:16> SELECT TX DATA RAM ADDRESS. 00 = MASTER, 01 = VECTOR, 10 = UNUSED, 11 = SLAVE.
(5) USING THIS INSTRUCTION WITH MWR <17:16> = 11 SETS THE TRAP ACK FLAG TO THE SLAVE PAL.

TK-9942

Figure 2-2 Microword Bit Fields

Table 2-2 Microword Bit Fields

Microword Bit Field	Microinstruction	Function
⟨7:5⟩	N/A	Defines type of microinstruction 000 or 001 = local store read (LS.RD) 010 = local store read transmit data low (LS.RD.TX.DAT.LO) 011 = local store read transmit data high (LS.RD.TX.DAT.HI) 100 = local store write (LS.WRT) 101 = DER read/write (DER) 110 or 111 = conditional jump (JMP)
⟨2:0⟩	conditional jump	Selects jump condition 000 = register ALU N 001 = register ALU Z 010 = register ALU C 011 = auxiliary ALU Z 100 = register C0 101 = unconditional jump to subroutine 110 = unconditional jump and pop stack 111 = unconditional jump
⟨2⟩	any except condition jump	Selects address source 0 = microprogram 1 = return to subroutine (use stack then pop)
⟨33⟩⟨1:0⟩	any	Selects end around conditions for shift, rotate or working register, Q data
⟨3⟩	any	Selects clocking of ALU condition code register 0 = do not clock 1 = do clock
⟨4⟩	any	Carry-in bit to ALU
⟨11:8⟩	any	A address for working register in 2901
⟨15:12⟩	any	B address for working register in 2901
⟨25:16⟩	local store read local store read transmit data high local store read transmit data low local store write	10-bit local store address

Table 2-2 Microword Bit Fields (Cont)

Microword Bit Field	Microinstruction	Function
⟨25:24⟩	DER	Selects trap control 00 = no change 01 = disable traps 1X = enable traps
⟨23⟩	DER	Selects DER read or write 0 = DER write 1 = DER read
⟨22:16⟩	DER	7-bit DER address
⟨5⟩⟨25:16⟩⟨14⟩	conditional jump	12-bit jump address
⟨18:16⟩⟨11:8⟩ ⟨0⟩	DER read	8-bit literal
⟨28:26⟩	any	ALU (2901) source control
⟨31:29⟩	any	ALU (2901) function control
⟨34:32⟩	any	ALU (2901) destination control
⟨35⟩	any	even parity bit

CHAPTER 3
DETAILED DESCRIPTION

3.1 INTRODUCTION

This chapter describes the hardware operation of the DMF32 in detail. Figure 3-1 (sheets 1 through 3) is a simplified figure of the data flow of the DMF32.

Refer to Figure 3-1 for the following description of the DMF32. The appropriate detailed descriptions are referenced in parentheses.

3.1.1 UNIBUS Interface

The DMF32 can function as either a bus master device or as a bus slave device with the UNIBUS. When the DMF32 is a master device, the master control FSM controls the master DATO or DATI bus cycle (3.2). Microcode enables the master control FSM to initiate the bus cycle. The enabled master control FSM applies MASTER REQUEST to the UNIBUS control. The UNIBUS control handles the arbitration protocol between the DMF32 and the bus slave device.

When the DMF32 becomes a bus master in a DATO cycle, the master control FSM enables the UNIBUS address register to apply the slave device address to the UNIBUS. This address is previously loaded into the UNIBUS address register by microcode. Next, the master control FSM enables the TX data RAMs to apply the data for the slave device to the UNIBUS. In a DATI cycle, the data on the UNIBUS is loaded into the master receive data register.

The interrupt control FSM controls the interrupt operation (3.3). The microcode initiates the operation of the interrupt control FSM. The interrupt control FSM applies INTERRUPT REQUEST to UNIBUS control. This enables the UNIBUS control. The UNIBUS control handles the arbitration protocol between the DMF32 and the slave device. The interrupt control FSM enables the TX data RAMs. The TX data RAMs applies the vector, that was loaded previously by microcode, to the UNIBUS. Next, the interrupt control applies the BUS INTERRUPT to the UNIBUS.

The UNIBUS slave I/O operation (3.4) is controlled by the slave control FSM. The slave control FSM controls the UNIBUS handshaking protocol, while trap routines control the data transfers. The UNIBUS address from the bus master device and the CSR dip switch settings are compared and when there is a match, the slave control FSM produces a TRAP REQUEST and monitors the UNIBUS handshaking protocol. In a slave DATO cycle, the TRAP REQUEST clocks the data on the UNIBUS into the slave receive data register.

The TRAP REQUEST is monitored by the trap control (3.5). The trap control enables the trap multiplexer, so that the trap address from the DATA I/O status register is applied to the control store instead of the microsequencer address.

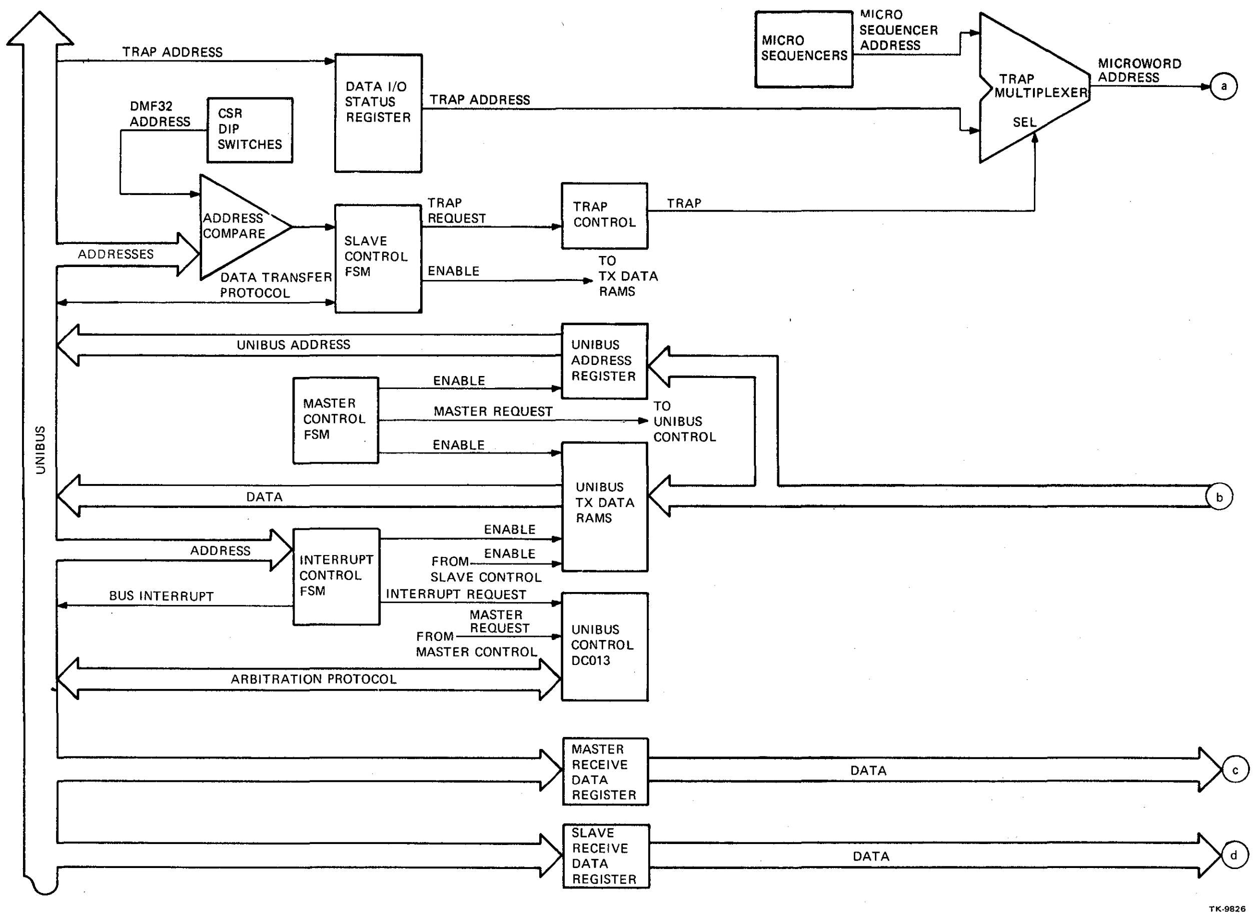


Figure 3-1 DMF32 Overview (Sheet 1 of 3)

3.1.2 DMF32 Intelligence

The microsequencer (3.6) consists of three cascaded 2911 microprogram sequencers. The 12-bit microsequencer address, that is applied to the control store (3.7), reads out a 36-bit microword from the control store. This read-out microword is stored in the microword register. The microword is also checked by the parity checker for even parity. If there is a parity error, the parity checker disables the master clock (3.8).

From the microword in the microword register, the ALU instruction code and the register file A and B address fields are applied to the ALU (3.9). The ALU performs either an arithmetic or logic function as defined by the ALU instruction code. The results of the ALU operation may be applied to the internal bus.

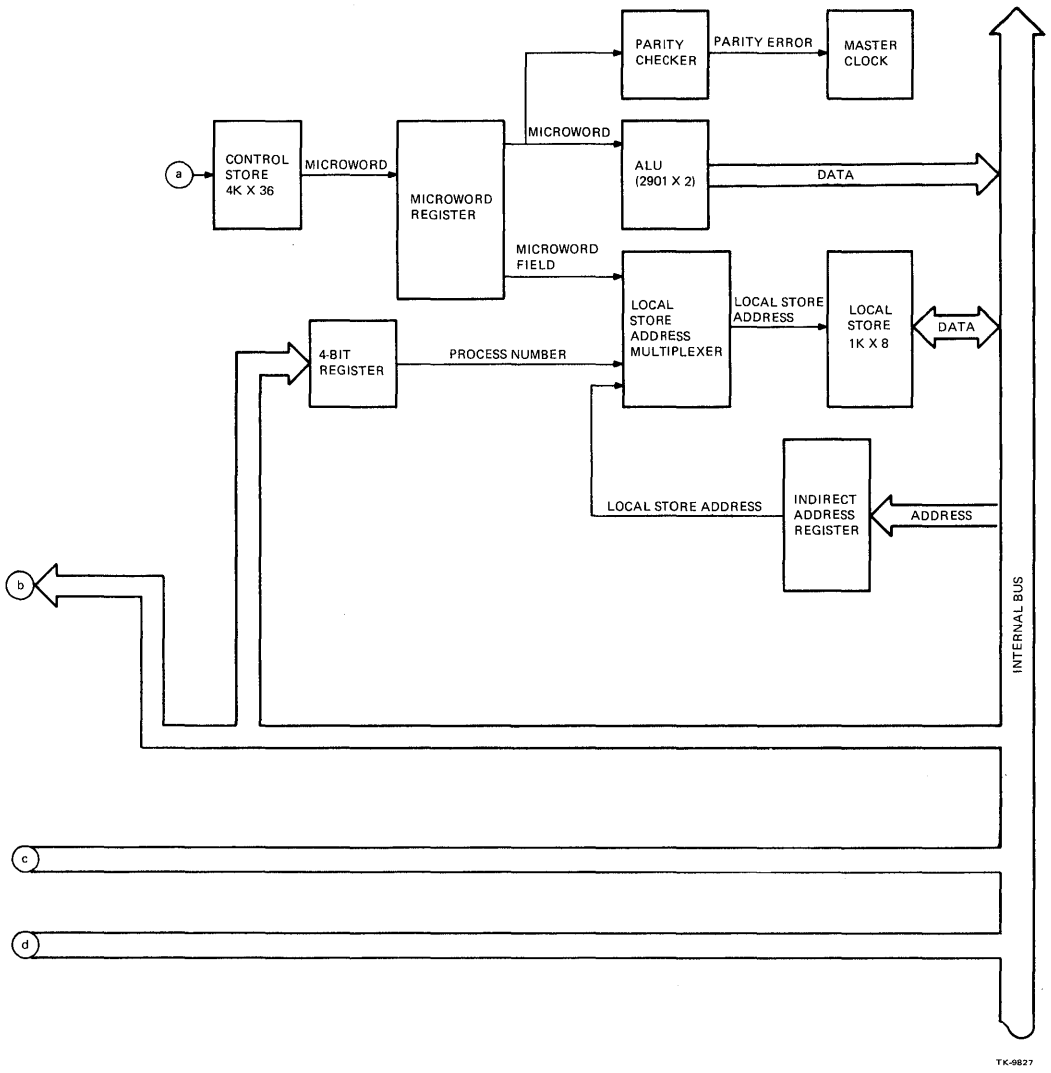


Figure 3-1 DMF32 Overview (Sheet 2 of 3)

3.1.3 Local Store

The local store (3.10) contains the DMF32 CSRs, device registers, process context, DMF32 vectors, temporary registers, and the DMF32 silos. The local store address can originate from one of three different sources:

- A field from the microword, stored in the microword register
- The process number from the four-bit latch, loaded in the four-bit latch by microcode
- The local store address register.

The DMF32 performs a discrete external register (DER) write to load either the four-bit latch or the nine-bit indirect address register. The LS.WRT instruction is used to load data from the internal bus into the local store. The LS.RD instruction is used to apply data to the internal bus from the local store.

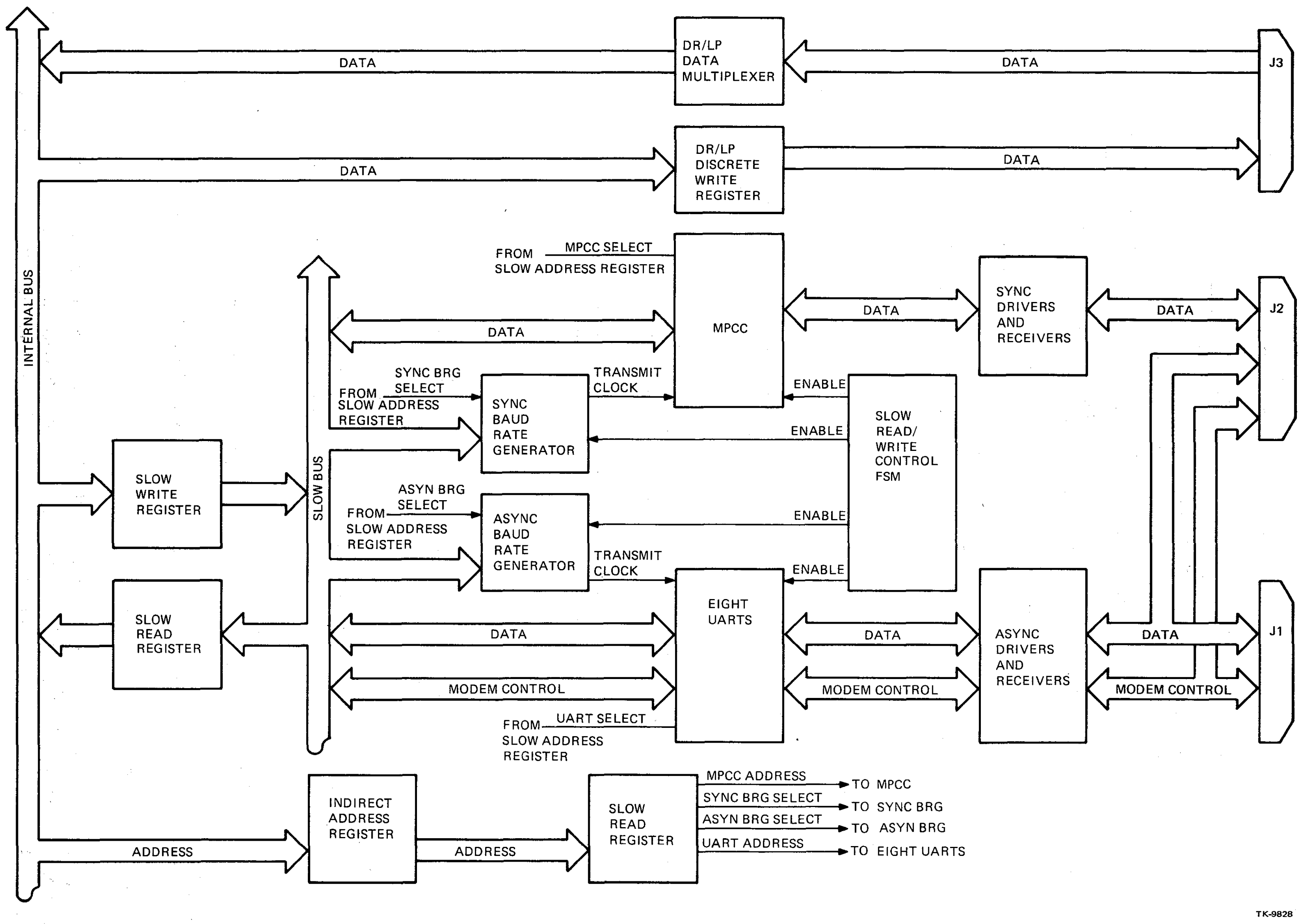


Figure 3-1 DMF32 Overview (Sheet 3 of 3)

3.1.4 Internal Bus
The DMF32 performs a DER write instruction (3.12) to load the data from the internal bus to one of the write discrete external registers. The UNIBUS address register, DR/LP register, three eight-bit addressable latches, and the indirect address register are write discrete external registers.

The DMF32 performs a DER read instruction (3.11) to apply data from a read discrete external register to the internal bus. The master receive register, slave receive data register, slow read register, DR/LP discrete read register, the data multiplexer register, and the literal are read discrete external registers.

The DMF32 performs a LS.RD.TX.DAT.LO instruction to load the transmit data RAMs (low byte) with the data on the internal bus. A LS.RD.TX.DAT.HI instruction loads the transmit data RAMs (high byte) with the data on the internal bus. The data in the transmit data RAMs is applied to the UNIBUS.

3.1.5 Slow Bus
The slow bus interfaces with the devices that require more than one microinstruction time to perform a read or write cycle (3.12). The MPCC, the two baud rate generators, and the eight UARTs are slower devices that interface with the slow bus. The slow bus is interfaced with the internal bus via a slow read register for the slow read cycles and a slow write register for the slow write cycle.

The slow address register is loaded with the address for the slow device via the indirect address register. The slow address register selects either the MPCC, one of the two baud rate generators, or one of the eight UARTs.

The slow read/write control FSM (3.13) controls the slow read (3.14) and slow write cycles (3.14).

The MPCC formats, transmits, and receives synchronous serial data. Also the MPCC supports bit-oriented or byte control protocols. The output of the MPCC is applied to the synchronous device via EIA/CCITT drivers (3.15).

The synchronous baud rate generator provides the transmit clock for the MPCC. The transmit clock rate is programmable.

Each UART serializes the parallel data characters received from the processor for transmission. Simultaneously, the UARTs can receive serial data and convert the data into parallel character format for input to the processor. The character length, parity, stop bit length, and baud rate selection are all programmable.

UART 0 and UART 1 have modem control and also can operate in split speed mode. The transmit and receive clocks of UART 2 through UART 7, and the receive clocks of UART 0 and UART 1 all use internal clocks. The asynchronous baud rate generator provides the transmit clock for both UART 0 and UART 1. The clock rate of the asynchronous baud rate generator is programmable.

The data from the processor for the printer or the parallel interface is applied to the device via the DR/LP discrete write register and the DR/LP drivers (3.16). The data from these devices is applied to the processor via the DR/LP receivers (3.16) and the DR/LP data multiplexer.

3.2 UNIBUS MASTER CONTROL LOGIC

The UNIBUS master control logic enables the DMF32 to become a UNIBUS master to perform the following:

- DATI (data-in)
- DATIP-DATO(B) (data-in-pause; data-out)
- DATO (data-out)
- DATO(B) (data-out byte).

Only one of the above four cycles can be performed per bus mastership.

The UNIBUS master control logic consists of the following:

- UNIBUS control FSM (E90) — requests and obtains UNIBUS mastership
- Master control FSM (E78) — controls the NPR cycle
- Master RX data registers (E68,39) — holds data from a DATI cycle
- TX data RAMs (E37,29,50,30) — holds data for a DATO(B) cycle
- UNIBUS address registers (E67,93) — holds low 16-bits of the UNIBUS address
- Eight-bit latches (E95,82,94) — holds the two MSB bits of the UNIBUS address and UNIBUS control bits.

3.2.1 Master Device Pre-Cycle Initiation

Prior to initiating a cycle, the microcode loads the data, address, and control registers. These registers are loaded differently for each cycle. Tables 3-1 through 3-4 list the registers and their preloaded contents.

Table 3-1 DATO Cycle Register Contents

Register	Contents
COMF C1 H	1
COMF TX C0 H	1
TX.ADR.LO⟨7:0⟩	low byte of UNIBUS address
TX.ADR.HI⟨15:8⟩H	high byte of UNIBUS address
TX.ADR⟨17:16⟩H	two MSBs of UNIBUS address
TX.DATA[00]⟨7:0⟩	low byte of data
TX.DATA[00]⟨15:8⟩	high byte of data

Table 3-2 DATOB Cycle Register Contents

Register	Contents
COMF C1 H	1
COMF TX C0 H	0
TX.ADR.LO⟨7:0⟩	low byte of UNIBUS address
TX.ADR.HI⟨7:0⟩	high byte of UNIBUS address
TX.ADR⟨17:16⟩	two MSB's of UNIBUS address
TX.DATA[00]⟨7:0⟩ or TX.DATA[00]⟨15:8⟩	byte of data

TX.ADR⟨0⟩ specifies to the slave whether the high or low byte is to be transferred.

Table 3-3 DATI Cycle Register Contents

Register	Contents
COMF C1 H	0
COMF TX C0 H	0
TX.ADR.LO⟨7:0⟩	low byte to be sent to slave
TX.ADR.HI⟨15:8⟩	high byte to be sent to slave

Table 3-4 DATIP-DATOB Cycle Register Contents

Register	Contents
COMF C1 H	0
COMF TX C0 H	1
TX.ADR.LO⟨7:0⟩	low byte of UNIBUS address
TX.ADR.HI⟨15:8⟩	high byte of UNIBUS address
TX.ADR.⟨17:16⟩	two MSBs of UNIBUS address
TX.DATA[00]⟨7:0⟩*	low data byte sent to slave device
TX.DATA[00]⟨15:8⟩*	high data byte sent to slave device

After the data, address, and control registers are loaded, the microcode initiates an NPR cycle by asserting the DER bit COMF NPR START H from the deasserted state. If COMF NPR START H is asserted from the previous NPR cycle, the microcode deasserts COMF NPR START H for at least two microcycles before reasserting COMF NPR START H.

The asserted COMF NPR START H is applied to the master control FSM (E78). The master control FSM (E78) initiates, controls, and terminates the UNIBUS master cycle.

Refer to Figure 3-2 and Figure 3-3 for the following description of the UNIBUS master control logic.

In state 0, COMF NPR START H is deasserted. Asserting COMF NPR START H causes the master control FSM (E78) to assert COMC MST REQ H. COMC MST REQ H is applied to the UNIBUS control (E90). The UNIBUS control (E90) asserts BUS NPR L to request bus mastership. Now in state 1, the DMF32 waits to become bus master.

* During the DATOB portions of the read-modify-write cycle the data byte that is sent to the slave device is either TX.DATA[00]⟨7:0⟩ or TX.DATA[00]⟨15:8⟩.

