EK-MS750-TD-001

VAX-11/750 Memory System
Technical Description

digital equipment corporation • maynard, massachusetts

First Edition, September 1980

Printed in U.S.A.

This document was set on DIGITAL's DECset-8000 computerized typesetting system.

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

FIGURES

TABLES

CHAPTER 1
INTRODUCTION

1.1 THE MANUAL'S PURPOSE AND FORMAT

This manual describes the VAX-11/750 memory controller (L0011), explaining its interface and detailing its operations. In addition, the appendices describe the functions of the memory controller's gate arrays as well as the functions of the MS750 memory array.

1.2 RELATED DOCUMENTS

Table 1-1 is a list of related hardware documents.

Manuals in hard copy form may be obtained from:

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

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

For information concerning microfiche libraries, contact:

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

1.3 PHYSICAL DESCRIPTION

The VAX-11/750's memory controller is an extended hex-multilayer board. (See Figure 1-1.) The L0011 memory controller has no adjustments, but there are two LED indicators on the module: a fault indicator and a power indicator. The red fault LED indicator lights when an illegal memory configuration is detected (one or more memory array board(s) is/are plugged into noncontiguous memory array slots). The green status LED indicator lights when the memory controller is powered by the main power supply or by the optional battery backup.

1.4 FUNCTIONAL DESCRIPTION

CMI devices, such as CPUs, communicate with the L0011 memory controller by way of the CMI bus. Also, the memory controller communicates to as many as eight MS750 MOS memory arrays via the internal bus. Operating with the 6.7 MHz CMI bus clock, the memory controller provides the timing and control signals for the three types of memory cycles: read, long-word write and byte write.

Table 1-1 Related Hardware Documents

Title	Document Number	Form
Central Processor Unit (CPU) Technical Description	EK-KA750-TD	Microfiche
Unibus Interface (UBI) Technical Description	EK-DW750-TD	Microfiche
Power System Technical Description	EK-PS750-TD	Microfiche
Massbus Adapter (MBA) Technical Description	EK-RH750-TD	Microfiche
Installation and Acceptance Test Manual	EK-SI750-IN-001	Hard Copy
Floating-Point Adapter (FPA) Technical Description	EK-RH750-TD	Microfiche

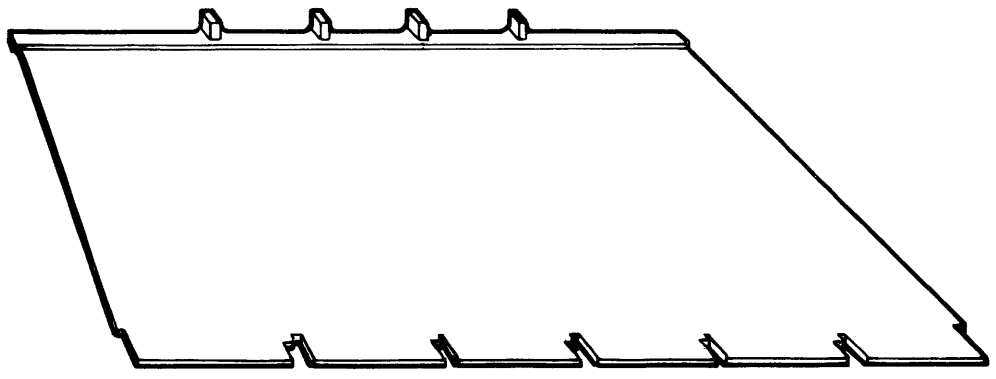

Figure 1-1 Memory Controller Module

Other CMI bus devices communicate with the memory controller using the following signal lines: 32 data/address lines, 2 status lines (which provide no error, single-bit error or multiple error status information), a data bus busy signal (which holds the bus until data transactions are complete), and a 6.7 MHz clock.

1.4.1 MOS Memory Refresh

Every 14 μs the memory controller refreshes a single row of addresses of the dynamic memory so that each of the 128 memory row addresses is refreshed every 2 ms. All MOS RAMs are refreshed simultaneously on all eight memory arrays.

A refresh oscillator, which is synchronized with the bus clock, determines when the refresh operation is to be performed. The address of the row to be refreshed is contained in a refresh address counter that is incremented at the conclusion of each refresh cycle. If the system clock is not present, refresh operations are performed under control of the memory controller's internal clock.

1.4.2 Power-On Initialization

A power-on initialization circuit in the memory controller writes all 0s and the proper error correction code (ECC) throughout the VAX-11/750's memory so that after a power-on sequence is completed, the memory responds to any cycle type, without irrelevant ECC error responses.