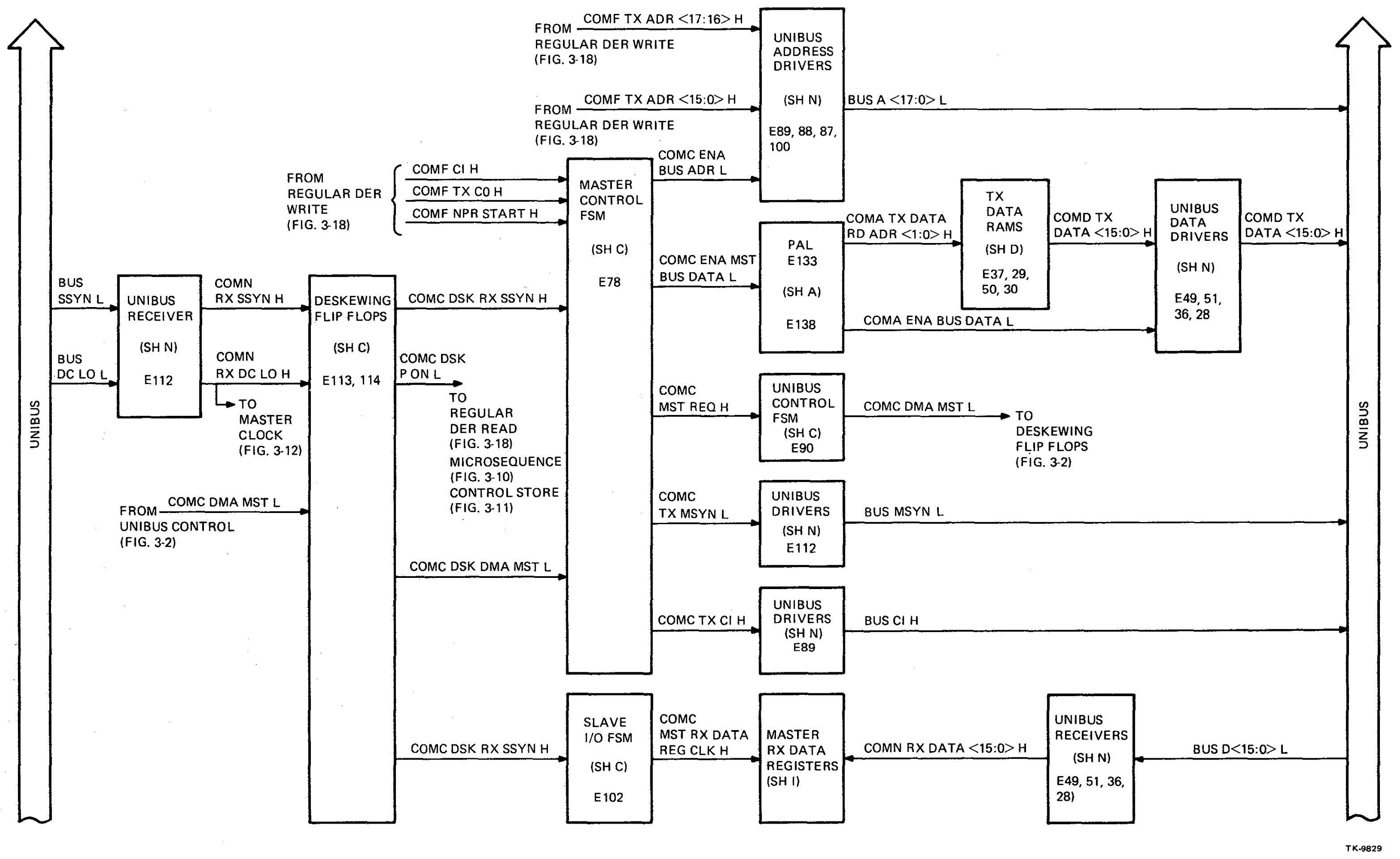

UNIBUS
FROM REGULAR DER WRITE (FIG. 3-18)
COMF TX ADR <17:16> H
COMF TX ADR <15:0> H
UNIBUS ADDRESS DRIVERS (SH N) E89, 88, 87, 100
BUS A <17:0> L
FROM REGULAR DER WRITE (FIG. 3-18)
COMF CI H
COMF TX C0 H
COMF NPR START H
MASTER CONTROL FSM (SH C) E78
COMC ENA BUS ADR L
COMC ENA MST BUS DATA L
PAL E133 (SH A) E138
COMA TX DATA RD ADR <1:0> H
TX DATA RAMS (SH D) E37, 29, 50, 30
COMD TX DATA <15:0> H
UNIBUS DATA DRIVERS (SH N) E49, 51, 36, 28
COMD TX DATA <15:0> H
COMA ENA BUS DATA L
BUS SSYN L
BUS DC LO L
UNIBUS RECEIVER (SH N) E112
COMN RX SSYN H
COMN RX DC LO H
TO MASTER CLOCK (FIG. 3-12)
DESKEWING FLIP FLOPS (SH C) E113, 114
COMC DSK RX SSYN H
COMC DSK P ON L
TO REGULAR DER READ (FIG. 3-18) MICROSEQUENCE (FIG. 3-10) CONTROL STORE (FIG. 3-11)
COMC MST REQ H
UNIBUS CONTROL FSM (SH C) E90
COMC DMA MST L
TO DESKEWING FLIP FLOPS (FIG. 3-2)
FROM UNIBUS CONTROL (FIG. 3-2)
COMC DMA MST L
COMC TX MSYN L
UNIBUS DRIVERS (SH N) E112
BUS MSYN L
COMC DSK DMA MST L
COMC TX CI H
UNIBUS DRIVERS (SH N) E89
BUS CI H
COMC DSK RX SSYN H
SLAVE I/O FSM (SH C) E102
COMC MST RX DATA REG CLK H
MASTER RX DATA REGISTERS (SH I)
COMN RX DATA <15:0> H
UNIBUS RECEIVERS (SH N) E49, 51, 36, 28)
BUS D<15:0> L
UNIBUS
TK-9829

Figure 3-2 UNIBUS Master Control Logic

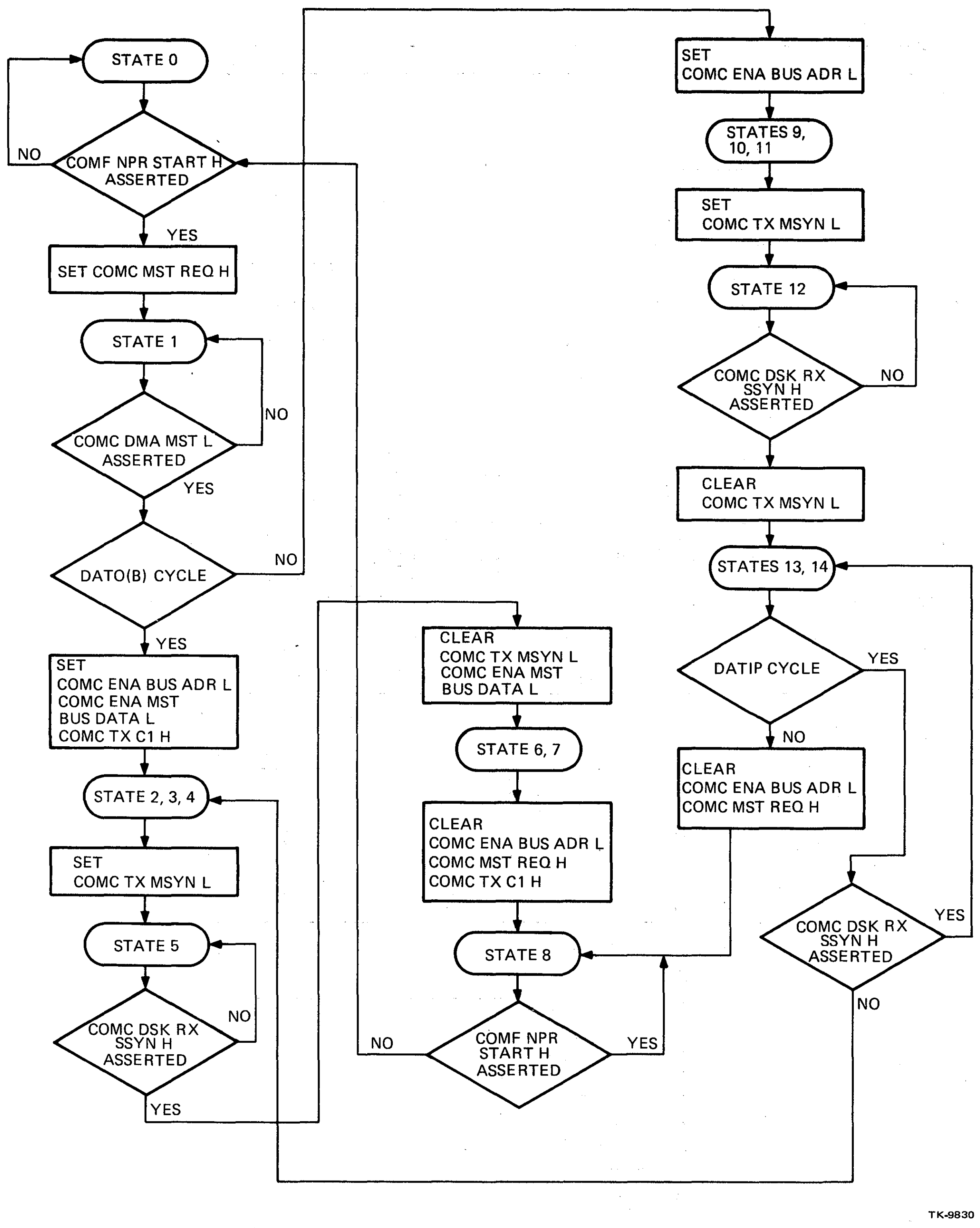

STATE 0
COMF NPR START H ASSERTED
NO
YES
SET COMC MST REQ H
STATE 1
COMC DMA MST L ASSERTED
NO
YES
DATO(B) CYCLE
NO
YES
SET COMC ENA BUS ADR L COMC ENA MST BUS DATA L COMC TX C1 H
STATE 2, 3, 4
SET COMC TX MSYN L
STATE 5
COMC DSK RX SSYN H ASSERTED
NO
YES
CLEAR COMC TX MSYN L COMC ENA MST BUS DATA L
STATE 6, 7
CLEAR COMC ENA BUS ADR L COMC MST REQ H COMC TX C1 H
STATE 8
COMF NPR START H ASSERTED
NO
YES
SET COMC ENA BUS ADR L
STATES 9, 10, 11
SET COMC TX MSYN L
STATE 12
COMC DSK RX SSYN H ASSERTED
NO
CLEAR COMC TX MSYN L
STATES 13, 14
DATIP CYCLE
YES
NO
CLEAR COMC ENA BUS ADR L COMC MST REQ H
COMC DSK RX SSYN H ASSERTED
YES
NO
TK-9830

Figure 3-3 Master Control FSM Flow

3.2.2 Master Device DATO(B) Cycle

When the DMF32 becomes bus master, UNIBUS control (E90) asserts COMC DMA MST L. COMF C1 H is applied to the master control FSM (E78). If COMF C1 H is asserted and COMF TX C0 = 0, a DATO(B) is to be performed. The master control FSM (E78) asserts COMC ENA BUS ADR L, COMC ENA MST BUS DATA L and COMC TX C1 H. The master control FSM (E78) now enters state 2.

COMC ENA BUS ADR L is applied to the UNIBUS address drivers (E99,88,87,100,89). COMC ENA BUS ADR L enables the UNIBUS address drivers. The UNIBUS drivers contain the master device address, which was previously loaded by microcode.

COMC ENA MST BUS DATA L is applied to the shift control (E133), which causes the shift control (E133) to assert COMA TX DATA RD ADR ⟨1:0⟩ H and COMA ENA BUS DATA L. COMA TX DATA RD ADR ⟨1:0⟩ H applies address 00 to the TX data RAMs (E37,29,50,30). COMA ENA BUS DATA L is applied to the UNIBUS data drivers (E49,51,36,28) to enable these UNIBUS data drivers.

The master control FSM enters the wait states of 3 and 4. Upon entering state 5, the master control FSM (E78) asserts COMC TX MSYN L. Refer to Table 3-5 for the truth table of the master control FSM (E78). COMC TX MSYN L is applied to the slave device via the UNIBUS drivers. BUS MSYN L is applied to the slave device to request the slave device to accept the data (BUS D⟨15:0⟩ L) from the UNIBUS data drivers.

In state 5, the master control FSM (E78) waits for the slave device to assert BUS SSYN L. BUS SSYN L informs the master control FSM (E78) that the slave device has completed the data transfer. BUS SSYN L is applied to the master control FSM (E78) via the UNIBUS driver (E112) and deskewing flip-flops (E113,114). Applying the deasserted COMC DSK RX SSYN L to the master control FSM (E78) causes the master control FSM (E78) to deassert COMC TX MSYN L and COMC ENA MST BUS DATA L. Deasserting COMC ENA MST BUS DATA L removes the data from the UNIBUS.

The master control FSM (E78) enters the wait states 6 and 7. Upon entering wait state 8, COMC ENA BUS ADR L, COMC TX C1 H, and COMC MST REQ H are all deasserted. The deasserted COMC MST REQ H is inverted and applied to the master control FSM (E78), that causes the DMF32 to relinquish bus mastership by deasserting BUS BBSY L.

The master control FSM (E78) loops in state 8 waiting for the microcode to deassert COMF NPR START H. When COMF NPR START H is deasserted, state 0 is entered.

Table 3-5 Master Control FSM (E78) Truth Table

DMA MST	C1	TX C0	NPR STRT	RX SSYN	ST	TX C1	EN AD	EN DAT	MST REQ	TX MSYN	S2	S1	S0	ST	TX C1	EN AD	EN DAT	MST REQ	TX MSYN	S2	S1
X	X	X	0	X	X	X	X	X	X	X	X	X	X	0	X	1	1	0	1	0	1
0	X	X	1	X	0	X	1	1	0	1	0	1	0	0	X	1	1	0	1	0	1
		X	X	1	X	0	X	1	1	0	1	0	1	0	1	X	1	1	1	1	1
1																					
1	X	X	1	X	1	X	1	1	1	1	1	1	1	1	X	1	1	1	1	1	1
0	1	X	1	X	1	X	1	1	1	1	1	1	1	2	1	0	0	1	1	1	1
X	X	X	1	X	2	1	0	0	1	1	1	1	1	3	1	0	1	1	1	1	1
X	X	X	1	X	3	1	0	1	1	1	1	1	0	4	1	0	0	1	1	1	0
X	X	X	1	X	4	1	0	0	1	1	1	0	1	5	1	0	0	1	0	1	1
X	X	X	1	0	5	1	0	0	1	0	1	1	1	5	1	0	0	1	0	1	1
X	X	X	1	1	5	1	0	0	1	0	1	1	1	6	1	0	1	1	1	1	1
X	X	X	1	X	6	1	0	1	1	1	1	1	1	7	1	0	1	1	1	1	1
X	X	X	1	X	7	1	0	1	1	1	1	1	0	8	0	1	1	0	1	1	1
X	X	X	1	X	8	0	1	1	0	1	1	1	1	8	0	1	1	0	1	1	1
0	0	X	1	X	1	X	1	1	1	1	1	1	1	9	0	0	1	1	1	1	1
X	X	X	1	X	9	0	0	1	1	1	1	1	1	10	0	0	1	1	1	1	1
X	X	X	1	X	10	0	0	1	1	1	1	1	0	11	0	0	1	1	1	1	0
X	X	X	1	X	11	0	0	1	1	1	1	0	1	12	0	0	1	1	0	1	1
X	X	X	1	0	12	0	0	1	1	0	1	1	1	13	0	0	1	1	1	1	1
X	X	X	1	1	12	0	0	1	1	0	1	1	1	13	0	0	1	1	1	1	0
X	X	X	1	X	13	0	0	1	1	1	1	0	0	14	0	0	1	1	1	0	1
X	X	0	1	X	14	0	0	1	1	1	0	1	1	8	0	1	1	0	1	1	1
X	X	1	1	1	14	0	0	1	1	1	0	1	1	14	0	0	1	1	1	0	1
X	X	1	1	0	14	0	0	1	1	1	0	1	1	2	1	0	0	1	1	1	1

3.2.3 DMF32 Master Device DATI(P) Cycle
For a DATI(P) cycle (COMF C1 H is deasserted), the master control FSM (E78) proceeds from state 1 to state 9, instead of state 1 to state 2 as in a DATO(B) cycle. Upon entering state 9, the master control FSM (E78) asserts COMC ENA BUS ADR L to enable the UNIBUS address drivers. After the wait states 10 and 11, the master control FSM (E78) asserts COMC TX MSYN L and enters state 12.

In state 12, the master control FSM (E78) waits for the slave device to assert BUS SSYN L. The slave device asserts BUS SSYN L to inform the DMF32 that the slave device has applied data to the UNIBUS. BUS SSYN L is applied to the master control FSM (E78) via the UNIBUS receivers (E112) and the deskewing flip-flops (E113,114). BUS SSYN L (COMC DSK RX SSYN L) is also applied to the combinational part of the slave I/O FSM (E102), which causes the slave I/O FSM (E102) to assert COMC MST RX DATA REG CLK H. COMC MST RX DATA REG CLK H clocks the master RX data registers (E68,39), so that the data (BUS D⟨15:0⟩L) from the slave device is loaded into the master RX data registers via UNIBUS receivers (E49,51,36,28).

Entering state 13, the master control FSM (E78) deasserts COMC TX MSYN L. After state 13 and 14, COMF TX C0 H, that is applied to the master control FSM (E78), determines if a DATI cycle (COMF C0 H is deasserted) was just performed or a DATIP cycle (COMF TX C0 H is asserted) was just performed. If a DATI cycle was performed, the master control FSM (E78) deasserts COMC ENA BUS ADR L and COMC MST REQ H, and enters state 8 to terminate the cycle.

If a DATIP cycle was just performed, the master control FSM (E78) now performs a DATO(B) cycle. After COMC DSK RX SSYN H deasserts, the master control FSM (E78) asserts both COMC ENA MST BUS DATA L and COMC TX C1 H, while COMC ENA BUS ADR L is still asserted from the DATIP cycle. The master control FSM (E78) enters state 2 to perform a DATO(B) cycle.

3.3 UNIBUS INTERRUPT LOGIC
The UNIBUS interrupt logic enables the DMF32 to become a UNIBUS master to perform an interrupt operation. The UNIBUS interrupt logic consists of the following:

- Interrupt control FSM (E103) — controls the BR cycle
- TX data RAMs (E37,29,50,30) — holds the interrupt vector
- BR priority switch pack (E77) — selects the BR request level.

Prior to initiating the BR cycle, the microcode loads the interrupt vector into location 01 of the TX data RAMs (E37,29,50,30) from the bus. Also, the microcode verifies that COMC INT REQ L is deasserted.

Microcode initiates a BR cycle by asserting COMF BR START H from a deasserted state. If COMF BR START H is asserted from the previous BR cycle, the microcode clears COMF BR START H for at least two microcycles before asserting COMF BR START H again. The microcode aborts a BR cycle by deasserting COMF BR START H.

Refer to Figure 3-4 and Figure 3-5 for the following description of the UNIBUS interrupt logic.

In state 0, the microcode asserts COMF BR START H from a deasserted state to initiate the BR cycle. Interrupt control FSM (E103) asserts and applies COMC INT REQ L to the UNIBUS control (E91). The UNIBUS control (E91) asserts BUS BRX L to request bus mastership. BUS BRX L is applied to the BR priority switch pack (E77). The BR priority switch pack routes the BUS BRX L, BUS BGX IN H, and BUS BGX OUT H from the UNIBUS control (E91) to selected BR and BG lines.

In state 1, the DMF32 becomes bus master by the UNIBUS control (E91) asserting COMC INT MST L and BUS BBSY L. As soon as the DMF32 becomes bus master, the interrupt control FSM (E103) asserts COMC ENA VEC L. Refer to Table 3-6 for the interrupt control FSM (E103) truth table. COMC ENA VEC L is applied to PAL (E133).

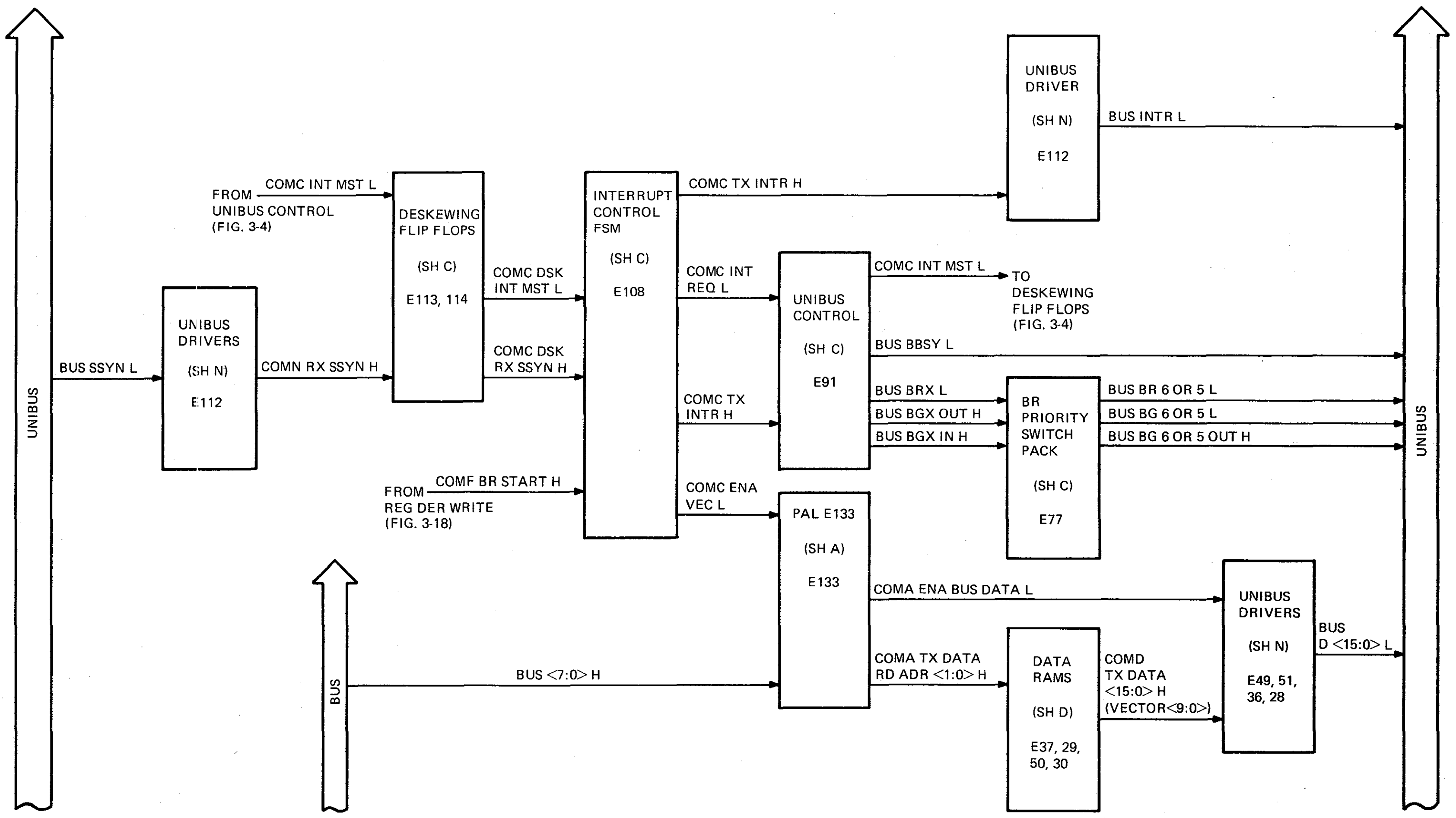

UNIBUS
BUS SSYN L
UNIBUS DRIVERS (SH N) E112
COMN RX SSYN H
FROM UNIBUS CONTROL (FIG. 3-4)
COMC INT MST L
DESKEWING FLIP FLOPS (SH C) E113, 114
COMC DSK INT MST L
COMC DSK RX SSYN H
INTERRUPT CONTROL FSM (SH C) E108
FROM REG DER WRITE (FIG. 3-18)
COMF BR START H
COMC TX INTR H
UNIBUS DRIVER (SH N) E112
BUS INTR L
COMC INT REQ L
COMC TX INTR H
UNIBUS CONTROL (SH C) E91
COMC INT MST L
TO DESKEWING FLIP FLOPS (FIG. 3-4)
BUS BBSY L
BUS BRX L
BUS BGX OUT H
BUS BGX IN H
BR PRIORITY SWITCH PACK (SH C) E77
BUS BR 6 OR 5 L
BUS BG 6 OR 5 L
BUS BG 6 OR 5 OUT H
COMC ENA VEC L
PAL E133 (SH A) E133
BUS
BUS <7:0> H
COMA ENA BUS DATA L
COMA TX DATA RD ADR <1:0> H
DATA RAMS (SH D) E37, 29, 50, 30
COMD TX DATA <15:0> H (VECTOR<9:0>)
UNIBUS DRIVERS (SH N) E49, 51, 36, 28
BUS D <15:0> L
UNIBUS
TK-9831

Figure 3-4 UNIBUS Interrupt Logic

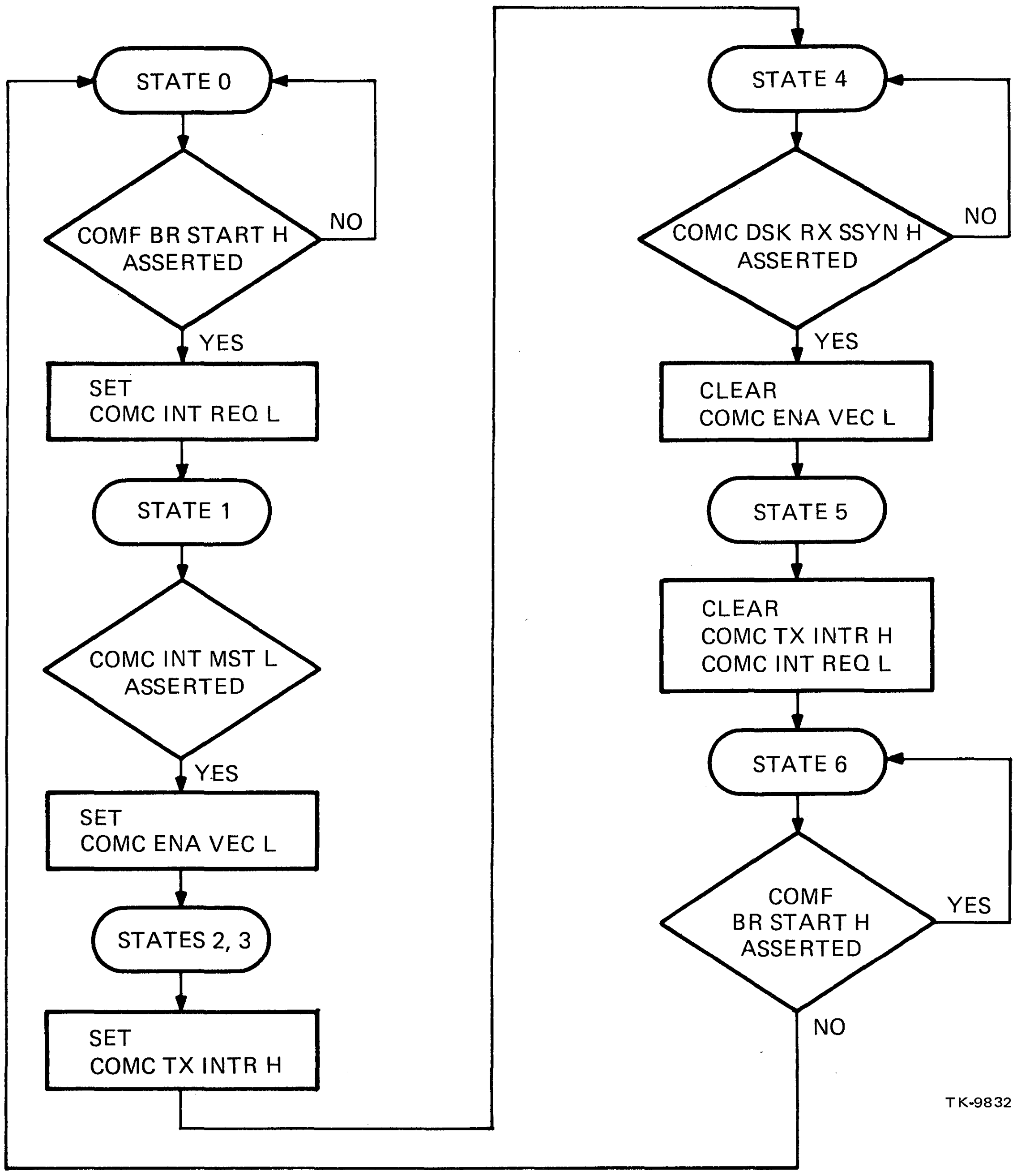


Figure 3-5 Interrupt Control FSM Flow

Table 3-6 Interrupt Control FSM (E103) Truth Table

INPUTS		PRESENT STATE						NEXT STATE				
INT MST	RX SSYN	BR STRT	STATE	TX INTR	EN VEC	INT REQ	S0	STATE	TX INTR	EN VEC	INT REQ	S0
X	X	0	X	X	X	X	X	0	0	1	1	1
X	X	1	0	0	1	1	1	1	0	1	0	1
1	X	1	1	0	1	0	1	1	0	1	0	1
0	X	1	1	0	1	0	1	2	0	0	0	1
X	X	1	2	0	0	0	1	3	0	0	0	0
X	X	1	3	0	0	0	0	4	1	0	0	1
X	0	1	4	1	0	0	1	4	1	0	0	1
X	1	1	4	1	0	0	1	5	1	1	0	1
X	X	1	5	1	1	0	1	6	0	1	1	0
X	X	1	6	0	1	1	0	6	0	1	1	0

The PAL (E133) produces COMA ENA BUS DATA L and COMA TX DATA RD ADR ⟨1:0⟩ H. COMA ENA BUS DATA L is applied to the UNIBUS drivers (E49,51,36,28) to enable the UNIBUS data drivers. COMA TX DATA RD ADR ⟨1:0⟩ H applies address 01 to the TX data RAMs (E37,29,50,30) to read the vector out of the TX data RAMs. The UNIBUS drivers apply the vector (BUS D ⟨15:0⟩ H) to the UNIBUS.

In state 3, the interrupt control FSM (E103) asserts COMC TX INTR H. The interrupt request (BUS INTR L) is applied to the UNIBUS. COMC TX INTR H is applied to the UNIBUS via the UNIBUS drivers (E112).

In state 4, the interrupt control FSM (E103) waits for the CPU to assert BUS SSYN L. BUS SSYN L is asserted and applied to the interrupt control FSM (E103) via UNIBUS receiver (E112) (COMN RX SSYN H) and deskewing flip-flops (E113,114) (COMC DSK RX SSYN H).

When in state 5, COMC DSK RX SSYN L is asserted, the interrupt control FSM (E103) deasserts COMC ENA VEC L. One state later (state 6) the interrupt control FSM (E103) deasserts both COMC TX INTR H and COMC INT REQ L. Next the UNIBUS control (E91) deasserts BUS BBSY L. The interrupt control FSM (E103) loops in state 6, waiting for COMF BR START H to be deasserted. When COMF BR START H is deasserted, the interrupt control FSM (E103) returns to state 0.

3.4 UNIBUS SLAVE I/O OPERATION

Microcode uses hardware traps to handle the UNIBUS slave I/O data transfers. The slave control FSM (E103) handles the UNIBUS handshaking protocol, while hardware trap routines handle data transfers to the UNIBUS [DATI(P) cycle] and handles data transfers from the UNIBUS holding registers [DATO(B) cycle].

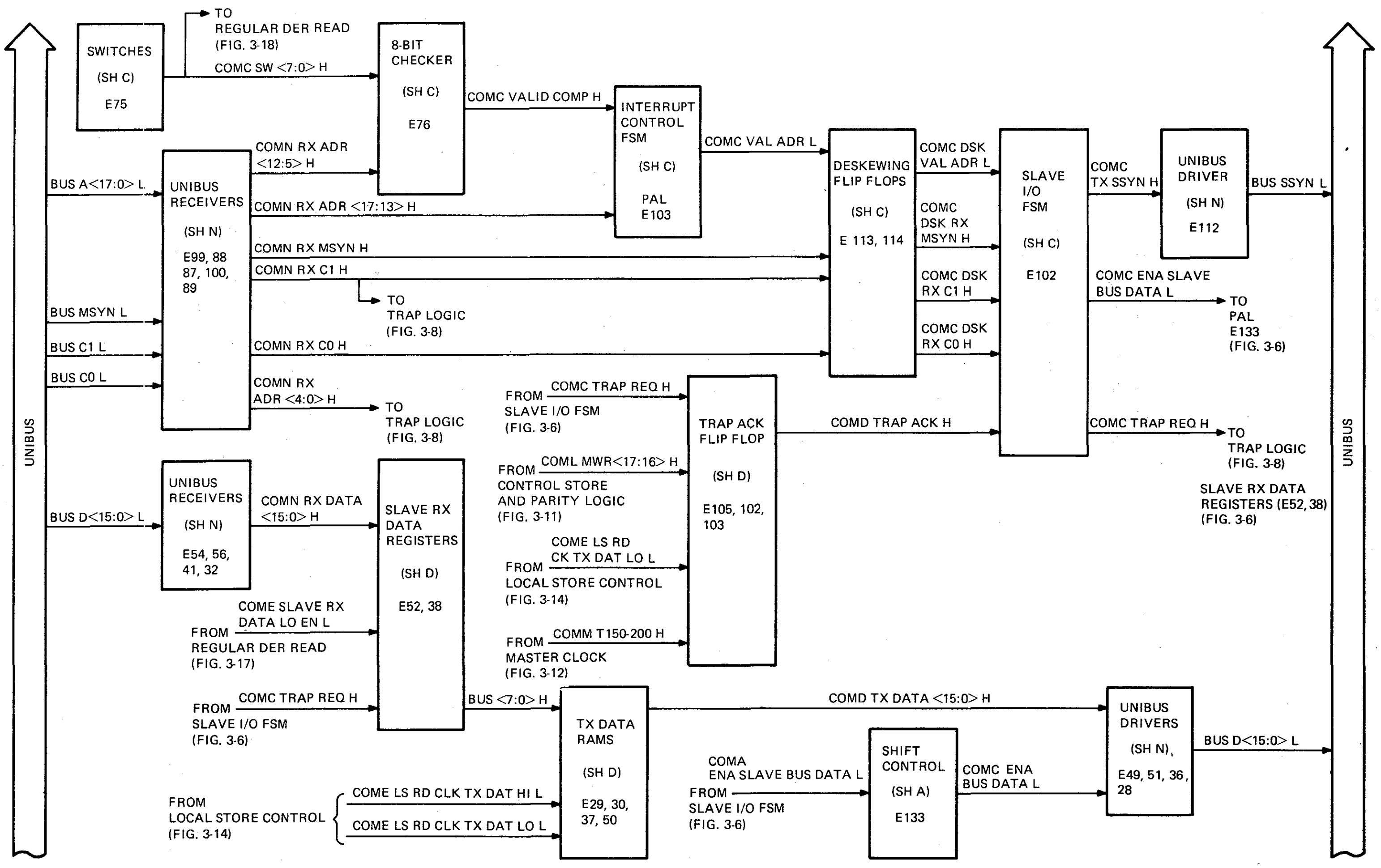
Refer to Figure 3-6 for the following descriptions of the slave DATO(B) and slave DATI(P) cycles.

3.4.1 Slave Device Addressing

When the interrupt control FSM (E103) asserts COMC VAL ADR L, a valid DMF32 CSR address is on the UNIBUS address lines. A valid DMF32 CSR address is detected in the following way.

The master device applies an address (BUS A ⟨17:0⟩ L) to the UNIBUS. The UNIBUS receivers (E99,88,87,100,89) apply the address (COMN RX ADR ⟨17:0⟩ H) to the interrupt control FSM (E103) (COMN RX ADR ⟨17:13⟩ H), eight-bit checker (E76) (COMN RX ADR ⟨12:5⟩ H), and the trap logic (COMN RX ADR ⟨4:0⟩ H). If address bits COMN RX ADR ⟨17:13⟩ H are all asserted (all ones), the interrupt control FSM (E103) determines that there is a legitimate reference to the UNIBUS I/O space.

The CSR dip switch (E75) contains the address (bits ⟨12:5⟩) for the DMF32 CSR. An eight-bit checker compares the CSR dip-switch address (COMC SW ⟨7:0⟩ H) with the UNIBUS address (COMN RX ADR ⟨12:5⟩ H). If the addresses are identical, the eight-bit checker (E76) asserts and applies COMC VALID COMP H to the interrupt control FSM (E103). The UNIBUS address bits COMN RX ADR ⟨4:0⟩ H are applied to the trap logic. With both COMC VALID COMP H and all the address bits ⟨17:13⟩ (COMN RX ADR ⟨17:13⟩ H) asserted, the interrupt control FSM (E103) asserts COMC VAL ADR L. COMC VAL ADR L is applied to the slave I/O FSM (E102) via the deskewing flip-flops (E113,114) (COMC DSK VAL ADR L). Refer to Table 3-7 for the truth table of the slave I/O FSM.



UNIBUS
SWITCHES (SH C) E75
TO REGULAR DER READ (FIG. 3-18)
COMC SW <7:0> H
8-BIT CHECKER (SH C) E76
COMC VALID COMP H
INTERRUPT CONTROL FSM (SH C) PAL E103
COMC VAL ADR L
DESKEWING FLIP FLOPS (SH C) E 113, 114
COMC DSK VAL ADR L
COMC DSK RX MSYN H
COMC DSK RX C1 H
COMC DSK RX C0 H
SLAVE I/O FSM (SH C) E102
COMC TX SSYN H
UNIBUS DRIVER (SH N) E112
BUS SSYN L
COMC ENA SLAVE BUS DATA L
TO PAL E133 (FIG. 3-6)
COMC TRAP REQ H
TO TRAP LOGIC (FIG. 3-8)
SLAVE RX DATA REGISTERS (E52, 38) (FIG. 3-6)
BUS A<17:0> L
UNIBUS RECEIVERS (SH N) E99, 88 87, 100, 89
COMN RX ADR <12:5> H
COMN RX ADR <17:13> H
COMN RX MSYN H
COMN RX C1 H
TO TRAP LOGIC (FIG. 3-8)
BUS MSYN L
BUS C1 L
BUS C0 L
COMN RX C0 H
COMN RX ADR <4:0> H
TO TRAP LOGIC (FIG. 3-8)
FROM SLAVE I/O FSM (FIG. 3-6)
COMC TRAP REQ H
TRAP ACK FLIP FLOP (SH D) E105, 102, 103
COMD TRAP ACK H
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COML MWR<17:16> H
COME LS RD CK TX DAT LO L
FROM LOCAL STORE CONTROL (FIG. 3-14)
COMM T150-200 H
FROM MASTER CLOCK (FIG. 3-12)
BUS D<15:0> L
UNIBUS RECEIVERS (SH N) E54, 56, 41, 32
COMN RX DATA <15:0> H
SLAVE RX DATA REGISTERS (SH D) E52, 38
COME SLAVE RX DATA LO EN L
FROM REGULAR DER READ (FIG. 3-17)
COMC TRAP REQ H
FROM SLAVE I/O FSM (FIG. 3-6)
BUS <7:0> H
TX DATA RAMS (SH D) E29, 30, 37, 50
COMD TX DATA <15:0> H
FROM LOCAL STORE CONTROL (FIG. 3-14)
COME LS RD CLK TX DAT HI L
COME LS RD CLK TX DAT LO L
COMA ENA SLAVE BUS DATA L
FROM SLAVE I/O FSM (FIG. 3-6)
SHIFT CONTROL (SH A) E133
COMC ENA BUS DATA L
UNIBUS DRIVERS (SH N), E49, 51, 36, 28
BUS D<15:0> L
TK-9833

Figure 3-6 UNIBUS Slave Control Logic

Table 3-7 Slave I/O FSM (E102) Truth Table

INPUTS					PRESENT STATE						NEXT STATE						
VAL ADR	RX MSYN	RX C1	TRAP ACK	PON	STATE	TX SSYN	TRAP REQ	EN DAT	S1	S0	STATE	TX SSYN	TRAP REQ	EN DAT	S1	S0	NOTE
X	X	X	X	0	X	X	X	X	X	X	0	0	0	1	1	1	1
0	1	X	X	1	0	0	0	1	1	1	1	0	1	1	1	1	2
X	X	X	0	1	1	0	1	1	1	1	1	0	1	1	1	1	3
X	X	0	1	1	1	0	1	1	1	1	2	0	0	0	1	1	4
X	X	X	X	1	2	0	0	0	1	1	3	0	0	0	0	1	5
X	X	X	X	1	3	0	0	0	0	1	4	1	0	0	1	1	6
X	1	X	X	1	4	1	0	0	1	1	4	1	0	0	1	1	7
X	0	X	X	1	4	1	0	0	1	1	5	1	0	1	1	1	8
X	X	X	X	1	5	1	0	1	1	1	6	1	0	1	0	1	9
X	X	X	X	1	6	1	0	1	0	1	0	0	0	1	1	1	10
X	X	1	1	1	1	0	1	1	1	1	7	1	0	1	1	0	11
X	1	X	X	1	7	1	0	1	1	0	7	1	0	1	1	0	12
X	0	X	X	1	7	1	0	1	1	0	0	0	0	1	1	1	13
1	X	X	X	1	0	0	0	1	1	1	0	0	0	1	1	1	
X	0	X	X	1	0	0	0	1	1	1	0	0	0	1	1	1	

1. Power on reset
2. Valid cycle
3. Wait for trap ack
4. Trap ack is set and it is a DATI(P) cycle, so enable data
5. Wait state
6. Set TX SSYN
7. Wait for RX MSYN to clear
8. Clear ENA slave data
9. Wait state
10. End of DATI(P) cycle
11. Valid DATO(B) cycle
12. Wait for RX MSYN to clear
13. Clear TX SSYN, end of DATO(B) cycle

3.4.2 Slave DATO(B) Cycle

Refer to Figure 3-6 and Figure 3-7 for the following description of the slave DATO(B) cycle.

In state 0, when COMC DSK VAL ADR L, COMC DSK RX MSYN H, and COMC DSK RX C1 H are all asserted and applied to the slave I/O FSM (E102), a DATO cycle initiation is indicated. With a DATO cycle indicated, the slave I/O FSM (E102) asserts COMC TRAP REQ H and waits for the assertion of COMD TRAP ACK H. COMC TRAP REQ H is applied to the trap logic to initiate a trap routine.

Also, COMC TRAP REQ H clocks the data on the UNIBUS data lines into the slave RX data registers (E52,38) via UNIBUS receivers (COMN RX DATA ⟨15:0⟩ H).

Trap microcode asserts COMD TRAP ACK H. COMD TRAP ACK H is applied to the slave I/O FSM (E102) to inform the slave I/O FSM (E102) that the trap microcode has read the data from the slave RX data registers (E52,38).

COME LS RD CLK TX DATA LO L, performed bit trap mode, clocks the NAND of COML MWR ⟨17:16⟩ H into the trap ACK flip-flop (E105). This asserts COMD TRAP ACK H that is applied to the slave I/O FSM (E102). Since COMC TRAP REQ H is the preset to the trap ACK flip-flop (E105), prior to COMC TRAP REQ H being asserted, COMD TRAP ACK L is deasserted.

With the assertion of COMD TRAP ACK H, the slave I/O FSM (E102) asserts COMC TX SSYN H and enters state 7. COMC TX SSYN H is applied to the bus master device via UNIBUS drivers (E112)(COMC TX SSYN H), and the UNIBUS (BUS SSYN H). Asserting COMC TX SSYN H informs the bus master that the DMF32 has received the data from the UNIBUS.

The bus master device deasserts BUS MSYN L to indicate that the bus master device considers the data transfer complete. BUS MSYN L is applied to the slave I/O FSM via the UNIBUS (BUS MSYN L), UNIBUS receivers (COMN RX MSYN L), and the deskewing flip-flops (COMC DSK RX MSYN H).

Applying the deasserted COMC DSK RX MSYN H to the slave I/O FSM causes the slave I/O FSM (E102) to deassert COMC TX SSYN H. The deasserted COMC TX SSYN H informs the bus master device that the DMF32 has concluded the data transfer. COMC TX SSYN H is applied to the bus master device via the UNIBUS driver (E112), and the UNIBUS (BUS SSYN L). Now the UNIBUS slave cycle handshaking is complete.

3.4.3 Slave DATI Cycle

Refer to Figure 3-6 and Figure 3-7 for the following description of the DATI cycle.

In state 0, when COMC DSK VAL ADR L and COMC DSK RX MSYN H are asserted, and COMC DSK RX C1 H is deasserted, the slave I/O FSM (E102) initiates the DATI cycle by asserting COMC TRAP REQ H. The slave I/O FSM (E102) is now in state 1. The trap microcode loads the TX data RAMS (E29,30) with the high byte of data from the local store via the bus (BUS ⟨7:0⟩ H) by asserting COME LS RD CLK TX DAT HI L. Next the microcode asserts COME LS RD CLK TX DAT LO L, that clocks the low byte of data into the TX data RAMs (E37,50) via the bus. COME LS RD CLK TX DAT LO L also clocks the trap ACK flip-flop (E105), that causes the trap ACK flip-flop (E105) to assert COMD TRAP ACK H.

The COMD TRAP ACK H is applied to the slave I/O FSM (E102). COMD TRAP ACK H informs the slave I/O FSM that the requested data in the TX data RAMs can be applied to the UNIBUS. The slave I/O FSM (E102) asserts COMC ENA SLAVE BUS DATA L to apply data to the UNIBUS data lines. COMC ENA SLAVE BUS DATA L is applied to PAL E133, that produces COMA ENA BUS DATA L. COMA ENA BUS DATA L is applied to the UNIBUS drivers that apply the data to the UNIBUS.

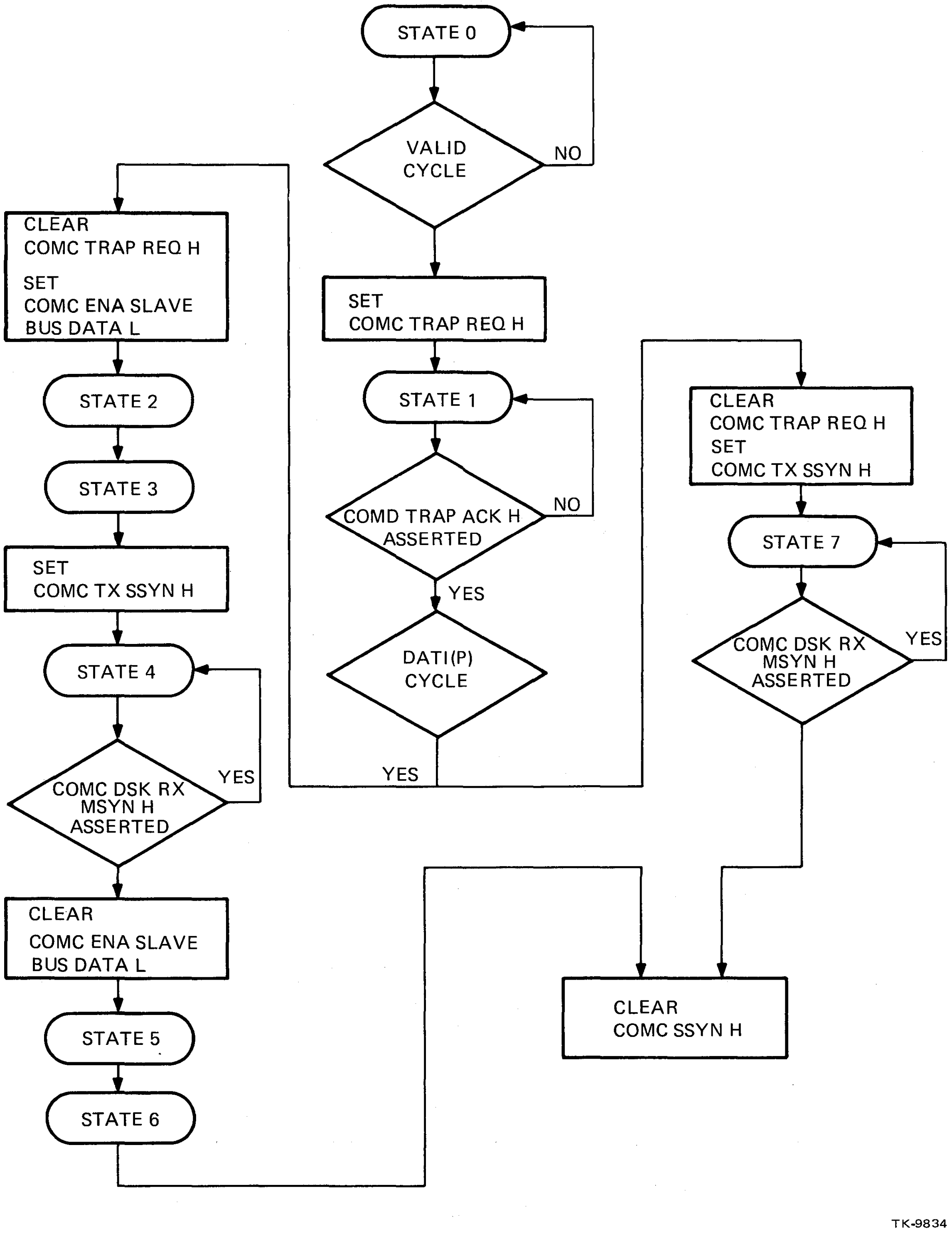


Figure 3-7 Slave Control FSM Flow

The slave I/O FSM (E102) enters state 2. The next state, state 3 is a wait state. State 3 provides the time to apply the data to the UNIBUS before the slave I/O FSM (E102) asserts COMC TX SSYN H. The slave I/O FSM (E102) asserts COMC TX SSYN H. COMC TX SSYN H is applied to the master device via the UNIBUS drivers (E112) and the UNIBUS (BUS SSYN L). The slave I/O FSM (E102) is now in state 4.