The initialization sequence is started when the system's ac power is applied. For memory systems that have an optional battery backup unit, the initialization sequence is not started if the ac power returns within the battery backup's support time.

The initialization process performs 16,384 long-word write cycles at the refresh rate of once every 14 μs. The memory system asserts the AC LO line until two initialization processes are complete (which takes approximately 500 ms).

1.4.3 Error Checking and Correction Logic (ECC)

The memory controller contains error correction logic that detects and corrects single-bit errors in the 32-bit data field. Double-bit errors are detected but not corrected by the ECC logic.

Seven check bits, which are generated from a 32-bit data field, are stored with that data field in its address location. These check bits are produced by generating the parity on selected groups of bits in the data field.

During a read cycle, the data field and its seven check bits are read from memory. The data field and the check bits are then applied to the two memory error correction (MEC) gate arrays, which generate syndrome bits. If the syndrome bits indicate an error, they are decoded to identify the incorrect data bit. Also, the error syndrome and the address of the page containing the error are stored in control/status register 0 (CSR0). Therefore, CSR0 contains information identifying whether the error is correctable or not, and if more than one error has occurred.

1.4.4 Memory System Diagnostic Modes

Three memory system diagnostic modes affect the ECC logic: ECC disable mode, page mode, and diagnostic check mode.

The ECC disable mode is used in conjunction with page mode to disable the ECC logic for a 512-byte page or for the entire memory. The ECC disable mode does not correct single-bit errors or log error information in control/status register 0 (CSR0). The seven check bits read from memory are stored in CSR1 <06:00>. As during a write operation in this mode, the generated check bits are also stored in CSR1 <06:00>. However, correct check bits (in any mode) are always written into memory.

The diagnostic check mode applies only to a single page. During a read operation in this mode, the contents of CSR1 <06:00> are substituted for the check bits read from memory.

1.4.5 Bootstrap ROMs

Four ROMs (512 words × 4 bits) on the memory controller are used for the bootstrap functions. These four ROMs are inserted into socket locations on the memory controller board.

CMI bits <23:02> from the CMI bus address the bootstrap ROMs. However, the memory controller cycles through the three least significant bits (<3:0>) in the process of forming the 32-bit data word from the 4-bit ROM outputs.

1.4.6 Battery Backup Option

When primary power is disconnected from the VAX-11/750, or a brownout occurs, battery backup circuitry provides the necessary dc power to preserve the MOS memory's contents for at least 10 minutes with full memory complement. During an ac power loss, +12 V and −5 V from the battery backup unit are applied to MOS memory, and +5 V is applied to certain sections of the memory control and driver section. The +5 V is applied to the memory controller refresh oscillator, refresh address counter, address multiplexer and address drivers, but not applied to the gate array logic. The green power LED remains lit to indicate that the memory controller is active. When the green power LED is lit, memory modules should not be removed.

1.4.7 Power Supply Requirements

The three modes of operation are defined as follows.

1. Active: Continuously running read/write memory cycles and refresh cycles
2. Standby: Running refresh cycles only (not on battery power)
3. Battery Backup: Running refresh cycles only (on battery backup)

Table 1-2 lists the power requirements for the memory controller and memory array boards.

Table 1-2 Power Supply Requirements

Voltages	Active	Standby	Battery
+12 VB ± 5%	9.9 A	1.8 A	1.8 A
−5 VB ± 5%	1.6 A	1.2 A	1.2 A
+5 VB ± 5%			
Eight array boards	8.4 A	5.1 A	5.1 A
Memory controller	4.0 A	4.0 A	4.0 A
+5 V ± 5%			
Eight array boards	4.1 A	4.1 A	0 A
Memory controller	2.7 A	2.7 A	0 A
+2.5 V ± 5%	3.1 A	3.1 A	0 A

CHAPTER 2
MEMORY CONTROLLER INTERFACE

2.1 CMI BUS ARCHITECTURE

Other devices use the synchronized tri-state CMI bus to communicate with the memory controller. Refer to Figure 2-1. The CMI bus has 32 lines to transfer the address/control information and data field between the memory controller and a master device. Four other lines are applied to the memory controller: two CMI bus status information lines, one CMI bus busy signal and one 6.7 MHz bus clock. The absence of the +5 V battery indicates the need for ECC initialization on power-up. Moreover, the CMI DC LO L signal indicates that all the power supply's dc voltages are within the specified limits. When an out-of-voltage condition occurs, CMI DC LO L is asserted.