Looping in state 4, the slave I/O FSM (E102) waits for the master device to deassert BUS MSYN L. The master device deasserts BUS MSYN L to indicate to the DMF32 that the bus master device considers the data transfer complete. BUS MSYN L is applied to the slave I/O FSM (E102) via the UNIBUS receivers (E112) (COMC RX MSYN L) and the deskewing flip-flops (E113,114) (COMC DSK RX MSYN H).

After the slave I/O FSM (E102) detects COMC DSK RX MSYN H being deasserted, the slave I/O FSM (E102) deasserts COMC ENA SLAVE BUS DATA L to remove the data from the UNIBUS and enters state 5. Next, state 6 is entered. State 6 is a wait state that provides the time to remove the data from the UNIBUS before COMC TX SSYN H is deasserted.

The slave I/O FSM (E102) deasserts COMC TX SSYN H and COMC TRAP REQ H. The deasserted COMC TX SSYN informs the bus master device that the DMF32 has concluded the data transfer. COMC TX SSYN H is applied to the bus master device via the UNIBUS drivers (E112) and the UNIBUS (BUS SSYN L). The slave I/O FSM (E102) enters state 0 to complete the slave DATI cycle.

3.4.4 Slave DATIP Cycle

A DATIP cycle is a DATI cycle followed by a DATO(B). The DMF32 uses two separate UNIBUS cycles to perform a DATIP. After asserting COMD TRAP ACK H, the trap microcode examines COMB REG C0 H to distinguish between a DATIP cycle and a DATI cycle.

If a DATIP cycle (COMB REG C0 H is asserted) and is to be performed, the microcode asserts COMF INHIBIT PUP H, enables traps, and then jumps to a wait loop. The DATO(B) cycle traps to this wait loop. COMF INHIBIT PUP H is asserted to inhibit pushing the stack when the DATO(B) trap occurs. The DATO(B) trap code clears COMF INHIBIT PUP H prior to returning from the trap code.

3.5 TRAP LOGIC

The trap logic circuit consists of a trap control FSM (E110), multiplexers (E115,116), data I/O status register (E101), and register C0 flip-flop (E150).

The trap logic handles the trap requests and transfers control to the trap routines. The trap control FSM (E110) controls entry to the trap routines, while the multiplexers (E115,116) provide the trap addresses. There are 32 possible trap addresses, ranging from FE0 to FFF (hex).

3.5.1 Trap Control

The trap control FSM (E110) controls the traps. COML MWR ⟨25:24⟩ H is applied to the trap control FSM (E110) to enable traps or disable traps during a DER microinstruction. Refer to Table 3-8 for the COML MWR ⟨25:24⟩ H bit configurations for trap control.

Table 3-8 Trap Control

COML MWR 25	COML MWR 24	Trap Control
0	0	no change
0	1	disable traps
1	X	enable traps

During a DER microinstruction, the no-change encoding has no effect on the trap control FSM (E110). A disable traps command inhibits any further traps from occurring until a subsequent enable traps command is performed. The microinstruction that disables traps cannot be trapped. An enable traps command enables the traps.

COML MWR ⟨7:5⟩ H is also applied to the trap control FSM (E110). Refer to Figure 3-8. The trap control FSM (E110) decodes COML MWR ⟨7:5⟩ H to determine if a DER instruction is being performed. During power-up, COMD DSK TRAP REQ H is applied to the trap control FSM (E110) to disable traps. When the trap control FSM (E110) asserts COMB TRAP L, a microtrap is initiated. Refer to Table 3-9 for the truth table of the trap control FSM (E110).

The slave I/O FSM (E102) initiates the trap request by asserting COMC TRAP REQ H and applying COMC TRAP REQ H to the trap request flip-flop (E105). At time T200, COMC TRAP REQ H is clocked into the trap request flip-flop (E105). The trap request flip-flop (E105) deskews COMC TRAP REQ H and applies COMD DSK TRAP REQ H to the trap control FSM (E110). Applying COMD DSK TRAP REQ H to the trap control FSM (E110) causes the trap control FSM (E110) to assert COMB TRAP L for one microcycle if trap control FSM (E110) is in the enable traps state. Refer to Figure 3-9 for trap control FSM (E110) state diagram.

3.5.2 Trap Addresses

The trap control FSM (E110) asserts COMB TRAP L to initiate the trap cycle. COMB TRAP L is applied to the multiplexers (E115,116) to select the trap microaddress. Applying COMB TRAP L to the multiplexers (E115,116) causes the multiplexers (E115,116) to select the data I/O status register (E101) rather than COMB 2911 ADR ⟨4:0⟩ H from the microsequencers, for the low five bits of the next microaddress.

The data I/O status register contains COMN RX ADR ⟨4:1⟩ H from the UNIBUS receivers and COMN RX C1 H, a UNIBUS control bit, from the UNIBUS receivers. These five bits provide the low five bits of the next microaddress, while the multiplexers (E115,116) assert the next higher three bits (COMB MW ADR ⟨7:5⟩ H). The three most significant bits of the microaddress are produced by COMB TRAP H being applied to the microsequencer (E106). COMB TRAP H forces the output of the microsequencer (E106) to tri-state (high). Thus, the eleven bits of the next microaddress consist of five lower bits from the data I/O status register (E101); the next higher three bits are from the multiplexers (E115,116) (all high); and the most significant three bits are from the microsequencer (E106) (all high).

3.5.3 The Trap Routine

The next microinstruction to be executed is the first microinstruction of the trap routine. COMB TRAP L is also applied to the microsequencers (E106,118,119) to prevent the incrementer in the microsequencer from incrementing. The address of the microinstruction that would have been performed if there were no trap is clocked into the microprogram counter of the microsequencers (E106,118,119) at end of the cycle. Therefore, the first microinstruction of the trap code pushes the microprogram counter onto the stack. This microinstruction must also jump out of the 32-word jump table. Jumping and pushing the stack is done by performing a jump to subroutine microinstruction. During a trap operation, the trap control FSM automatically enters the disable trap state.

The UNIBUS receivers apply COMN RX C0 H to the data I/O status register (E101). From the data I/O status register (E101), COMN RX C0 H is clocked into the register C0 flip flop by COMB TRAP DISABLE STATE H. The DATI trap microcode examines the bit COMB REG C0 H to distinguish between the DATI and DATIP cycle.

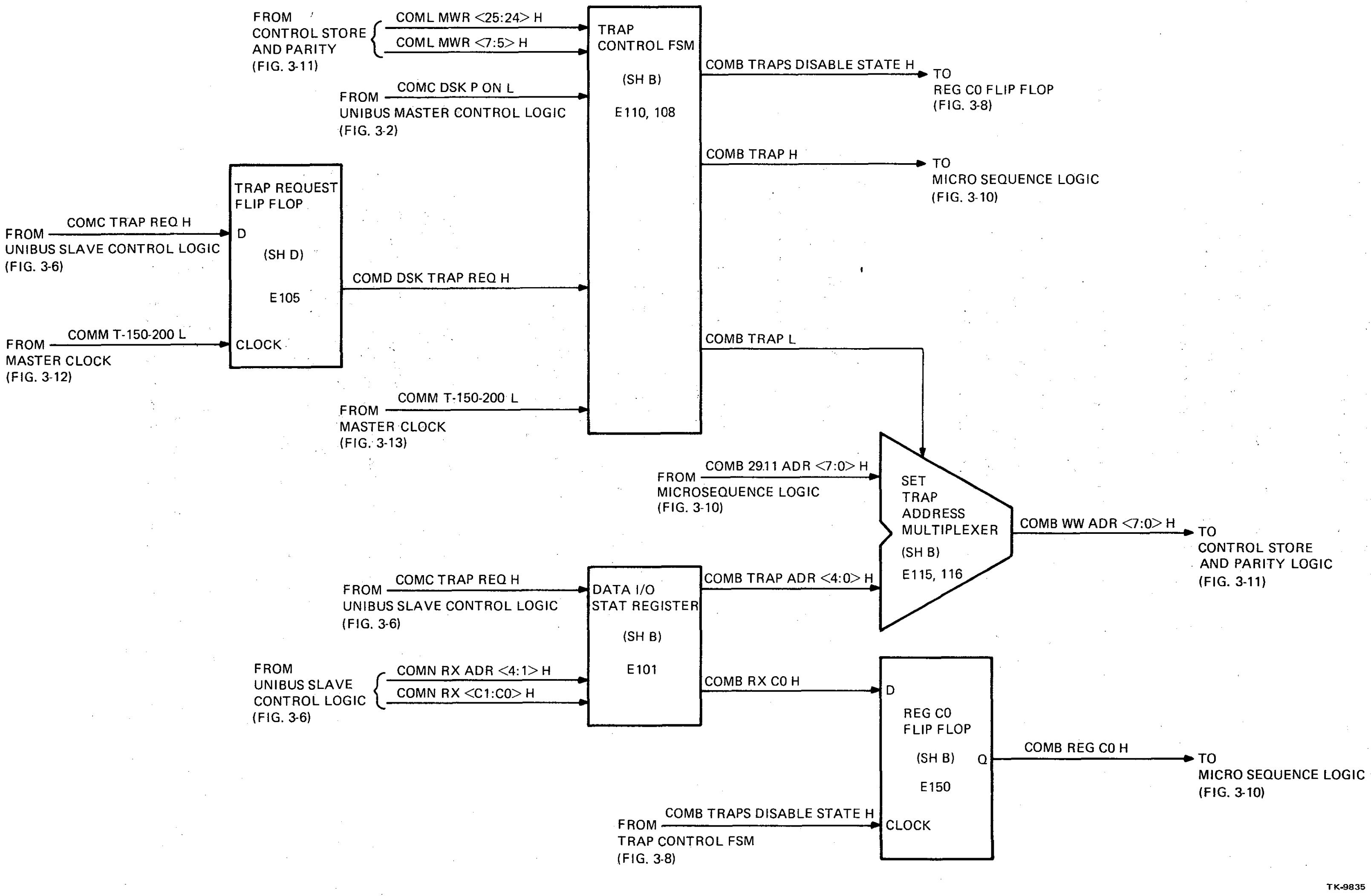
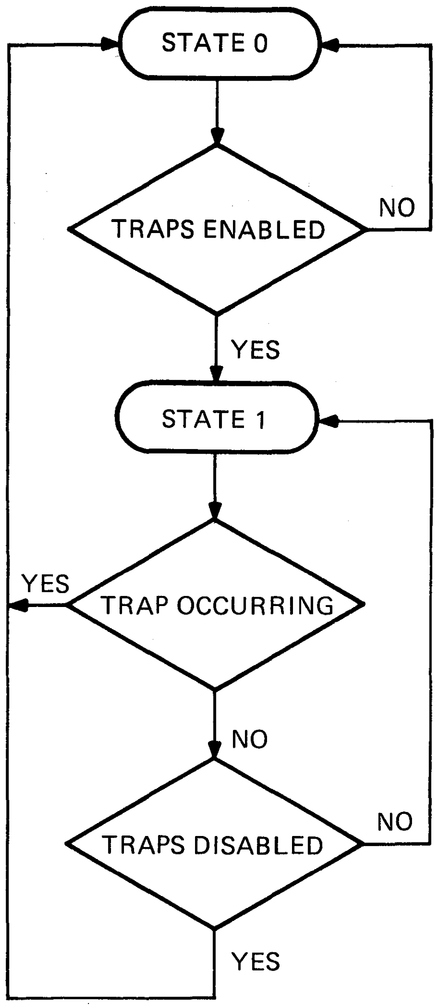

FROM CONTROL STORE AND PARITY (FIG. 3-11)
COML MWR <25:24> H
COML MWR <7:5> H
TRAP CONTROL FSM
(SH B)
E110, 108
COMC DSK P ON L
FROM UNIBUS MASTER CONTROL LOGIC (FIG. 3-2)
COMB TRAPS DISABLE STATE H
TO REG C0 FLIP FLOP (FIG. 3-8)
COMB TRAP H
TO MICRO SEQUENCE LOGIC (FIG. 3-10)
TRAP REQUEST FLIP FLOP
D
(SH D)
E105
CLOCK
COMC TRAP REQ H
FROM UNIBUS SLAVE CONTROL LOGIC (FIG. 3-6)
COMD DSK TRAP REQ H
COMM T-150-200 L
FROM MASTER CLOCK (FIG. 3-12)
COMB TRAP L
COMM T-150-200 L
FROM MASTER CLOCK (FIG. 3-13)
COMB 2911 ADR <7:0> H
FROM MICROSEQUENCE LOGIC (FIG. 3-10)
SET TRAP ADDRESS MULTIPLEXER
(SH B)
E115, 116
COMB WW ADR <7:0> H
TO CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COMC TRAP REQ H
FROM UNIBUS SLAVE CONTROL LOGIC (FIG. 3-6)
DATA I/O STAT REGISTER
(SH B)
E101
COMB TRAP ADR <4:0> H
FROM UNIBUS SLAVE CONTROL LOGIC (FIG. 3-6)
COMN RX ADR <4:1> H
COMN RX <C1:C0> H
COMB RX C0 H
D
REG C0 FLIP FLOP
(SH B)
Q
E150
COMB REG C0 H
TO MICRO SEQUENCE LOGIC (FIG. 3-10)
COMB TRAPS DISABLE STATE H
FROM TRAP CONTROL FSM (FIG. 3-8)
CLOCK
TK-9835

Figure 3-8 Trap Logic

Table 3-9 Trap Control FSM Truth Table

INPUTS					PRESENT STATE		O/P	NEXT STATE	
COML MWR 25	24	7	6	5	COMC TRAP REQ H	S0	COMC TRAP L	S0	
1	X	1	0	1	X	1	1	0	ENA TRAPS
0	X	1	0	1	X	1	1	1	STAY DISAB
X	X	0	X	X	X	1	1	1	STAY DISAB
X	X	X	1	X	X	1	1	1	STAY DISAB
X	X	X	X	0	X	1	1	1	STAY DISAB
1	X	1	0	1	1	0	0	1	TRAP IN PROG
X	0	1	0	1	1	0	0	1	TRAP IN PROG
1	X	1	0	1	0	0	1	0	STAY ENA
X	0	1	0	1	0	0	1	0	STAY ENA
0	1	1	0	1	X	0	1	1	DISAB TRAPS
X	X	0	X	X	1	0	0	1	TRAP IN PROG
X	X	X	1	X	1	0	0	1	TRAP IN PROG
X	X	X	X	0	1	0	0	1	TRAP IN PROG
X	X	0	X	X	0	0	1	0	STAY ENA
X	X	X	1	X	0	0	1	0	STAY ENA
X	X	X	X	0	0	0	1	0	STAY ENA



TK-9836

Figure 3-9 Trap Control FSM States

3.6 MICROSEQUENCE LOGIC

The microsequence logic consists of a microsequence control PAL (E132) and three 2911 microprogram sequencers. The three cascaded 2911 microprogram sequencers provide a 12-bit address. The 12-bit microprogram address addresses the 4K words of the PROM control store to sequence through a series of microinstructions.

Refer to Figure 3-10 for the following description of the microsequence logic.

3.6.1 Microsequence Control

The microsequence control (E132) controls the three 2911 microprogram sequencers. The outputs from the microsequence control (E132), that are applied to the three 2911 microprogram sequencers, determine the data source for the next microinstruction address. The selected data source, which is the address of the next microword from the PROM control store, is applied to the bus from the 2911 microprogram sequencers.

The inputs to the microsequence control (E132) are defined in Table 3-10. Table 3-13 is the truth table for the microsequence control (E132).

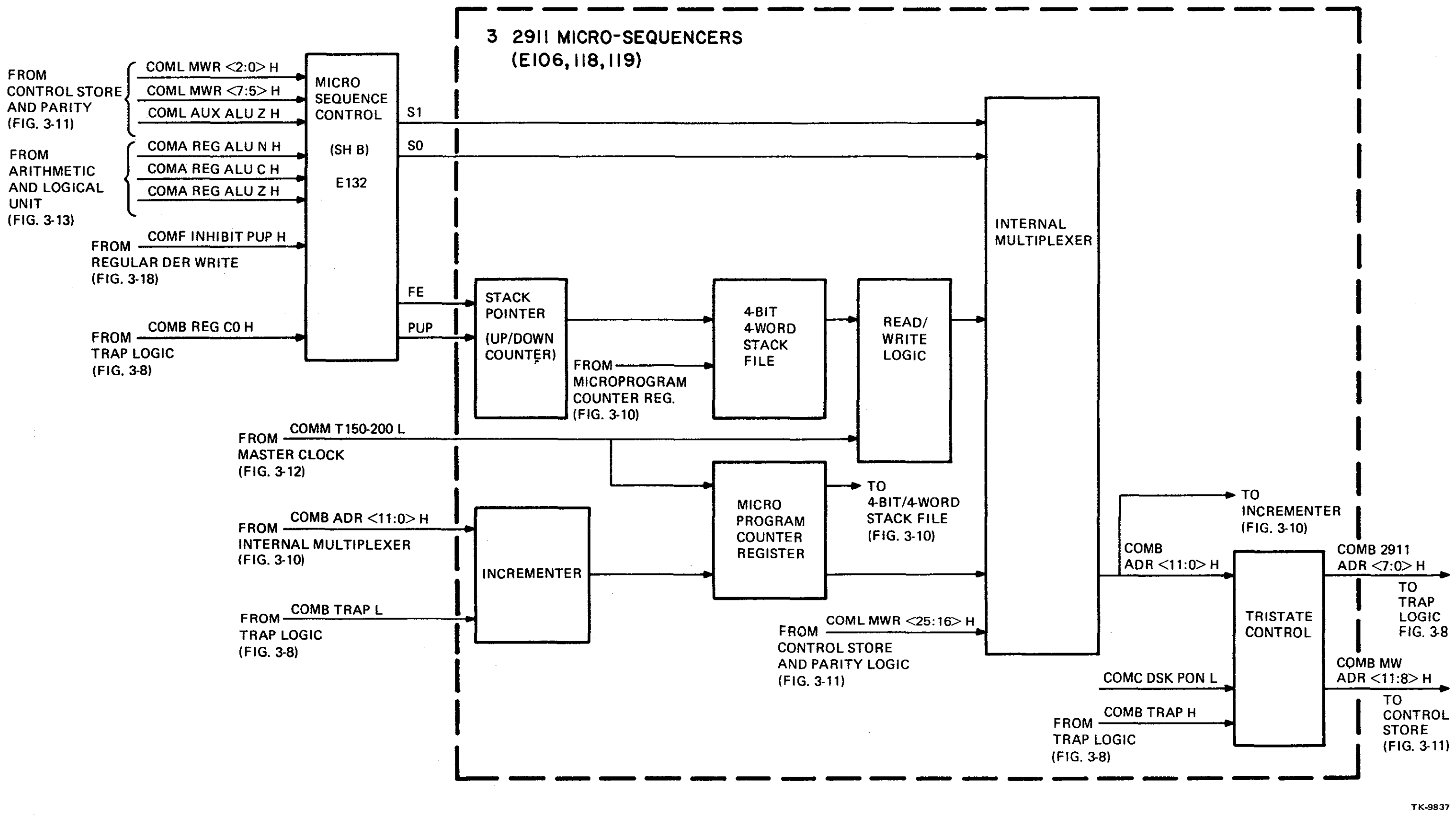

3 2911 MICRO-SEQUENCERS (E106, 118, 119)
FROM CONTROL STORE AND PARITY (FIG. 3-11)
COML MWR <2:0> H
COML MWR <7:5> H
COML AUX ALU Z H
FROM ARITHMETIC AND LOGICAL UNIT (FIG. 3-13)
COMA REG ALU N H
COMA REG ALU C H
COMA REG ALU Z H
FROM REGULAR DER WRITE (FIG. 3-18)
COMF INHIBIT PUP H
FROM TRAP LOGIC (FIG. 3-8)
COMB REG C0 H
MICRO SEQUENCE CONTROL (SH B) E132
S1
S0
FE
PUP
STACK POINTER (UP/DOWN COUNTER)
4-BIT 4-WORD STACK FILE
READ/ WRITE LOGIC
INTERNAL MULTIPLEXER
FROM MICROPROGRAM COUNTER REG. (FIG. 3-10)
FROM MASTER CLOCK (FIG. 3-12)
COMM T150-200 L
MICRO PROGRAM COUNTER REGISTER
TO 4-BIT/4-WORD STACK FILE (FIG. 3-10)
FROM INTERNAL MULTIPLEXER (FIG. 3-10)
COMB ADR <11:0> H
INCREMENTER
FROM TRAP LOGIC (FIG. 3-8)
COMB TRAP L
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COML MWR <25:16> H
TO INCREMENTER (FIG. 3-10)
COMB ADR <11:0> H
TRISTATE CONTROL
COMC DSK PON L
FROM TRAP LOGIC (FIG. 3-8)
COMB TRAP H
COMB 2911 ADR <7:0> H
TO TRAP LOGIC FIG. 3-8
COMB MW ADR <11:8> H
TO CONTROL STORE (FIG. 3-11)
TK-9837

Figure 3-10 Microsequence Logic

Table 3-10 Microsequence Control Inputs

Input	Definition
COML MWR ⟨2:0⟩ H	For a conditional jump microinstruction (i.e. COML MWR ⟨7:5⟩ H equals "11"), COML MWR ⟨2:0⟩ H has the meanings listed in Table 3-11. For any microinstruction other than a conditional jump, COML MWR ⟨2:0⟩ H has the meanings listed in Table 3-12.
COML MWR ⟨7:5⟩ H	These three bits determine the type of microinstruction to be performed.
COMA REG ALU N H COMA REG ALU Z H COMA REG ALU C H	These three bits are registered ALU condition codes.
COMA AUX ALU Z H	COML AUX ALU Z H is also the registered ALU zero bit, however, this bit is different from COMA REG ALU Z H in the following way. COMA REG ALU Z H is clocked into the ALU condition code register (with the other condition codes) depending upon the status of COML MWR ⟨3⟩ H, while COML AUX ALU Z H is always clocked into a flip-flop at the end of each microinstruction. The COML AUX ALU Z H bit is the only ALU condition used by TRAP microcode.
COMF INHIBIT PUP H	When the microcode sets this bit, pushing and popping of the stack is inhibited. This bit is used with an interlocked slave read-modify-write UNIBUS cycle.
COMB REG C0 H	This bit is the UNIBUS C0 control bit that is tested by the trap microcode.

Table 3-11 COML MWR ⟨2:0⟩ H (Conditional Jump)

COML MWR ⟨2:0⟩ H			
2	1	0	Condition
0	0	0	REG ALU N
0	0	1	REG ALU Z
0	1	0	REG ALU C
0	1	1	AUX ALU Z
1	0	0	DATIO C0
1	0	1	Unconditional jump to subroutine (JSB)
1	1	0	Unconditional jump and pop stack
1	1	1	Unconditional jump

Table 3-12 COML MWR ⟨2:0⟩ H (Non-Conditional Jump)

COML MWR ⟨2:0⟩ H			
2	1	0	next address source
0	0	0	microprogram counter
0	0	1	microprogram counter
0	1	0	microprogram counter
0	1	1	microprogram counter
1	0	0	stack then POP (RSB)
1	0	1	stack then POP (RSB)
1	1	0	stack then POP (RSB)
1	1	1	stack then POP (RSB)

Table 3-13 Microsequence Control Truth Table

COML MWR 7 6	COML MWR 2 1 0	COMA REG ALU N Z C	COMA AUX ALU Z	COMB REG C0	COMF INHIBIT PUP	OUTPUTS S1	S0	FE	PUP	
1 1	0 0 0	0 X X	X	X	X	0	0	1	X	ALU N UNSUC
1 1	0 0 0	1 X X	X	X	X	1	1	1	X	ALU N SUC
1 1	0 0 1	X 0 X	X	X	X	0	0	1	X	ALU Z UNSUC
1 1	0 0 1	X 1 X	X	X	X	1	1	1	X	ALU Z SUC
1 1	0 1 0	X X 0	X	X	X	0	0	1	X	ALU C UNSUC
1 1	0 1 0	X X 1	X	X	X	1	1	1	X	ALU C SUC
1 1	0 1 1	X X X	0	X	X	0	0	1	X	AUX Z UNSUC
1 1	0 1 1	X X X	1	X	X	1	1	1	X	AUX Z SUC
1 1	1 1 1	X X X	X	X	X	1	1	1	X	UNCONDX JUMP
1 1	1 0 1	X X X	X	X	1	1	1	1	X	UNCONDX JUMP
1 1	1 0 1	X X X	X	X	0	1	1	0	1	UNCONDX JSB
1 1	1 1 0	X X X	X	X	1	1	1	1	X	UNCONDX JUMP
1 1	1 1 0	X X X	X	X	0	1	1	0	0	UCD JMP & POP
1 1	1 0 0	X X X	X	0	X	0	0	1	X	DAT C0 UNSUC
1 1	1 0 0	X X X	X	1	X	1	1	1	X	DAT C0 SUC
0 X	1 X X	X X X	X	X	0	1	0	0	0	STK & POP
X 0	1 X X	X X X	X	X	0	1	0	0	0	STK & POP
0 X	1 X X	X X X	X	X	1	1	0	1	X	STK
X 0	1 X X	X X X	X	X	1	1	0	1	X	STK
0 X	0 X X	X X X	X	X	X	0	0	1	X	UPC
X 0	0 X X	X X X	X	X	X	0	0	1	X	UPC

3.6.2 Address Source Selection

Sl and SO are select lines. These select lines are applied to the internal multiplexer of the 2911 microprogram sequencers from the microsequencer control (E132). The internal multiplexer selects one of the three address sources for the next microaddress source. Table 3-14 lists the microaddress source selection.