2.1.1 Address/Control and Data Formats

Two types of formats are transferred over the 32-bit data lines of the CMI bus: an address/control format and a 32-bit data field. Refer to Figure 2-2. Immediately after a successful arbitration cycle, the ADDRESS/CONTROL information is applied to the CMI bus for one bus cycle.

During a write operation, the WRITE data is applied to the CMI bus in the bus cycle immediately following the transmission of the ADDRESS/CONTROL. For a read operation, the memory controller uses the 32-bit data format to transmit data for one bus cycle to the master device. Status information is also transmitted with the data format.

2.1.2 Status Lines

Status lines define the correctness of a data transfer. During the final bus cycle of a transaction, the slave device applies the status information to the CMI bus for one bus cycle.

In a write operation, CMI DBBZ L is not asserted by the slave device unless a previous fast-speed write operation is still in progress; the status lines become valid when CMI DBBZ L is deasserted by the master device. If an error is detected during the read-modify portion of a byte write operation, NO ERROR is indicated on the status lines and the erroneous data is preserved in memory. Thus, a write error is indicated when the data in error is read out of memory during a read operation.

The status lines are interpreted as shown in Table 2-1.

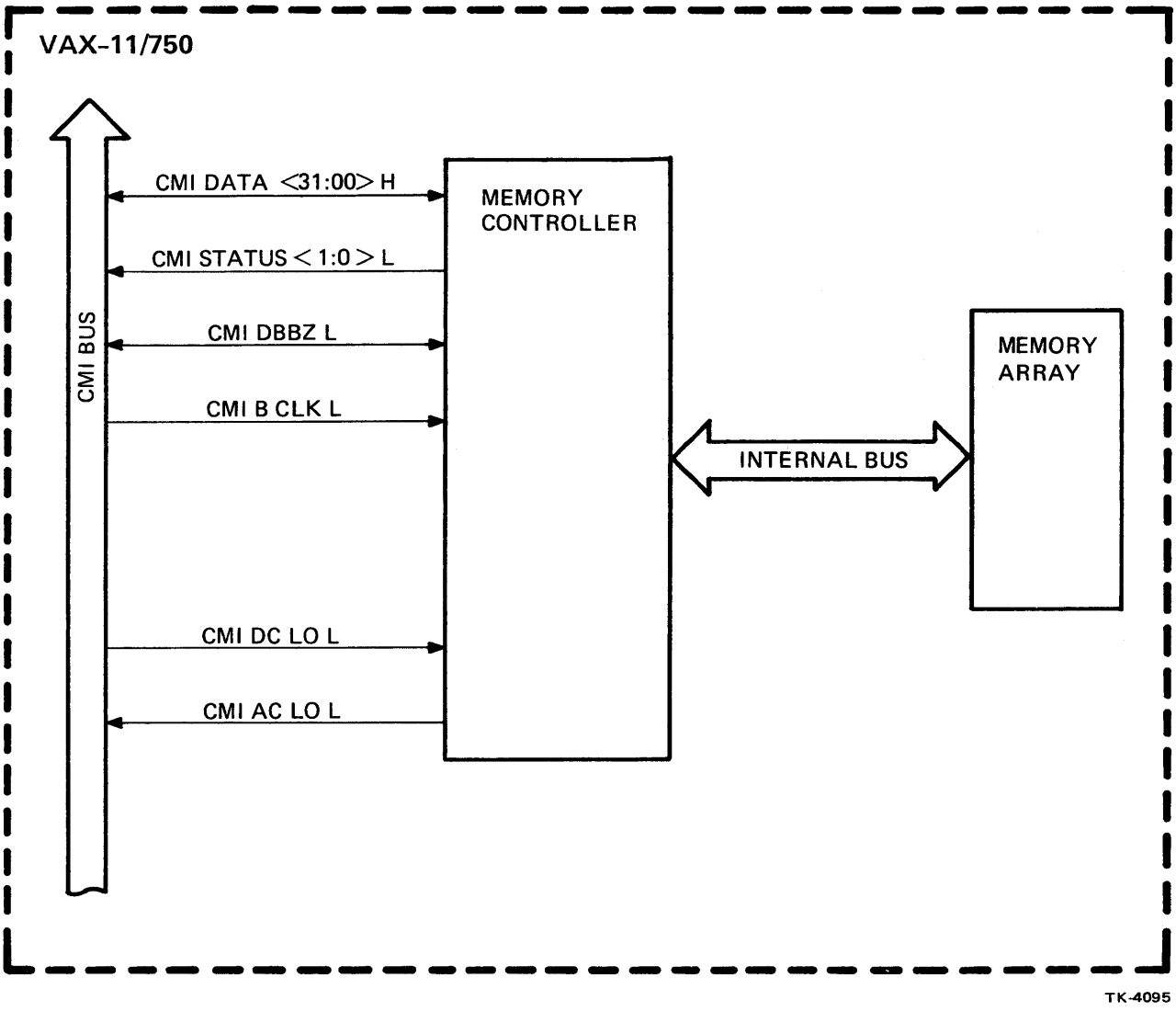

Figure 2-1 Memory Controller Interface

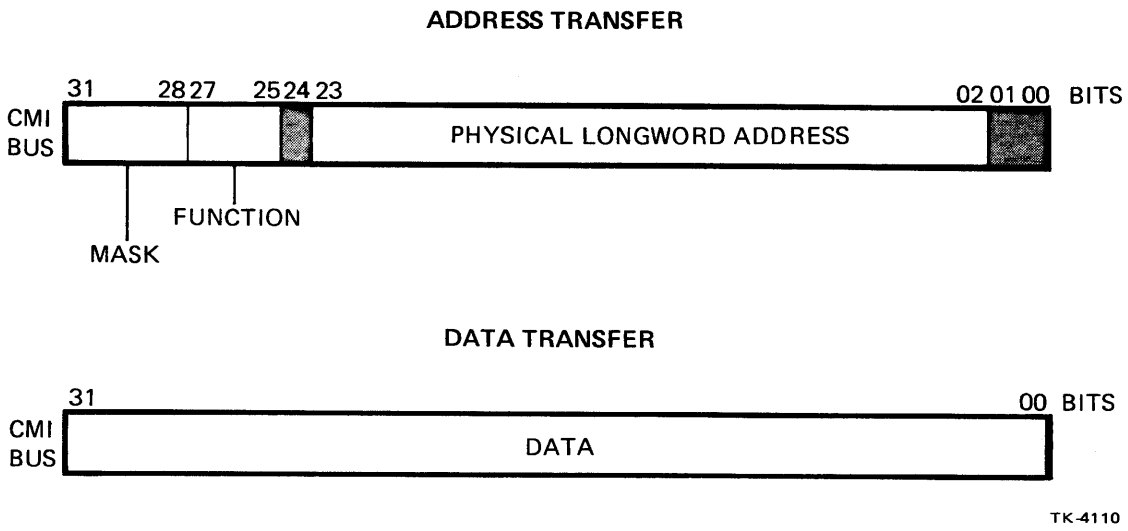

Figure 2-2 Address/Data Formats

Table 2-1 Status Interpretation

Status Bit 1	Status Bit 0	Interpretation
0	0	no response; nonexistent memory
0	1	data contains an uncorrectable error
1	0	data has been corrected
1	1	good status; data has no errors

2.1.3 CMI Bus Busy