Table 3-14 Microaddress Source Selection

S1	S0	Address Source
0	0	microprogram counter
0	1	not used
1	0	stack
1	1	direct input (COML MWR ⟨25:16⟩ H

When the direct input is selected the address in the microword is applied to the bus. The direct input is applied to the bus via the 2911 internal multiplexer and tri-state control.

When the microprogram counter is selected, the address from the internal multiplexer is applied to the incrementer. The incrementer increments the address and applies the address to the microprogram counter, that applies the address to the bus via the internal multiplexer and the tri-state control.

If there is a trap request, COMB TRAP L is asserted. Asserting COMB TRAP L prevents the incrementer from incrementing. It also prevents the address of the microinstruction that would have been executed if there were no trap request from incrementing; this is clocked into the microprogram counter at the end of the cycle. The first microinstruction of the trap code pushes the microprogram counter onto the stack.

The 12-bit/four-word stack file can be selected as a source to the internal multiplexer. The stack file provides return address linkage, when subroutines are being executed. The stack pointer always points to the last word written into the file. The stack pointer operates as an up/down counter with separate push/pop (PUP) input and file enable (FE) input.

The two bits PUP and FE are applied to the stack pointer up/down counter from the microsequence control (E132). These two bits determine if the stack is either pushed or popped, or no change. Refer to Table 3-15 for PUP/FE stack operation selection.

Table 3-15 PUP/FE Stack Operation Selection

FE	PUP	Operation
0	0	POP stack (decrement stack pointer)
0	1	push stack (increment stack pointer then push microprogram counter onto stack)
1	X	no change

When FE is clear (low) and the PUP is set (high), the push operation is enabled. The stack pointer is incremented and the stack file is written with the required return address. The return address is the next microinstruction address following the subroutine jump that initiated the push operation.

When FE and PUP are both clear (low), a pop operation is performed. The return address on the stack is used to return from the subroutine. At the next low to high transition of COMM T150–T200 L, the stack pointer is decremented.

If FE is set (high), the stack pointer is not incremented or decremented, regardless if PUP is set (high) or cleared (low).

3.7 CONTROL STORE AND PARITY LOGIC
The microword originates in the PROM control store. Refer to Figure 3-11. The PROM control store consists of nine 4K × 4 PROMs. These PROMs provide a 36-bit wide word (35 control bits and 1 even parity bit). The microword COML MW ⟨35:0⟩ H is read out of the PROM control store when the microsequencer addresses (COMB MW ADR ⟨11:0⟩ H) the PROM control store. From the PROM control store the microword is read into the microword registers at time T200. COMC DSK P ON L and COMA ALU Z H are also clocked into microword registers at time COMM T150 to 200 L.

When the microword is read out of the PROM control store, the microword is also applied to the parity checkers. COMF FATAL FAULT H from the eight-bit latches is also applied to the parity checkers. At time T150, the output of the parity checker is applied to the device sick flip-flop.

Whenever there is no parity error, that is COML DEVICE SICK H is deasserted, the green LED remains on. If a parity error is detected, that is COML DEVICE SICK H is asserted, then the green LED is not lit and microinstructions stop executing. COML DEVICE SICK H is also applied to the master clock, master control FSM and interrupt control FSM, so that these FSMs will be initialized in the event of a parity error.

3.8 MASTER CLOCK
The master clock circuitry provides the internal timing for the DMF32.

Refer to Figure 3-12 for the following description of the master clock circuitry.

The 40.0 MHz oscillator (E152) provides the reference signal for all the clock signals, except for the baud rate generators. The 40.0 MHz signal is applied to the divide-by-four flip-flops (E136,122). The divide-by-four flip-flops (E136,122) divide and buffer the 40 MHz signal. The divide-by-four flip-flops produce COMM CLK H.

COMM CLK H, a 10 MHz square wave, is applied to the clock phase generator (E151,122,123,129,83). COMM CLK H causes the clock phase generator to produce the clock phases. The clock phases provide the internal timing for the DMF32. The phases are as follows:

- COMM T0–50 H
- COMM T50+150 H
- COMM T100–200 H
- COMM T150–200 L
- COMM T100–150 H
- COMM T50–150 L
- COMM T50–150 H

When either COMN RX DC LO H or COML DEVICE SICK H is asserted, the clock phase generator is inhibited except for the COMM T50+150 H output. COMM T50+150 H remains active to be applied to the interrupt control FSM (E103), master control FSM (E78), and slave I/O FSM (E102) to prevent hanging-up the UNIBUS.

COMN RX DC LO H is asserted whenever BUS DC LO L is asserted on the UNIBUS. BUS DC LO L is asserted to inform bus devices that the dc power is about to fail. When a parity error is detected, COML DEVICE SICK H is asserted. COML DEVICE SICK H remains asserted until COMC DSK P ON L is asserted. COMC DSK P ON L is asserted whenever either COMN RX DC LO L or COMN RX INIT H (jumper W1 must be installed) is asserted.

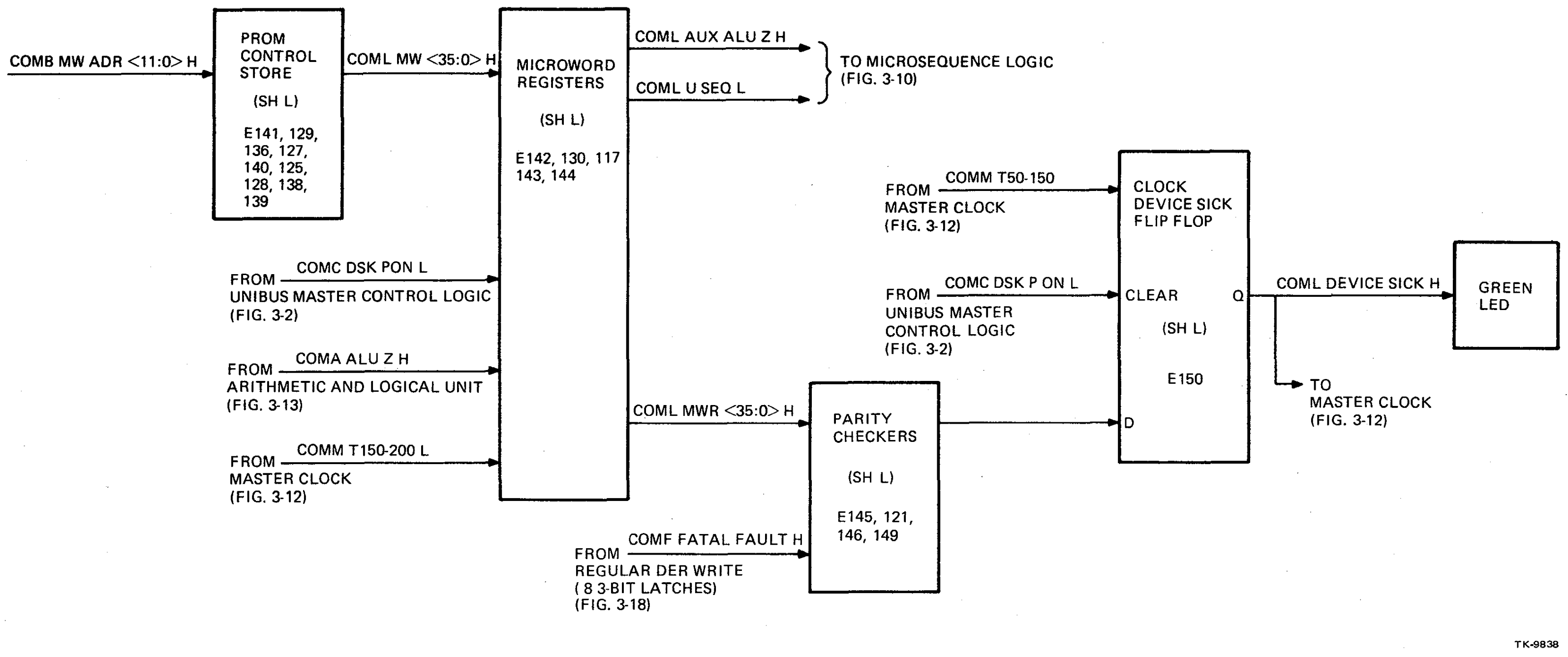

COMB MW ADR <11:0> H
PROM CONTROL STORE
(SH L)
E141, 129, 136, 127, 140, 125, 128, 138, 139
COML MW <35:0> H
MICROWORD REGISTERS
(SH L)
E142, 130, 117 143, 144
COML AUX ALU Z H
COML U SEQ L
TO MICROSEQUENCE LOGIC (FIG. 3-10)
FROM UNIBUS MASTER CONTROL LOGIC (FIG. 3-2)
COMC DSK PON L
FROM ARITHMETIC AND LOGICAL UNIT (FIG. 3-13)
COMA ALU Z H
FROM MASTER CLOCK (FIG. 3-12)
COMM T150-200 L
COML MWR <35:0> H
PARITY CHECKERS
(SH L)
E145, 121, 146, 149
FROM REGULAR DER WRITE (8 3-BIT LATCHES) (FIG. 3-18)
COMF FATAL FAULT H
FROM MASTER CLOCK (FIG. 3-12)
COMM T50-150
FROM UNIBUS MASTER CONTROL LOGIC (FIG. 3-2)
COMC DSK P ON L
CLOCK DEVICE SICK FLIP FLOP
CLEAR
Q
D
(SH L)
E150
COML DEVICE SICK H
GREEN LED
TO MASTER CLOCK (FIG. 3-12)
TK-9838

Figure 3-11 Control Store and Parity Logic

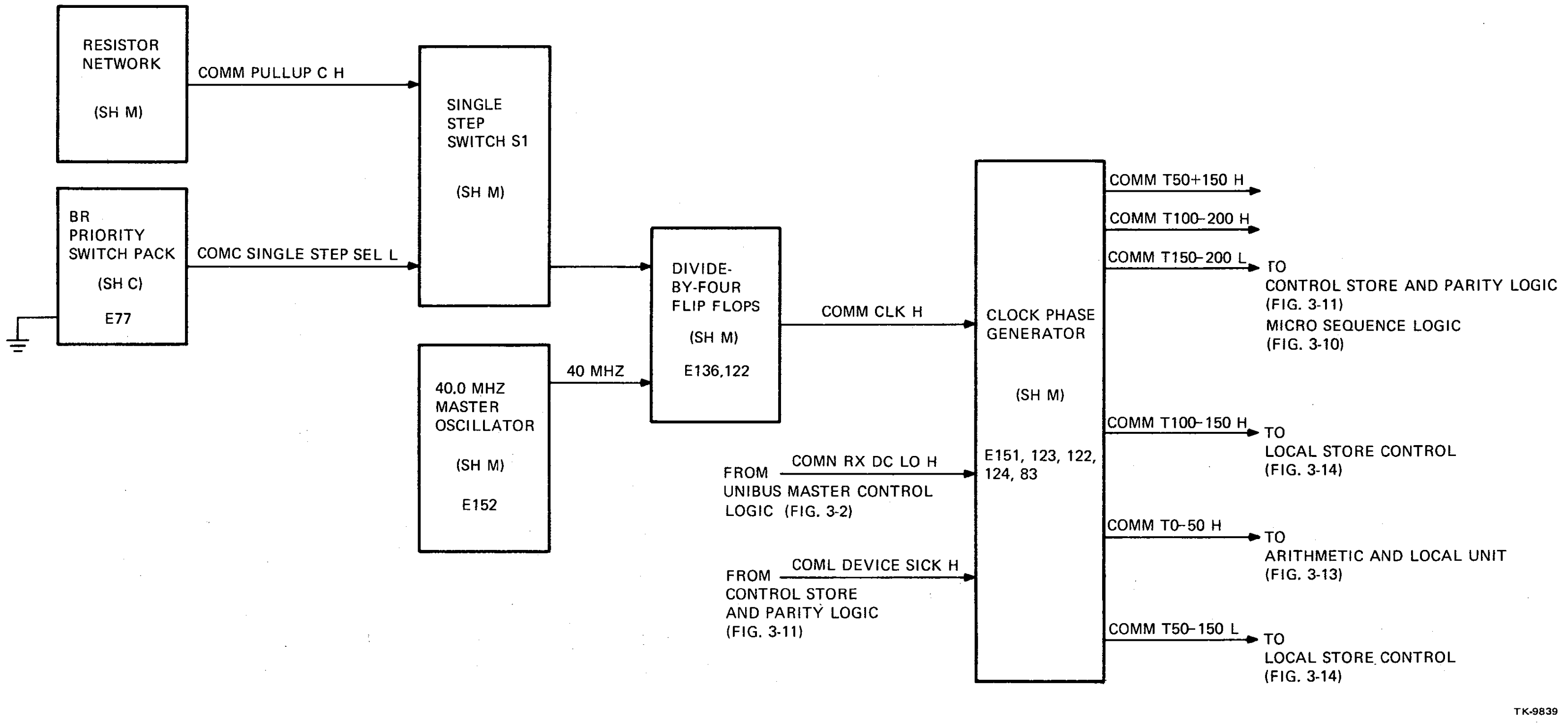

RESISTOR NETWORK (SH M)
COMM PULLUP C H
SINGLE STEP SWITCH S1 (SH M)
BR PRIORITY SWITCH PACK (SH C) E77
COMC SINGLE STEP SEL L
40.0 MHZ MASTER OSCILLATOR (SH M) E152
40 MHZ
DIVIDE-BY-FOUR FLIP FLOPS (SH M) E136,122
COMM CLK H
CLOCK PHASE GENERATOR (SH M) E151, 123, 122, 124, 83
FROM UNIBUS MASTER CONTROL LOGIC (FIG. 3-2)
COMN RX DC LO H
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COML DEVICE SICK H
COMM T50+150 H
COMM T100-200 H
COMM T150-200 L
TO CONTROL STORE AND PARITY LOGIC (FIG. 3-11) MICRO SEQUENCE LOGIC (FIG. 3-10)
COMM T100-150 H
TO LOCAL STORE CONTROL (FIG. 3-14)
COMM T0-50 H
TO ARITHMETIC AND LOCAL UNIT (FIG. 3-13)
COMM T50-150 L
TO LOCAL STORE CONTROL (FIG. 3-14)
TK-9839

Figure 3-12 Master Clock

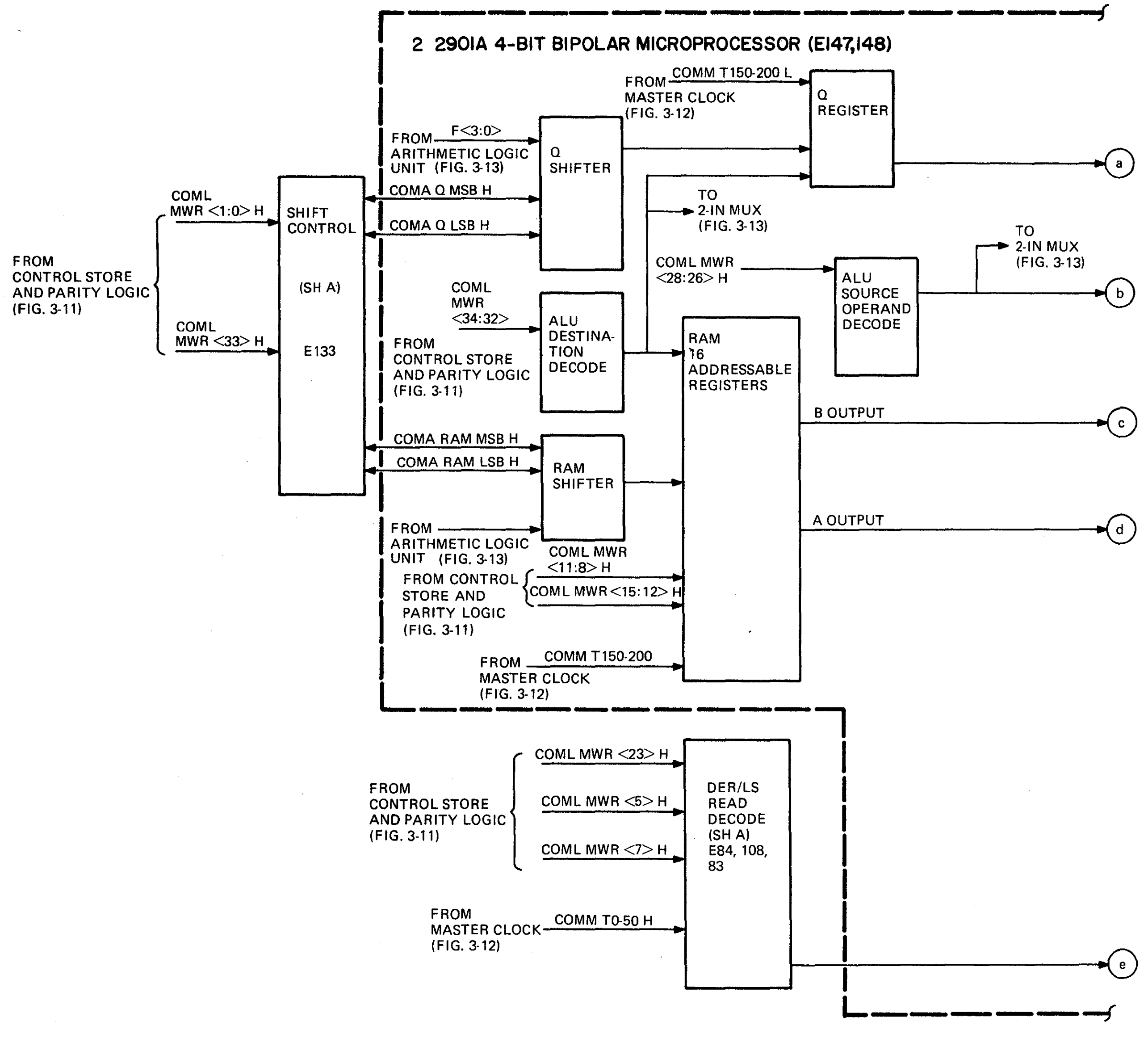
The single step circuitry consists of the preset and clear inputs of the divide-by-four flip-flops (E136,122), a single step toggle switch (S1), BR priority dip switch (E77) and pullups (R27).

To single step, the dip switch SP-10 on the BR priority switch pack (E77) is closed to ground COMC SINGLE STEP SEL L. Single-step switch S1 is toggled to alternately preset and clear the divide-by-four flip-flops. Each toggle of the single-step switch causes a new phase to be entered. Therefore, the single-step switch is toggled four times to execute one microinstruction.

3.9 ARITHMETIC AND LOGICAL UNIT

The arithmetic and logical unit circuit consists of a shift control PAL (E133), a DER/LS read decode circuit (E89,88,113), a ALU condition code register (E131), and two 2901 four-bit bipolar microprocessors. The 2901 four-bit bipolar microprocessors are cascaded to form a eight-bit data path.

Refer to Figure 3-13 for the following description of the arithmetic and logical unit.



2 2901A 4-BIT BIPOLAR MICROPROCESSOR (E147,148)
FROM MASTER CLOCK (FIG. 3-12) COMM T150-200 L
Q REGISTER
FROM ARITHMETIC LOGIC UNIT (FIG. 3-13) F<3:0>
Q SHIFTER
COMA Q MSB H
COMA Q LSB H
COML MWR <1:0> H
SHIFT CONTROL (SH A) E133
COML MWR <33> H
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
TO 2-IN MUX (FIG. 3-13)
COML MWR <28:26> H
ALU SOURCE OPERAND DECODE
TO 2-IN MUX (FIG. 3-13)
COML MWR <34:32>
ALU DESTINATION DECODE
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
RAM 16 ADDRESSABLE REGISTERS
B OUTPUT
A OUTPUT
COMA RAM MSB H
COMA RAM LSB H
RAM SHIFTER
FROM ARITHMETIC LOGIC UNIT (FIG. 3-13)
COML MWR <11:8> H
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COML MWR<15:12> H
FROM MASTER CLOCK (FIG. 3-12) COMM T150-200
COML MWR <23> H
COML MWR <5> H
COML MWR <7> H
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
DER/LS READ DECODE (SH A) E84, 108, 83
FROM MASTER CLOCK (FIG. 3-12) COMM T0-50 H
a
b
c
d
e
TK-9840

Figure 3-13 Arithmetic and Logic Unit (Sheet 1 of 2)

3.9.1 ALU Functions
The ALU is a high-speed arithmetic/logic circuit that can perform three binary arithmetic functions and five logic functions. Applying COML MWR ⟨31:29⟩ H to the ALU function decode circuit selects one of eight functions to be performed. Table 3-16 defines the ALU function selection.

3.9.2 Shift Control
The shift control (E133) controls the shift logic by writing and reading data on the shift lines to the RAM shifter circuit and the Q shifter circuit. COMA RAM MSB H and COMA RAM LSB H are applied to the RAM shifter circuit, while COMA Q MSB H and COMA RAM LSB H are applied to the Q shifter.

Applying COML MWR ⟨33⟩ H and COML MWR ⟨1:0⟩ H to the shift control (E133) determines the shift operation. COML MWR ⟨33⟩ H determines the direction of the shift or rotate. COML MWR ⟨1:0⟩ H specifies the end conditions for a single or double precision shift rotate. Table 3-17 defines the bit codes for the shift and rotates.

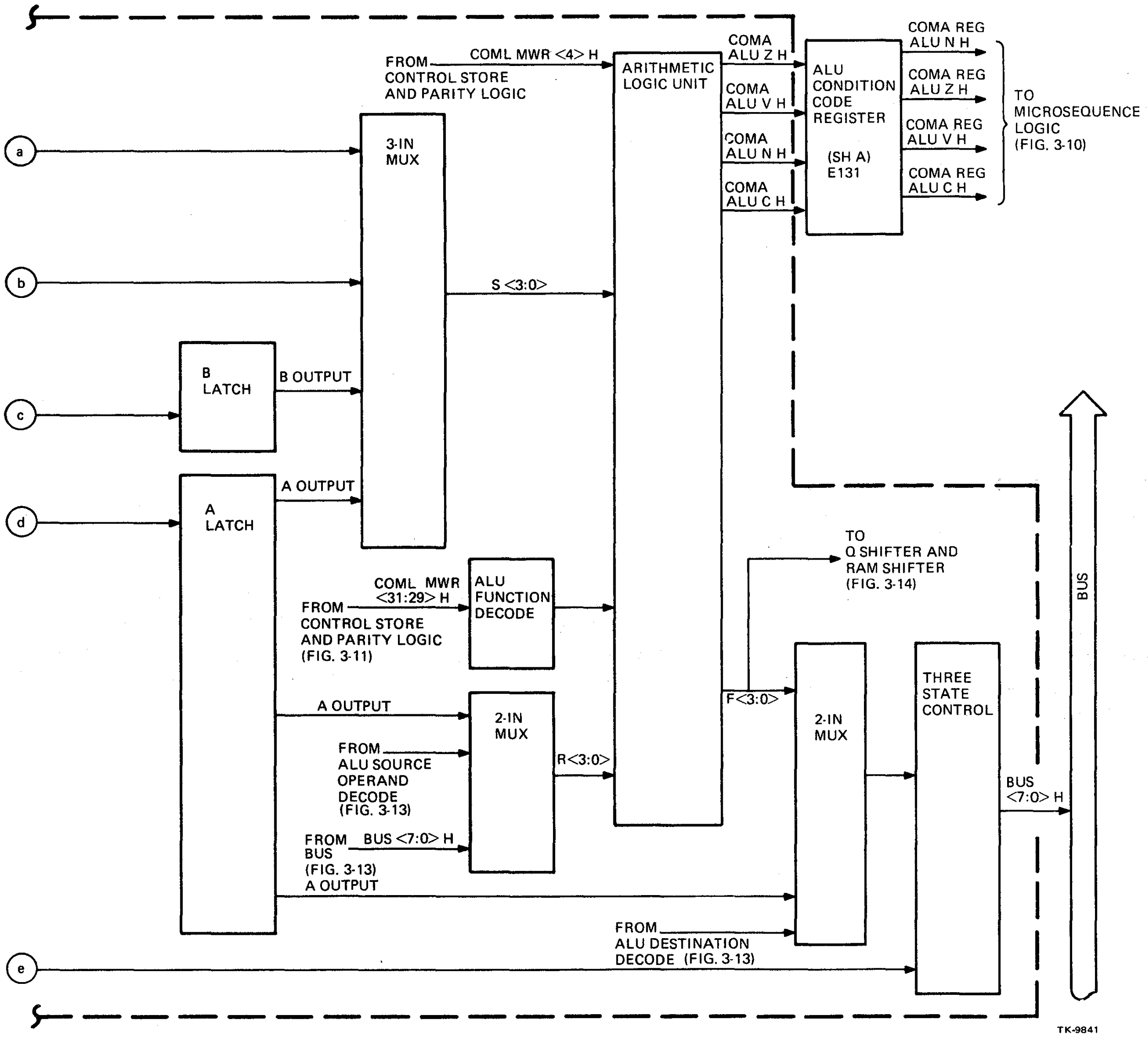


Figure 3-13 Arithmetic and Logic Unit (Sheet 2 of 2)

Table 3-16 ALU Function Selection

COML MWR ⟨31:29⟩ H 31	30	29	ALU Function
0	0	0	R plus S plus carry-in
0	0	1	S minus R minus carry-in
0	1	0	R minus S minus carry-in
0	1	1	R or S
1	0	0	R and S
1	0	1	/R and S
1	1	0	R EX-OR S
1	1	1	R EX-NOR S

Table 3-17 Shift Control (E133) Truth Table

COML MWR 33	1	0	Operation	COMA RAM MSB	COMA RAM LSB	COMA Q MSB	COMA Q LSB
0	0	0	sp shift down	0	HI Z	0	HI Z
0	0	1	dp shift down	0	HI Z	RAM LSB	HI Z
0	1	0	sp rotate down	RAM LSB	HI Z	Q LSB	HI Z
0	1	0	dp rotate down	Q LSB	HI Z	RAM LSB	HI Z
1	0	0	sp shift up	HI Z	0	HI Z	0
1	0	1	dp shift up	HI Z	Q MSB	HI Z	0
1	1	0	sp rotate up	HI Z	RAM MSB	HI Z	Q MSB
1	1	1	dp rotate up	HI Z	Q MSB	HI Z	RAM MSB

Applying COML MWR ⟨34:32⟩ H to the ALU destination decode circuit produces control signals that control the RAM shifter circuit. The RAM shifter circuit either non-shifts, shifts up one position (toward MSB), or shifts down one position (toward LSB) the input data to the RAM.

The Q shifter has two input lines: COMA Q MSB H and COMA Q LSB H. These two input lines operate similarly to the RAM shifter circuit inputs: COMA RAM MSB H and COMA RAM LSB H. When in the shift-up or shift-down mode, the Q register is shifted in a specified direction with the data input/output line of the Q register being an input line for a shift up or an output line for a shift down. In the no-shift mode, the Q input/output lines of the Q register are tri-state.

3.9.3 RAM and A/B Latches
Either COML MWR ⟨11:8⟩ H or COML MWR ⟨15:12⟩ H addressing the RAM (16 addressable registers) reads out a data file. COML MWR ⟨11:8⟩ H reads out a data file from the RAM into the A latch, while COML MWR ⟨15:12⟩ H reads out a data file from the RAM into the B latch. When both COML MWR ⟨11:8⟩ H and COML MWR ⟨15:12⟩ H are identical, the same data file is applied simultaneously at both the A and B outputs.

The clock input COMM T150–200 L controls the RAM, the A and B latches, and the Q register. Whenever COMM T150–200 L is deasserted, the data from the RAM outputs are applied to the ALU via the A and B latches. When COMM T150–200 L is asserted, both latch A and latch B latch the last data entered in the latches. Asserting COMM T150–200 L whenever COML MWR ⟨34:32⟩ H is coded to enable a file write operation, new data, as defined by the four-bit B address field, is written into the RAM. Data is clocked into the Q register on the low-to-high transition of COMM T150–200 L.

The operands from the bus (BUS ⟨7:0⟩ H) and the Q operands provide an essential function. The BUS ⟨7:0⟩ H input loads the RAM with external data (e.g. local store data). The Q register is an internal eight-bit data source that can be used for multiply/divide operations, as a data-holding register, or as an accumulator.

3.9.4 ALU Source Operands
The high-speed ALU can perform three binary arithmetic and five logic operations on the two eight-bit input words (R⟨7:0⟩ and S⟨7:0⟩). The R-input field is applied by a two-input multiplexer, whereas the S-input field is applied by a three-input multiplexer. Both the R and S multiplexers have an inhibit capability, where no data is applied to the arithmetic logic unit, which is the equivalent of a zero source operand. The A output of the RAM and the BUS ⟨7:0⟩ H are applied to the R-input multiplexer, while the S-input multiplexer has three inputs: one from the A output of the RAM, one from the B output of the RAM, and one from the Q register.

3.9.5 Source Operand Selection
Applying COML MWR ⟨28:26⟩ H to the ALU source operand decode circuit selects the source operand. Table 3-18 defines the source operand selection.

Table 3-18 Source Operand Selection

COM MWR ⟨28:26⟩ H			ALU Source Operands	
28	**27**	**26**	**R**	**S**
0	0	0	Work Register [A]	0
0	0	1	Work Register [A]	Work Register [B]
0	1	0	0	Q register
0	1	1	0	Work Register [B]
1	0	0	0	Work Register [A]
1	0	1	BUS ⟨7:0⟩ H	Work Register [A]
1	1	0	BUS ⟨7:0⟩ H	Q register
1	1	1	BUS ⟨7:0⟩ H	0

3.9.6 ALU Destination

Outputs from the ALU can be applied and stored in either the RAM or the Q register, or can be applied to the bus. Applying COML MWR ⟨34:32⟩ H to the ALU destination decode circuit enables one of eight destinations to be selected. Table 3-19 lists the destination selection codes.

The three state control circuit is enabled by the DER/LS read decode circuit output; when this control signal is deasserted, the outputs of the three state control circuit are tri-state disabled. The output of the three state control is tri-state disabled whenever there is a DER read microinstruction (i.e., COML MWR ⟨23⟩ H, COML MWR ⟨7⟩ H, and COML MWR ⟨5⟩ H are all asserted) or a LS.RD microinstruction is being performed (i.e., MWR ⟨7⟩ not asserted). Also COMM T0–50 H disables the output of the three state control circuit from time T0 to T50.

The two-input multiplexer selects either the A OUTPUT of the RAM or the ALU output; COML MWR ⟨34:32⟩ H being applied to the ALU destination decode circuit controls the destination selection.

3.9.7 Condition Codes

The condition code register (E131) is clocked with ALU condition codes at the end of the cycle, if COML MWR ⟨3⟩ H is asserted. The condition codes are defined in Table 3-20.

Table 3-19 ALU Destination Codes

COML	**MWR**	**⟨34:32⟩**	**H**	**DER/LS Read Decode**
34	**33**	**32**	**ALU Destination**	**Circuit is asserted (low)**
0	0	0	Q register = ALU	BUS ⟨7:0⟩ = ALU
0	0	1	none	BUS ⟨7:0⟩ = ALU
0	1	0	work register [B] = ALU	BUS ⟨7:0⟩ = work register [A]
0	1	1	work register [B] = ALU	BUS ⟨7:0⟩ = ALU
1	0	0	work register [B] = ALU/2*	
			Q register = Q register/2*	BUS ⟨7:0⟩ = ALU
1	0	1	work register [B] = ALU/2*	BUS ⟨7:0⟩ = ALU
1	1	0	work register [B] = ALU *2	
			Q register = Q register *2	BUS ⟨7:0⟩ = ALU
1	1	1	work register [B] = ALU *2	BUS ⟨7:0⟩ = ALU

Table 3-20 Condition Codes

Condition Code	**Definition**
COMA ALU Z H	The result of an ALU operation is zero.
COMA ALU V H	The results of an arithmetic two-complement operation has overflowed into the sign bit.
COMA ALU N H	Most significant bit (sign bit) output of the ALU.
COMA ALU C H	Carry-out of the ALU.

* Refer to Table 3-17 Shift Control (E133) Truth Table for end around conditions.

3.10 LOCAL STORE CONTROL

The local store control circuitry consists of local store (two 1K × 4 RAMs) (E134,135), decoder (E111), NOR gates (E137), multiplexers (E95,107,109), process register (E95,94,82), indirect address register (E55), AND gates (E97,98), and TX data RAMs (E37,29,50,30).

In one microcycle, local store control circuitry can read data from the local store into a working register, or write data from a working register to local store. The DMF32 microarchitecture cannot perform read modify write cycles to and from the local store.

Refer to Figure 3-14 for the following descriptions of the local store write (3.10.1), local store read (3.11.2), local store addressing (3.10.3), and special local store read instructions (3.10.4).

3.10.1 Local Store Write

COML MWR ⟨7:5⟩ H is applied to the decoder (E111). If COML MWR ⟨7:5⟩ H equals "100", a local store write microinstruction is performed. When the decoder (E111) detects a local store write to be performed, the decoder (E111) asserts COME LS WRITE L. COME LS WRITE L is applied to the NOR gates (E137), at time COMN T50–150L with COME LS WRITE L asserted, both WRITE ENABLE and CHIP ENABLE signals are asserted and applied to the local store to perform a local store write microinstruction.

3.10.2 Local Store Read

COML MWR ⟨7⟩ H is applied to the NOR gates (E137). When COML MWR ⟨7⟩ H is deasserted, the NOR gates (E137) asserts and applies CHIP ENABLE to the local store to perform a local store read microinstruction.

3.10.3 Local Store Addressing

A ten-bit address is used to address the local store. The two most significant address bits (COML MWR ⟨25:24⟩ H) are applied to the multiplexers (E96,107,109). COML MWR ⟨25:24⟩ H selects the address source for the local store. Refer to Table 3-21 for the local store address source selection.

When COML MWR ⟨25:24⟩ H equals either "00" or "01", the multiplexers (E96,107,109) select COMF LS ADR ⟨7:0⟩ H from the indirect address register (E55) for the low byte of the local store address. This address can indirectly address the first 512 locations of the local store. Refer to Figure 3-15.

When COML MWR ⟨25:24⟩ H equals "10", the multiplexers (E96,107,109) select the low byte of the local store address from two sources. The low nibble of the address low byte is COML MWR ⟨19:16⟩ H, while the high nibble is from the process register (COMF PROC ⟨3:0⟩ H). The address divides the 256 byte process space into 16 segments of 16 bytes each. The process register (COMF PROC ⟨3:0⟩ H) addresses a particular 16-byte segment, while microcode (COML MWR ⟨19:16⟩ H) directly addresses each of the 16 bytes within the segment.

If COML MWR ⟨25:24⟩ H equals "11", then the microcode (COML MWR ⟨23:16⟩ H) provides the low byte of the local store address. Microinstructions directly address this 256-byte segment of the local store.

3.10.4 Special Local Store Read Instructions

Two special local store read microinstructions are used for trap routines. When COML MWR ⟨7:5⟩ H equals either "010" or "011", local store data is clocked into one of two special discrete external registers (E37,29,50,30). If COML MWR ⟨7:5⟩ H equals "010", the decoder (E111) asserts COME LS READ CLK TX DATA LO L. COME LS READ CLK TX DATA LO L clocks local store data into the low byte of TX data RAMs (E37,29,50,30). COML MWR ⟨17:16⟩ H specifies which location in the TX data RAMs (E37,29) is to be loaded.

If COML MWR ⟨7:5⟩ H equals "011", the decoder (E111) asserts COME LS READ CLK TX DATA HI L. COME READ CLK TX DATA HI L clocks the local store data into the high byte of data RAMs. COML MWR ⟨17:16⟩ H specifies which location in the TX data RAMs (E50,30) is to be loaded. Refer to Figure 3-16.

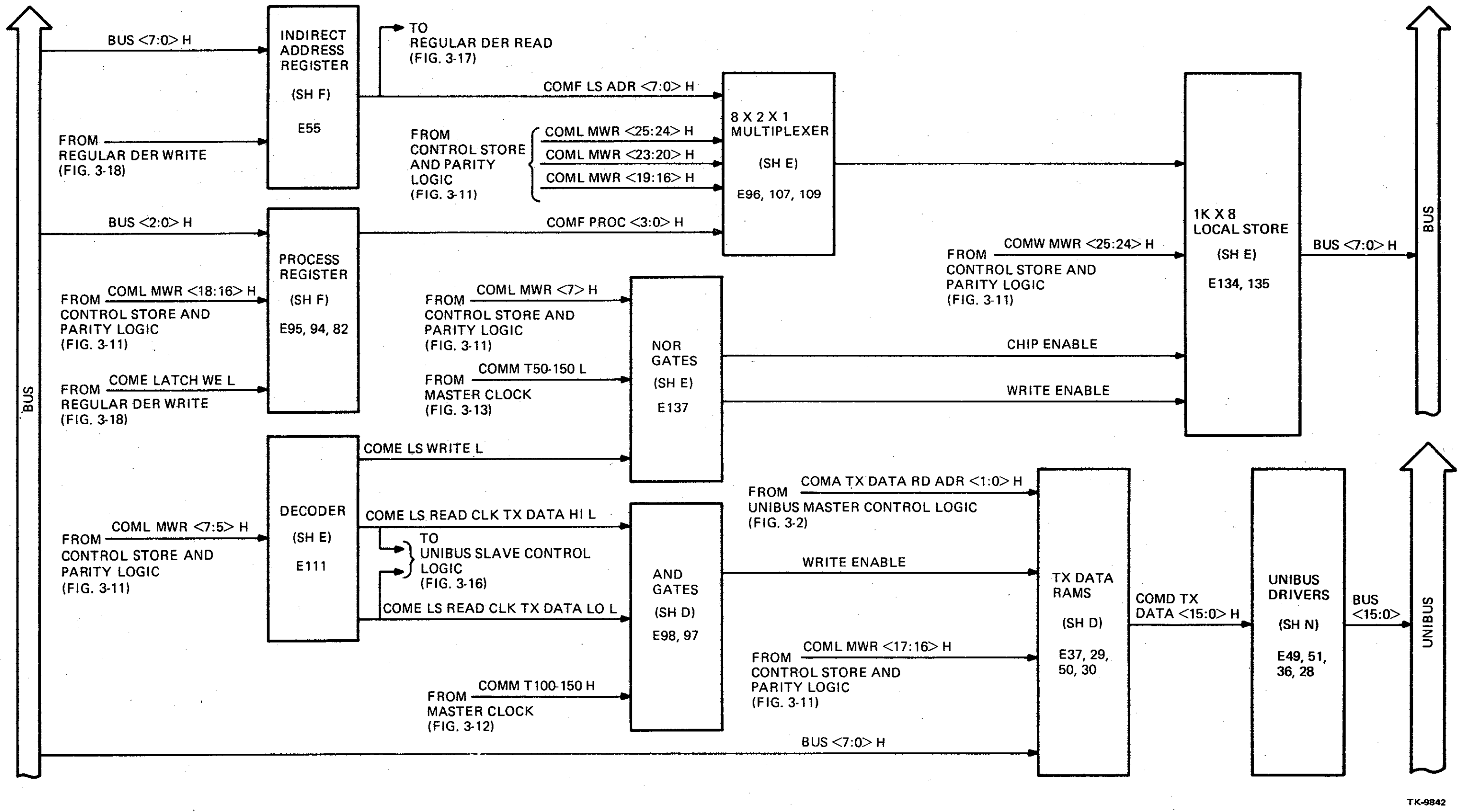

BUS
BUS <7:0> H
INDIRECT ADDRESS REGISTER
(SH F)
E55
FROM REGULAR DER WRITE (FIG. 3-18)
TO REGULAR DER READ (FIG. 3-17)
COMF LS ADR <7:0> H
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COML MWR <25:24> H
COML MWR <23:20> H
COML MWR <19:16> H
8 X 2 X 1 MULTIPLEXER
(SH E)
E96, 107, 109
BUS <2:0> H
COMF PROC <3:0> H
PROCESS REGISTER
(SH F)
E95, 94, 82
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COML MWR <18:16> H
FROM REGULAR DER WRITE (FIG. 3-18)
COME LATCH WE L
1K X 8 LOCAL STORE
(SH E)
E134, 135
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COMW MWR <25:24> H
BUS <7:0> H
BUS
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COML MWR <7> H
FROM MASTER CLOCK (FIG. 3-13)
COMM T50-150 L
NOR GATES
(SH E)
E137
CHIP ENABLE
WRITE ENABLE
DECODER
(SH E)
E111
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COML MWR <7:5> H
COME LS WRITE L
COME LS READ CLK TX DATA HI L
TO UNIBUS SLAVE CONTROL LOGIC (FIG. 3-16)
COME LS READ CLK TX DATA LO L
AND GATES
(SH D)
E98, 97
FROM MASTER CLOCK (FIG. 3-12)
COMM T100-150 H
FROM UNIBUS MASTER CONTROL LOGIC (FIG. 3-2)
COMA TX DATA RD ADR <1:0> H
WRITE ENABLE
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COML MWR <17:16> H
BUS <7:0> H
TX DATA RAMS
(SH D)
E37, 29, 50, 30
COMD TX DATA <15:0> H
UNIBUS DRIVERS
(SH N)
E49, 51, 36, 28
BUS <15:0>
UNIBUS
TK-9842

Figure 3-14 Local Store Control

Table 3-21 Local Store Address Source Selection

MWR ⟨25:24⟩	Address Source
0 0	COMF LS ⟨7:0⟩ H
0 1	COMF LS ⟨7:0⟩ H
1 0	COMF PROC ⟨3:0⟩ H COML MWR ⟨19:16⟩ H
1 1	COML MWR ⟨23:16⟩ H

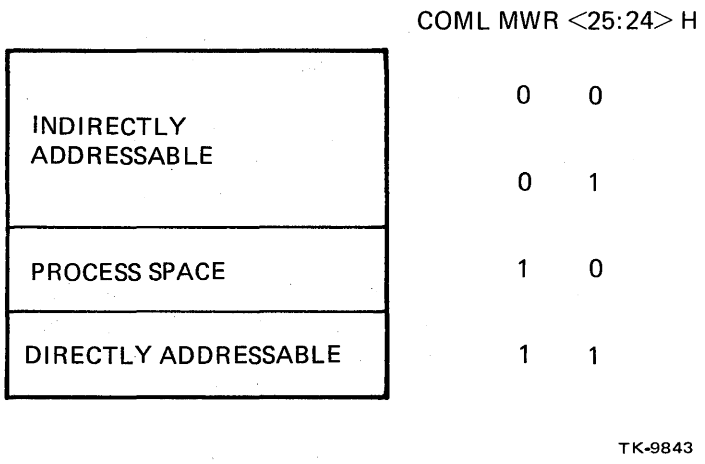


Figure 3-15 Local Store Addressing

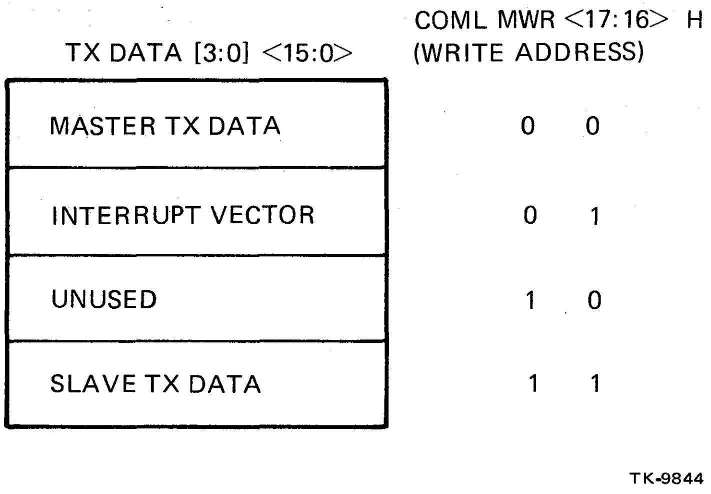


Figure 3-16 TX Data RAM Register Addressing

3.11 REGULAR DISCRETE EXTERNAL REGISTERS READ/WRITE CYCLES

Microcode executes a DER microinstruction to access a regular discrete register (DER). Each regular discrete external register is either read or write only. When COML MWR ⟨23⟩ H is deasserted, a DER write is performed; while if COML MWR ⟨23⟩ H is asserted, a DER read is performed. COML MWR ⟨22:16⟩ H specifies the address of a discrete register. Since there are more addresses than there are physical registers, most of the discrete registers respond to multiple addresses. Also, a read-only or write-only register can have the same address.

COML MWR ⟨22⟩ H determines between a regular DER cycle or a slow read or write cycle. When COML MWR ⟨22⟩ H is asserted, a regular DER cycle is performed; while if COML MWR ⟨22⟩ H is deasserted, a slow read or write cycle is performed. Refer to Section 3.12 for the description of a slow read or write cycle.

3.11.1 Regular Discrete External Register Read

Refer to Figure 3-17 for the following description of the regular DER read.

The decoder (E111) decodes COML MWR ⟨7:5⟩ H to determine if a DER microinstruction is to be performed. If COML MWR ⟨7:5⟩ H equals "101", then decoder (E111) asserts COME DISC REG R/W L which is applied to decoder (E80,84,108). With COME DISC REG R/W L, COML MWR ⟨22⟩ H (DER access), and COML MWR ⟨23⟩ H (DER read) all asserted, one of the outputs of the decoder (E80,84,108) is asserted. The address (COML MWR ⟨21:19⟩ H) selects the output of the decoder (E80,84,108), that is to be asserted. Refer to Table 3-22 for the address selection of the decoder (E80,84,108). Table 3-23 is the DER read register map.

The enable input (COMM T0-50 H) of the decoder (E80,84,108) prevents the decoder (E80,84,108) from asserting an output at time T0-50. This permits a previously enabled tri-state output to return to the high Z state before another register is enabled. The outputs of decoder (E80,84,108) are applied to the read only discrete external registers, so as to gate the contents of the register directly onto the bus when read.

The data multiplexers (E72,41,42,74), consisting of eight 4-to-1 multiplexers, are used to multiplex in both external and internal signals to the DMF32. The eight outputs from the data multiplexer (E72,41,42,74) are applied to the data multiplexer register (E71) which deskews the eight inputs. The data multiplexer register (E71) is clocked at time COMM T150-200 L, so that COML MWR ⟨18:17⟩ H sets up the data multiplexer (E72,41,42,74) one microcycle prior to reading in the data multiplexer register (E71). When reading in more than one byte from the data multiplexer (E72,41,42,74), pipelining can be used. Refer to Table 3-24.

The UNIBUS address switches multiplexer (E66,E72) and the data multiplexer register (E71) are used to read in the values of the UNIBUS address switches. This is done so that the microcode can determine the UNIBUS address of the DMF32 to do a direct memory access to itself as part of the power-up self test. The UNIBUS address switches are read in one bit at a time. COMF LS ADR ⟨2:0⟩ H selects which UNIBUS switch bit is to be read.

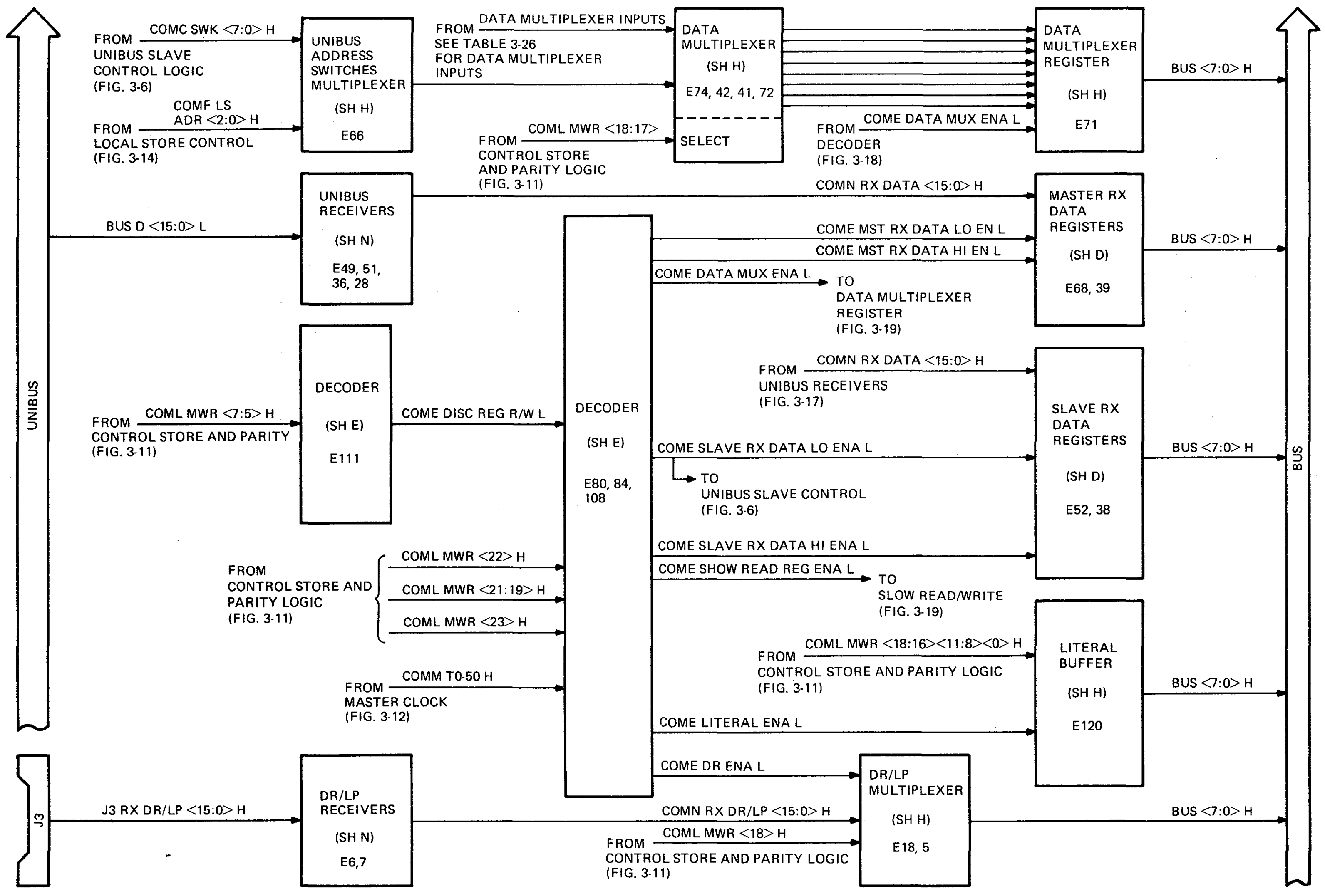

UNIBUS
FROM UNIBUS SLAVE CONTROL LOGIC (FIG. 3-6)
COMC SWK <7:0> H
FROM LOCAL STORE CONTROL (FIG. 3-14)
COMF LS ADR <2:0> H
UNIBUS ADDRESS SWITCHES MULTIPLEXER (SH H) E66
FROM SEE TABLE 3-26 FOR DATA MULTIPLEXER INPUTS
DATA MULTIPLEXER INPUTS
DATA MULTIPLEXER (SH H) E74, 42, 41, 72
SELECT
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COML MWR <18:17>
FROM DECODER (FIG. 3-18)
COME DATA MUX ENA L
DATA MULTIPLEXER REGISTER (SH H) E71
BUS <7:0> H
BUS D <15:0> L
UNIBUS RECEIVERS (SH N) E49, 51, 36, 28
COMN RX DATA <15:0> H
COME MST RX DATA LO EN L
COME MST RX DATA HI EN L
COME DATA MUX ENA L
TO DATA MULTIPLEXER REGISTER (FIG. 3-19)
MASTER RX DATA REGISTERS (SH D) E68, 39
BUS <7:0> H
FROM CONTROL STORE AND PARITY (FIG. 3-11)
COML MWR <7:5> H
DECODER (SH E) E111
COME DISC REG R/W L
DECODER (SH E) E80, 84, 108
FROM UNIBUS RECEIVERS (FIG. 3-17)
COMN RX DATA <15:0> H
COME SLAVE RX DATA LO ENA L
TO UNIBUS SLAVE CONTROL (FIG. 3-6)
SLAVE RX DATA REGISTERS (SH D) E52, 38
BUS <7:0> H
BUS
COME SLAVE RX DATA HI ENA L
COME SHOW READ REG ENA L
TO SLOW READ/WRITE (FIG. 3-19)
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COML MWR <22> H
COML MWR <21:19> H
COML MWR <23> H
FROM MASTER CLOCK (FIG. 3-12)
COMM T0-50 H
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
COML MWR <18:16><11:8><0> H
LITERAL BUFFER (SH H) E120
BUS <7:0> H
COME LITERAL ENA L
COME DR ENA L
J3
J3 RX DR/LP <15:0> H
DR/LP RECEIVERS (SH N) E6,7
COMN RX DR/LP <15:0> H
COML MWR <18> H
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
DR/LP MULTIPLEXER (SH H) E18, 5
BUS <7:0> H
TK-9845

Figure 3-17 Regular Discrete External Register Read

Table 3-22 Decoder (E80) Address Selection

Address Bits 21	20	19	Selected Output
0	0	0	COME SLAVE RX DATA HI ENA L
0	0	1	COME SLAVE RX DATA LO ENA L
0	1	0	COME MST RX DATA HI ENA L
0	1	1	COME MST RX DATA LO ENA L
1	0	0	COME DATA MUX ENA L
1	0	1	COME SLOW READ REG ENA L
1	1	0	COME DR ENA L
1	1	1	COME LITERAL ENA L

Table 3-23 DER Read Register Map

Address MWR ⟨22:16⟩ (hex)	DER Read Register
0:3F	any read of locations 0:3F initiates a slow read cycle
40:47	slave RX data register-high byte (E38)
48:4F	slave RX data register-low byte (E52)
50:57	master RX data-high byte (E39)
58:5F	master RX data-low byte (E68)
60:61	data MUX [0]
62:63	data MUX [1]
64:65	data MUX [2]
66:67	data MUX [3]
68:6F	slow read register (E70)
70:73	RX DR/LP register-high byte (E18)
74:77	RX DR/LP register-low byte (E5)
78:7F	literal register (E120)

Table 3-24 Pipelining COML MWR ⟨18:17⟩ H

Read One Multiplexer Byte		Read Two Multiplexer Bytes	
Microcycle 1	setup MWR ⟨18:17⟩	Microcycle 1	setup MWR ⟨18:17⟩
Microcycle 2	read data multiplexer register	Microcycle 2	read data multiplexer register, setup MWR ⟨18:17⟩ for next read
		Microcycle 3	read data multiplexer register

COML MWR ⟨18:17⟩ H is applied to the select pins of the data multiplexers (E72,41,42,74). COML MWR ⟨18:17⟩ H selects one of the four inputs to the multiplexer sections. Table 3-25 lists the selection configurations for the data multiplexers.

The input signals to the data multiplexers (E72,41,42,74) are defined in Table 3-26.

The tri-state DR/LP multiplexers (E18,85) multiplexers a word of receive data from the parallel interface onto the bus via the DR/LP receivers (E6,E7). COML MWR ⟨18⟩ selects between the high or low byte being applied to the bus.

A literal field from the microword is read via the literal buffer (E120) onto the bus into a working register or the Q register by executing a DER read. COME LITERAL ENA L enables the literal buffer (E120). The literal consists of COML MWR ⟨18:16⟩ H, COML MWR ⟨11:8⟩ H, and COML MWR ⟨0⟩ H.

When the DMF32 is a master device, the data from the slave device is loaded into the master RX data registers (E68,39) from the UNIBUS via the UNIBUS receivers. COME MST RX DATA LO ENA L and COME MST RX HI ENA L are applied to the master RX data registers (E68,39) from the decoder (E80). COME MST RX DATA LO ENA L and COME MST RX DATA HI ENA L determine if the low or high byte is applied to the bus respectively.

When the DMF32 is a slave device, the data from the master device is loaded into the slave RX data register (E52,38) from the UNIBUS via the UNIBUS receivers. COME SLAVE RX DATA LO ENA L and COME SLAVE RX DATA HI ENA L are applied to the slave RX data registers from the decoder (E80). COME SLAVE RX DATA LO ENA L and COME SLAVE RX DATA HI ENA L determine if the low or high byte is applied to the bus respectively.

Table 3-25 Data Multiplexer Input Selections

COML MWR ⟨18:17⟩ H		
18	**17**	**Input Pin Selections**
0	0	0
0	1	1
1	0	2
1	1	3

Table 3-26 Data Multiplexer Signal Inputs

Signal	Definition
COMC SW ⟨7:0⟩ H	UNIBUS address switch bit, which is selected by COMF LS ADR ⟨2:0⟩ H. (Pin 0)
COMJ USRT TX URN H	This is a status bit from the 2652 USRT, which indicates a transmit under run condition. (Pin 0)
COMJ USRT TX BE H	This is a status bit from the 2652 USRT, which indicates a transmit buffer empty condition. (Pin 0)
COMJ USRT TX ACT H	This is a status bit from the 2652 USRT, which indicates a transmitter active condition. (Pin 0)
COMJ USRT RX SA H	This is a status bit from the 2652 USRT, which indicates that receive status is available. (Pin 0)
COMJ USRT RX S/F H	This is a status bit from the 2652 USRT, which indicates that a sync or flag character has been received. (Pin 0)
COMJ USRT RX ACT H	This is a status bit from the 2652 USRT, which indicates a receiver active condition. (Pin 0)
COMV RX DR REQ A L	This parallel interface bit is a request line originating from the user device. (Pin 1)
COMR RX UART 0 RI L	This is the ring indicator modem control signal, which originates from the modem, and is associated with asynchronous line zero. (Pin 1)
COMN RX ADR ⟨0⟩ H	This is the UNIBUS address bit zero. Trap microcode can read this bit to determine whether a DATOB cycle is accessing the high or low byte. (Pin 1)
COMR RX UART 0 CTS L	This is the clear to send modem control signal, which originates from the modem, and is associated with asynchronous line zero. (Pin 1)
COMR RX UART 0 S.CAR L	This is the secondary carrier modem control signal, which originates from the modem, and is associated with asynchronous line zero. (Pin 1)
COMR RX UART 0 USER RX L	This is the user receive modem control signal, which originates from the modem, and is associated with asynchronous line zero. (Pin 1)
COMC INT REQ L	This is the UNIBUS interrupt request signal asserted by the interrupt control FSM (E103). (Pin 1)

Table 3-26 Data Multiplexer Signal Inputs (Cont)

Signal	Definition
COMC MST REQ L	This is the UNIBUS master request signal asserted by the master control FSM (E78). (Pin 1)
COMP RX USRT DSR L	This is the data set ready modem control signal, which originates from the modem, and is associated with the synchronous line. (Pin 2)
COMP RX USRT RI L	This is the ring indicator modem control signal, which originates from the modem, and is associated with the synchronous line. (Pin 2)
COMP RX USRT CAR L	This is the carrier detect modem control signal, which originate from the modem, and is associated with the synchronous line. (Pin 2)
COMP RX USRT CTS L	This is the clear to send modem control signal, which originates from the modem, and is associated with the synchronous line. (Pin 2)
COMC GPS 1 L	This bit is the state of the general purpose switch (E75-pin 12). (Pin 2)
COMP RX USRT USER RX L	This is the user receive modem control signal, which originates from the modem and is associated with the synchronous line. (Pin 2)
COMC GPS 1 L	This bit is the state of the general purpose switch (E75-pin 11). (Pin 2)
J1 DIST PANEL SW 1 L	This bit is the state of the dip switch S3 pin 4 on the distribution module, which indicates to the microcode whether the DR (i.e. parallel interface) or LP (i.e. line printer controller) is to be used. (Pin 2)
COMV RX DR REQ B L	This parallel interface bit is a request line originating from the user device. (Pin 3)
COMS RX UART 1 RI L	This is the ring indicator modem control signal, which originates from the modem, and is associated with the asynchronous line one. (Pin 3)
J1 DIST PANEL SW 2 L	This bit is the state of the dip switch S3 pin 5 on the distribution module, which indicates to the microcode whether the DR (i.e. parallel interface) or LP (i.e. line printer controller) is to be used. (Pin 3)
COMS RX UART 1 CTS L	This is the clear to send modem control signal, which originates from the modem, and is associated with the asynchronous line one. (Pin 3)

Table 3-26 Data Multiplexer Signal Inputs (Cont)

Signal	Definition
COMS RX UART 1 S.CAR L	This is the secondary carrier modem control signal, which originates from the modem, and is associated with the asynchronous line one. (Pin 3)
COMS RX UART 1 USER RX L	This is the user receive modem control signal, which originates from the modem, and is associated with asynchronous line one. (Pin 3)
COMC TX MSYN H	This is the UNIBUS master sync control signal originating from the master control FSM (E78). (Pin 3)
COMC TX C1 H	This is the UNIBUS C1 control signal originating from the master control FSM (E78). (Pin 3)

3.11.2 Regular Discrete External Register Write

Refer to Figure 3-18 for the following description of the regular DER write.

Decoder (E111) decodes COML MWR ⟨7:5⟩ H. If COML MWR ⟨7:5⟩ H equals "101", then the decoder (E111) asserts COME DISC REG R/W L. With COME DISC REG R/W L asserted, COML MWR ⟨22⟩ H asserted, and COML MWR ⟨23⟩ H deasserted, the decoder (E81,84,108) produces one of six clock signals. This clock signal is applied to the specific DER which is to be written to. COML MWR ⟨21:19⟩ H, that is applied to the decoder (E81,49,108), determines which one of the six clock signals is to be driven to the low state. Refer to Table 3-27 for the clock signal selection and Table 3-28 for the DER write register map. The timing signal COMM T100–150 H causes the clock signal to have a positive edge at T150. At time T150, the data is clocked into the specific DER.

COME LS ADR CLK H is applied to the clock of the indirect address register (E55). This register indirectly addresses local store (E134,135), indirectly addresses the slow read/write registers (E69,70), and addresses the multiplexer (E66).

COME TX ADR LO CLK H provides the clock for the UNIBUS address register (E67), that contains the low byte of the UNIBUS address used in a DMA transfer. COME TX ADR HI CLK H provides the clock for the UNIBUS address register (E93), that contains the high byte of the UNIBUS address used in a DMA transfer.

COME TX DR/LP LO CLK H provides the clock for the DR/LP data register (E10), that contains the low byte of the DR/LP data, while COME TX DR/LP HI CLK H provides the clock for the DR/LP data register (E9) that contains the high byte of the DR/LP data.

COME LATCH WE L is the write enable signal that is applied to the addressable latches (E95,94,82). COML MWR ⟨18:16⟩ H selects one of the eight three-bit registers. When a three-bit register is enabled by COME LATCH WE L and also is selected by COML MWR ⟨18:16⟩ H, the data (BUS ⟨2:0⟩ H) is written into the selected register. These registers contain miscellaneous data.

Table 3-29 defines the bits for the eight three-bit registers (E95,94,82).

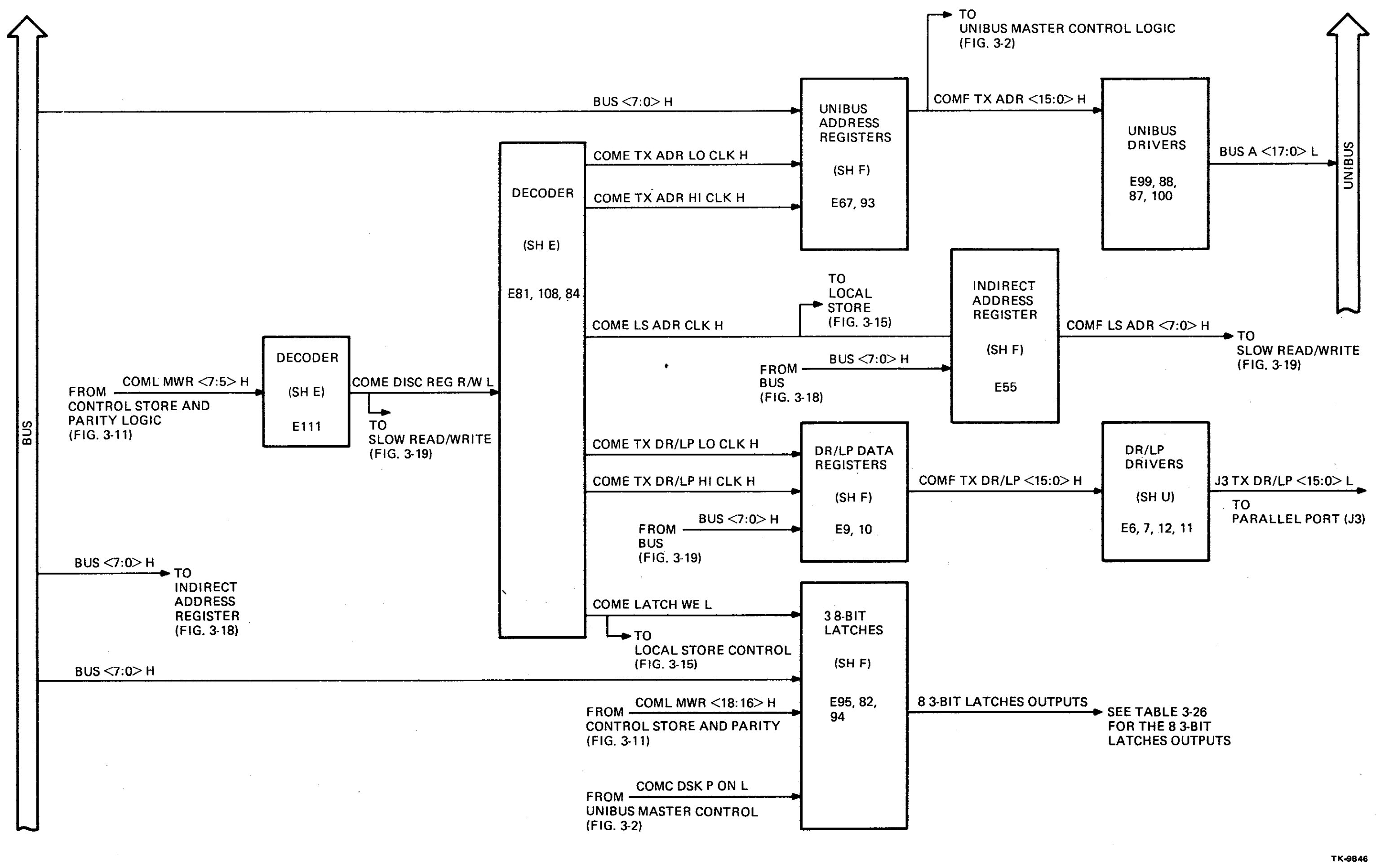

BUS
BUS <7:0> H
COML MWR <7:5> H
FROM CONTROL STORE AND PARITY LOGIC (FIG. 3-11)
DECODER (SH E) E111
COME DISC REG R/W L
TO SLOW READ/WRITE (FIG. 3-19)
DECODER (SH E) E81, 108, 84
COME TX ADR LO CLK H
COME TX ADR HI CLK H
UNIBUS ADDRESS REGISTERS (SH F) E67, 93
TO UNIBUS MASTER CONTROL LOGIC (FIG. 3-2)
COMF TX ADR <15:0> H
UNIBUS DRIVERS E99, 88, 87, 100
BUS A <17:0> L
UNIBUS
COME LS ADR CLK H
TO LOCAL STORE (FIG. 3-15)
FROM BUS (FIG. 3-18)
BUS <7:0> H
INDIRECT ADDRESS REGISTER (SH F) E55
COMF LS ADR <7:0> H
TO SLOW READ/WRITE (FIG. 3-19)
COME TX DR/LP LO CLK H
COME TX DR/LP HI CLK H
FROM BUS (FIG. 3-19)
BUS <7:0> H
DR/LP DATA REGISTERS (SH F) E9, 10
COMF TX DR/LP <15:0> H
DR/LP DRIVERS (SH U) E6, 7, 12, 11
J3 TX DR/LP <15:0> L
TO PARALLEL PORT (J3)
BUS <7:0> H
TO INDIRECT ADDRESS REGISTER (FIG. 3-18)
COME LATCH WE L
TO LOCAL STORE CONTROL (FIG. 3-15)
BUS <7:0> H
3 8-BIT LATCHES (SH F) E95, 82, 94
COML MWR <18:16> H
FROM CONTROL STORE AND PARITY (FIG. 3-11)
COMC DSK P ON L
FROM UNIBUS MASTER CONTROL (FIG. 3-2)
8 3-BIT LATCHES OUTPUTS
SEE TABLE 3-26 FOR THE 8 3-BIT LATCHES OUTPUTS
TK-9846

Figure 3-18 Regular Discrete External Register Write

Table 3-27 Decoder (E81,84,108) Clock Selection

Address Bits COML MWR ⟨21:19⟩ H 21	20	19	Selected Clock Signal
0	0	0	COME LS ADR CLK H
0	0	1	COME TX ADR LO CLK H
0	1	0	COME TX ADR HI CLK H
0	1	1	not used
1	0	0	COME LATCH WE L
1	0	1	not used
1	1	0	COME TX DR/LP LO CLK H
1	1	1	COME TX DR/LP HI CLK H

Table 3-28 DER Write Register Map

Address Bits COML MWR ⟨22:16⟩ H (hex)	DER Write Register
0:3F	any write to locations 0:3F initiates a slow write cycle
40:47	indirect address register (E55)
48:4F	transmit address register-low byte(E67)
50:57	transmit address register-high byte (E93)
58:5F	no registers
60	latch 0
61	latch 1
62	latch 2
63	latch 3
64	latch 4
65	latch 5
66	latch 6
67	latch 7
68:6F	no registers
70:77	transmit DR/LP register-low byte (E10)
78:7F	transmit DR/LP register-high byte (E9)

Table 3-29 Eight 3-Bit Registers (E95,94,82) Bits

Bits	Definition
COMF PROC ⟨3:0⟩ H	These four bits are the process register.
COMF USRT TX CLK SOU H	This bit controls the source of the transmit clock for the USRT (E40), andthe source of the transmit clock applied to the modem. When this bit is clear, the transmit clock for the USRT (E40) is the one originating from the modem, while the transmit clock applied to the modem is held marking. If this bit is set, the transmit clock for the USRT (E40) originates from the baud rate generator (E32), and the transmit clock applied to the modem also originates from the BRG.
COMF FATAL FAULT H	This diagnostic bit being set forces a control store parity error.
CONF TX ADR ⟨17:16⟩ H	These two bits are the two most significant bits of the UNIBUS address register.
COMF CI H	This is the UNIBUS CI control bit that is applied to master control FSM (E78).
COMF TX CO H	This is the UNIBUS CO control bit.
COMF USRT RX ENA H	This is the receive enable control bit for the USRT (E40).
COMF USRT TX ENA H	This is the transmit enable control bit for the USRT (E40).
COMF USRT MAINT H	This is maintenance control bit for the USRT.
COMF TX DR DATA XMTD L	This is a parallel interface control bit that is applied to the user device.
COMF TX DR CTRL ONE H	This is a parallel interface control bit that is applied to the user device.
COMF TX DR CTRL ZERO H	This is a parallel interface control bit that is applied to the user device.
COMF INHIBIT PUP H	The setting of this bit inhibits any pushing or popping of the micr-osequencer stack.
COMF TX DR N.D.R. HI L	This is a parallel interface control bit that is applied to the user device.
COMF TX DR N.D.R. LO L	This is a parallel interface control bit that is applied to the user device.

Table 3-29 Eight 3-Bit Registers (E95,94,82) Bits (Cont)

Bits	Definition
COMF USRT SINGLE STEP CLK H	The setting of this bit causes the USRT receive clock to be inverted. COMF USRT SINGLE STEP CLK H is exclusive "or"ed with COMP RX USRT RX CLK H (USRT receive clock originating from the modem) to produce COMB USRT RX CLK H (the receive clock that is applied to the USRT).
COMF NPR START H	The zero to one transition of this bit initiates a UNIBUS NPR cycle. COMF NPRSTART H is applied to the master control FSM (E78).
COMF BR START H	The zero to one transition of this bit initiates a UNIBUS interrupt cycle. COMF BRSTART H is applied to the interrupt control FSM (E103).

3.12 SLOW READ/WRITE CYCLES

The UARTs, USRT, and BRGs have read and write cycle times that are four microinstructions long. Rather than lengthening the microcycle, the slow read/write circuitry enables slow access to occur concurrent with other microinstructions.

A slow read cycle is initiated by executing a DER read microinstruction with COML MWR ⟨22⟩ H deasserted. The address (COMF LS ADR ⟨4:0⟩ H) for the desired slow device is loaded into the indirect adddress register (E55) prior to initiating the read. Four microcycles after the read cycle is initiated, the requested data is automatically loaded into the slow read register (E70). While the multicycle slow read is in progress, other microinstructions can be executed. The slow read microinstruction only initiates a slow read cycle, and does not read in any data. After waiting three microinstruction states, the microcode reads in the requested data by performing a regular DER read of the slow read register (E70).

A slow write cycle is initiated by executing a DER write microinstruction with COML MWR ⟨22⟩ H deasserted. During this microinstruction, the data from the bus is automatically clocked into the slow write register (E69). Prior to initiating the slow write cycle, the address (COMF LS ADR ⟨4:0⟩ H) is loaded into the indirect address register (E55). While the multicycle slow write is occurring, other microinstructions can be executed.

3.13 SLOW READ/WRITE CONTROL LOGIC

Refer to Figure 3-19 for the following description of the slow read and slow write cycles. Table 3-30 shows the required timing between successive slow reads and writes.

The slow read/write control FSM (E73) controls the slow read and slow write cycles. COMF LS ADR ⟨5:4⟩ H, COML MWR ⟨22⟩ H, and COME DISC REG R/W L are all applied to the slow read/write control FSM. COMF LS ADR ⟨5:4⟩ H selects either the USRT, one of the UARTs or one of the BRGs to be accessed. Refer to Table 3-31 for the slow device selection. COML MWR ⟨22⟩ H being deasserted specifies a slow DER cycle, if a DER cycle is to be performed. COME DISC REG R/W L being asserted indicates that a DER microinstruction is being performed.

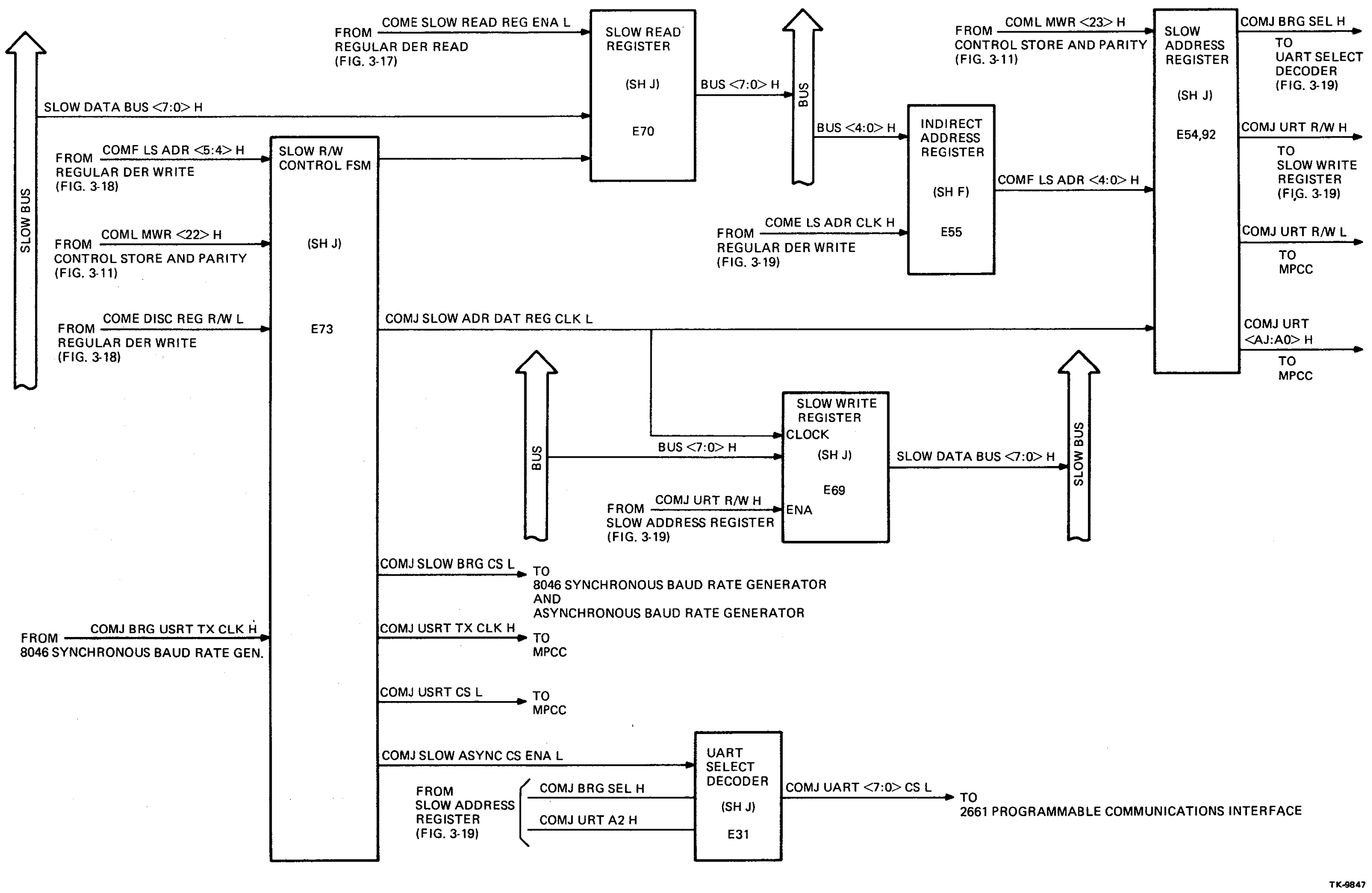

SLOW BUS
SLOW DATA BUS <7:0> H
FROM REGULAR DER READ (FIG. 3-17)
COME SLOW READ REG ENA L
SLOW READ REGISTER (SH J) E70
BUS <7:0> H
BUS
BUS <4:0> H
INDIRECT ADDRESS REGISTER (SH F) E55
COME LS ADR CLK H
FROM REGULAR DER WRITE (FIG. 3-19)
COMF LS ADR <4:0> H
FROM CONTROL STORE AND PARITY (FIG. 3-11)
COML MWR <23> H
SLOW ADDRESS REGISTER (SH J) E54,92
COMJ BRG SEL H
TO UART SELECT DECODER (FIG. 3-19)
COMJ URT R/W H
TO SLOW WRITE REGISTER (FIG. 3-19)
COMJ URT R/W L
TO MPCC
COMJ URT <AJ:A0> H
TO MPCC
FROM REGULAR DER WRITE (FIG. 3-18)
COMF LS ADR <5:4> H
SLOW R/W CONTROL FSM (SH J) E73
FROM CONTROL STORE AND PARITY (FIG. 3-11)
COML MWR <22> H
FROM REGULAR DER WRITE (FIG. 3-18)
COME DISC REG R/W L
COMJ SLOW ADR DAT REG CLK L
SLOW WRITE REGISTER
CLOCK
(SH J)
E69
ENA
BUS <7:0> H
FROM SLOW ADDRESS REGISTER (FIG. 3-19)
COMJ URT R/W H
SLOW DATA BUS <7:0> H
COMJ SLOW BRG CS L
TO 8046 SYNCHRONOUS BAUD RATE GENERATOR AND ASYNCHRONOUS BAUD RATE GENERATOR
FROM 8046 SYNCHRONOUS BAUD RATE GEN.
COMJ BRG USRT TX CLK H
COMJ USRT TX CLK H
TO MPCC
COMJ USRT CS L
TO MPCC
COMJ SLOW ASYNC CS ENA L
UART SELECT DECODER (SH J) E31
FROM SLOW ADDRESS REGISTER (FIG. 3-19)
COMJ BRG SEL H
COMJ URT A2 H
COMJ UART <7:0> CS L
TO 2661 PROGRAMMABLE COMMUNICATIONS INTERFACE
TK-9847

Figure 3-19 Slow Read/Write Logic

Table 3-30 Required Timing Between Successive Slow Reads and Writes

Microinstruction				
	Case A	**Case B**	**Case C**	**Case D**
1.	SLOW READ INIT	SLOW READ INIT	SLOW WRITE INIT	SLOW WRITE INIT
2.	ANY MICROINSTRUCTION	ANY MICROINSTRUCTION	ANY MICROINSTRUCTION	ANY MICROINSTRUCTION
3.	ANY MICROINSTRUCTION	ANY MICROINSTRUCTION	ANY MICROINSTRUCTION	ANY MICROINSTRUCTION
4.	ANY MICROINSTRUCTION	ANY MICROINSTRUCTION	ANY MICROINSTRUCTION	ANY MICROINSTRUCTION
5.	READ THE SLOW READ REGISTER	READ THE SLOW READ REGISTER	ANY MICROINSTRUCTION	ANY MICROINSTRUCTION
6.	ANY MICROINSTRUCTION	SLOW READ INIT	SLOW WRITE INIT	SLOW READ INIT
7.	SLOW WRITE INIT	ANY MICROINSTRUCTION	ANY MICROINSTRUCTION	ANY MICROINSTRUCTION

Case A: Slow Read then Slow Write
Case B: Slow Read then Slow Read
Case C: Slow Write then Slow Write
Case D: Slow Write then Slow Read

Table 3-31 USRT, UARTs, and BRGs Selection

Indirect Address Register Bits						Selected Device
⟨5⟩		⟨4⟩		⟨3⟩		⟨2⟩ ⟨1⟩ ⟨0⟩
0	0	0	0	X	X	UART0[3:0]⟨7:0⟩
0	0	0	1	X	X	UART1[3:0]⟨7:0⟩
0	0	1	0	X	X	UART2[3:0]⟨7:0⟩
0	0	1	1	X	X	UART3[3:0]⟨7:0⟩
0	1	0	0	X	X	UART4[3:0]⟨7:0⟩
0	1	0	1	X	X	UART5[3:0]⟨7:0⟩
0	1	1	0	X	X	UART6[3:0]⟨7:0⟩
0	1	1	1	X	X	UART7[3:0]⟨7:0⟩
1	0	0	X	X	X	USRT[7:0]⟨7:0⟩
1	1	0	X	X	X	ASYNC BRG⟨7:0⟩
1	1	1	X	X	X	SYNC BRG⟨4:0⟩

MWR⟨22⟩:	= 0 for slow access 1 for regular DER access
MWR⟨23⟩:	= 0 for a write 1 for a read

Refer to Table 3-32 for the slow read/write control FSM (E73) truth table.

The outputs pertaining to the slow read/write cycles of the slow read/write control FSM (E73) perform the following operations.

COMJ SLOW READ REG CLK H is applied to the clock of the slow read register (E70). COMJ SLOW READ REG CLK H clocks the requested data (SLOW BUS ⟨7:0⟩ H) into the slow read register (E70) near the end of a slow read cycle.

COMJ USRT CS L is the select signal for the USRT (E40).

COMJ SLOW ADR DATA REG CLK L is applied to both the slow address register (E54) and the slow write register (E69). COMJ SLOW ADR DATA REG CLK L clocks both COMF LS ADR ⟨4:0⟩ H and COML MWR ⟨23⟩ H into the slow address register (E54). The outputs of the slow address register (E54) select either the USRT, one of the UARTs or one of the BRGs that is to be accessed.

COMJ SLOW BRG CS L is applied with COMJ BRG SEL H via NOR gates E79-10 and E79-4 to both the asynchronous and the synchronous baud rate generators. COMJ SLOW BRG CS L is the strobe signal for the write only baud rate generators.

COMJ SLOW ASYN CS ENA L is applied to decoder (E31). Decoder E31 selects one of the eight UARTs.

Table 3-32 Slow Read/Write PAL Truth Table

MWR ⟨22⟩	IAR ⟨5⟩	IAR ⟨4⟩	DER	PON	STATE	SLO RD REG CLK	ADR DAT REG CLK	S CS	A CS	B CS	SO	STATE	SLO RD REG CLK	ADR DAT REG CLK	S CS	A CS	B CS	SO
X	X	X	X	0	X	X	X	X	X	X	X	0	1	0	1	1	1	1
0	X	X	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1
1	X	X	X	1	0	1	0	1	1	1	1	0	1	0	1	1	1	1
X	X	X	1	1	0	1	0	1	1	1	1	0	1	0	1	1	1	1
X	0	X	X	1	1	1	1	1	1	1	1	2	1	1	1	0	1	1
X	X	X	X	1	2	1	1	1	0	1	1	3	0	1	1	0	1	1
X	X	X	X	1	3	0	1	1	0	1	1	4	1	1	1	0	1	0
X	X	X	X	1	4	1	1	1	0	1	0	0	1	0	1	1	1	1
X	1	0	X	1	1	1	1	1	1	1	1	5	1	1	0	1	1	1
X	X	X	X	1	5	1	1	0	1	1	1	6	0	1	0	1	1	1
X	X	X	X	1	6	0	1	0	1	1	1	7	1	1	0	1	1	0
X	X	X	X	1	7	1	1	0	1	1	0	0	1	0	1	1	1	1
X	1	1	X	1	1	1	1	1	1	1	1	8	1	1	1	1	0	1
X	X	X	X	1	8	1	1	1	1	0	1	9	0	1	1	1	0	1
X	X	X	X	1	9	0	1	1	1	0	1	10	1	1	1	1	0	0
X	X	X	X	1	10	1	1	1	1	0	0	0	1	0	1	1	1	1

COMF LS ADR ⟨5:4⟩ H determines which of the following is to be accessed:

- One of the eight UARTs (COMF LS ADR ⟨5⟩ H = 0)
- The USRT (COMF LS ADR ⟨5:4⟩ H = 10)
- One of the BRGs (COMF LS ADR ⟨5:4⟩ H = 11).

If an UART is to be accessed, then COMJ SLOW ASYNC CS ENA L is asserted. COMJ SLOW ASYNC CS ENA L is applied to the UART select decoder (E31), which causes the select pin of the UART selected by COMF LS ADR ⟨4:2⟩ H to be enabled.

If the USRT is to be accessed, then COMJ USRT CS L is asserted. COMJ USRT CS L enables the USRT.

If one of the BRGs is to be accessed, then COMJ SLOW BRG CS L is asserted.

3.14 SLOW READ AND WRITE STATES

Refer to Figure 3-20 for the following description of the slow read and write states.

Looping in state 0, the slow read and write logic waits for a slow cycle to be initiated. When COML MWR ⟨22⟩ H is deasserted and COME DISC REG R/W L is asserted, a slow read or a slow write cycle is initiated. With a slow read or a slow write initiated, the slow read and write control FSM (E73) asserts COMJ SLOW ADR DAT REG CLK L and enters state 1.

Asserting COMJ SLOW ADR DAT REG CLK L causes the slow address register (E54) and slow write register (E69) to be clocked. The address (COMF LS ADR ⟨4:0⟩ H) for a slow device and COML MWR ⟨23⟩ H are loaded into the slow address register (E54). COML MWR ⟨23⟩ H produces COMJ URT R/W H. COMJ URT R/W H determines whether a slow read or a slow write is to be performed. COMJ URT R/W L deasserted is the write enable signal for the USRT and the eight UARTs. Also, when COMJ URT R/W H is deasserted, COMJ URT R/W H enables the slow write register (E69) to apply data to the slow data bus.

COMF LS ADR ⟨5:4⟩ H determines which of the following is to be accessed:

- One of the eight UARTs (COMF LS ADR ⟨5⟩ H = 0)
- The USRT (COMF LS ADR ⟨5:4⟩ H = 10)
- One of the BRGs (COMF ADR ⟨5:4⟩ H = 11).

If a UART is to be accessed, then COMJ SLOW ASYNC CS ENA L is asserted and state 2 is entered. COMJ SLOW ASYN CS ENA L is applied to the UART select decoder (E31), which causes select pin of the UART selected by COMF LS ADR ⟨4:2⟩ H to be enabled.

If the USRT is to be accessed, then COMJ USRT CS L is asserted, and state 5 is entered. COMJ USRT CS L enables the USRT.

If one of the BRGs is to be accessed, then COMJ SLOW BRG CS L is asserted and state 8 is entered. COMJ SLOW BRG CS L enables either the asynchronous BRG or the synchronous BRG, depending on whether COMJ BRG SEL H is deasserted or asserted respectively.

From state 2, 8, or 5, wait state 3, 9, or 6 is entered respectively. During a read, data is accessed; during a write, data that is to be written is set up. At this time COMJ SLOW READ REG CLK H is deasserted to be asserted during the next state.

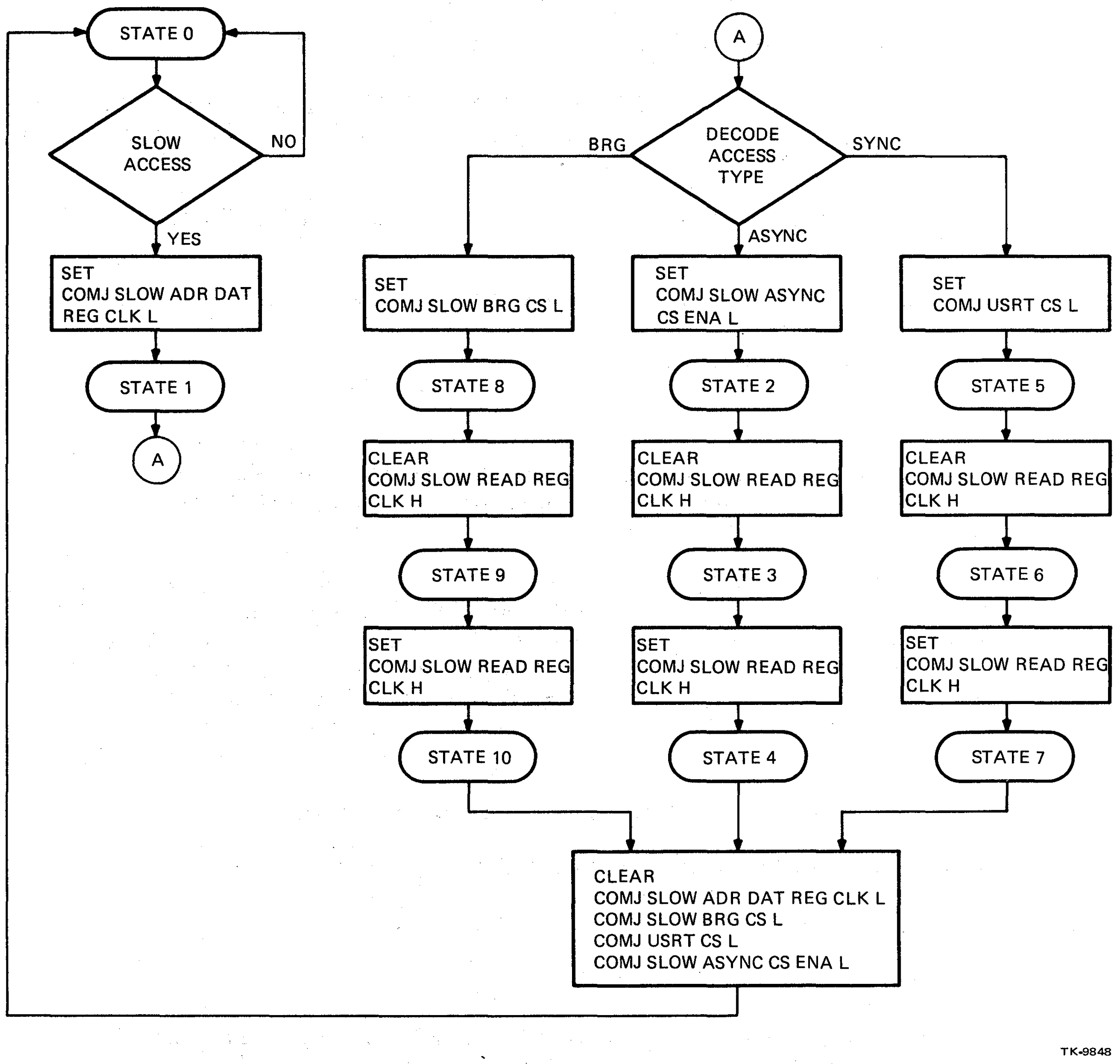

STATE 0
SLOW ACCESS
NO
YES
SET COMJ SLOW ADR DAT REG CLK L
STATE 1
A
A
DECODE ACCESS TYPE
BRG
ASYNC
SYNC
SET COMJ SLOW BRG CS L
SET COMJ SLOW ASYNC CS ENA L
SET COMJ USRT CS L
STATE 8
STATE 2
STATE 5
CLEAR COMJ SLOW READ REG CLK H
CLEAR COMJ SLOW READ REG CLK H
CLEAR COMJ SLOW READ REG CLK H
STATE 9
STATE 3
STATE 6
SET COMJ SLOW READ REG CLK H
SET COMJ SLOW READ REG CLK H
SET COMJ SLOW READ REG CLK H
STATE 10
STATE 4
STATE 7
CLEAR COMJ SLOW ADR DAT REG CLK L COMJ SLOW BRG CS L COMJ USRT CS L COMJ SLOW ASYNC CS ENA L
TK-9848

Figure 3-20 Slow Read/Write FSM Flow

From state 3, 9, or 6, state 4, 10, or 7 are entered respectively. COMJ SLOW READ REG CLK H is asserted to clock the data on the slow bus into the slow read register (E70). During a slow read cycle, the data from the slow bus is the data read from the USRT on one of the UARTs. During a slow write cycle, the data is the data that is in the slow write register (E69); thus for a slow write cycle, the slow read register (E70) is clocked with the data that is initially loaded into the slow write register (E69). This can be used for diagnostics to form a loopback through the slow bus.

From state 4, 10, or 7, state 0 is entered. In state 0, COMJ SLOW ADR DAT REG CLK L and any of the following: COMJ SLOW ASYNC CS ENA L, COMJ SLOW BRG CS L, and COMJ USRT CS L are deasserted. This completes the slow cycle.

3.15 EIA/CCITT DRIVERS AND RECEIVERS

The EIA/CCITT drivers are 9636 dual single-ended line drivers. The 9636 chips are compatible with both RS423 and RS232-C electrical specifications. All synchronous data, asynchronous data, clock, control signals originating from the DMF32 use 9636 drivers. Each 9636 driver is an eight pin mini-dip with two drivers.

The waveshape control pin (pin 1) of the mini-dip package is connected to ground via a resistor. The value of the resistor determines the slew rate of the signals originating from the two driver outputs. These slew rate resistors are R18 (1K ohm) and R20 (1M). The slew rate of all the data and clock signals are controlled by R20, while the slew rates for the other modem control signals are controlled by R18.

The EIA/CCITT receivers consist of 9637 AS chips. The 9637 AS chips are eight pin mini-dips with two receivers that are compatible with both RS232-C and RS423 electrical specifications. All synchronous and asynchronous data, timing, and control signals originating from the modem that are received by the DMF32 use 9637 as chips.

DMF32 fail safes all of the modem receive signals to the OFF state. The synchronous line receivers (E33,34,43,44) have jumpers (W3,4,5,6) connecting the positive inputs of these receivers to ground. The other synchronous line drivers and receivers have jumperable connections. Removing these jumpers enable the following signals J2 USRT RX D H, J2 USRT DSR RTN L, J2 USRT DCE TX CLK RTN L, and J2 USRT DCE RX CLK RTN L to receive differentially.

3.16 DR/LP DRIVERS AND RECEIVERS

The line printer uses a subset of the drivers and receivers of the parallel interface. Therefore, both a parallel interface and a line printer cannot be used simultaneously.

3.16.1 DR/LP TTL Drivers

The TTL drivers for 16 of the signals (J3 TX DR/LP ⟨15:0⟩ L), use two 74S240 inverting buffers (E11,12). Four additional signals (J3 TX DR N.D.R. ⟨LO:HI⟩ H, J3 TX DR N.D.R.H, and J3 TX DR DATA XMID H) are driven with a 7437 buffer (E85). One additional signal J2 TX DR INIT L is driven by a 7437 (E122). Two signals, J2 TX DR CTRL ONE L and J2 TX DR CTRL ZERO L use 74LS240 drivers.

3.16.2 DR/LP TTL Receivers

The TTL receivers for 16 of the signals (J3 RX DR/LP ⟨15:0⟩ L) use two 74LS240 inverting buffers (E6,7). Two additional signals (J3 RX DR REQ ⟨A:B⟩ H) are received by another 74LS240 inverting buffer (E8).

DMF32 MULTI-FUNCTION COMMUNICATION INTERFACE TECHNICAL DESCRIPTION

Reader's Comments

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc? Is it easy to use? __________

What features are most useful? __________

What faults or errors have you found in the manual? __________

Does this manual satisfy the need you think it was intended to satisfy? __________

Does it satisfy *your* needs? __________ Why? __________

Please send me the current copy of the *Documentation Products Directory,* which contains information on the remainder of DIGITAL's technical documentation.

Name	Street
Title	City
Company	State/Country
Department	Zip

Additional copies of this document are available from:

Digital Equipment Corporation
Accessories and Supplies Group
P.O. Box CS2008
Nashua, New Hampshire 03061

Attention: Documentation Products
Telephone: 1-800-258-1710

Order No. EK-DMF32-TD

VW

Fold Here

Do Not Tear – Fold Here and Staple

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD, MA.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Educational Services/Quality Assurance
12 Crosby Drive, BU/E08
Bedford, MA 01730

Digital Equipment Corporation • Bedford, MA 01730