The CMI DBBZ L signal indicates the availability of the CMI bus. When CMI DBBZ L is deasserted, a master device can take control of the CMI bus at the beginning of the next CMI cycle.

CMI DBBZ L is asserted by the bus master for one CMI bus cycle when the ADDRESS/CONTROL is applied to the CMI bus. However, during a read operation, the slave device asserts CMI DBBZ L in the cycle following the master device's assertion of CMI DBBZ L. The slave device then continues asserting CMI DBBZ L until the READ data is ready for transmission. In a write operation, CMI DBBZ L is not asserted by the memory controller if the memory controller was previously in the idle state.

2.1.4 CMI Bus Clock

The CPU provides a single-phase 6.7 MHz bus clock (CMI B CLK L) that allows synchronous system activity. The positive-going edge of CMI B CLK L begins the CMI bus cycles.

2.2 DATA TRANSACTIONS

CMI bus data transactions are divided into three steps:

1. arbitration (160 ns)
2. address/control format decoding (160 ns)
3. data transfer and status ($\geqslant$ 160 ns)

2.3 ARBITRATION

To transfer data between the master device (a CPU, for example) and the memory controller (always a slave device), the master device must gain control of the CMI bus. It does so by:

1. asserting an assigned priority bit that is higher in priority than the priority bits being asserted by other devices on the CMI bus
2. asserting CMI DBBZ L (if CMI DBBZ L was not asserted the previous bus cycle)
3. asserting the ADDRESS/CONTROL on the CMI bus for one bus cycle.

2.4 ADDRESS/CONTROL DECODING

The memory controller decodes the three fields of the 32-bit address/control format. These three fields are mask, function and address.

2.4.1 Address Field

The memory controller interprets the address field (CMI bits <23:02>) as shown in Table 2-2.

Table 2-2 Address Field

Bits	Function
<23:18>	selects a specific memory array
<17:16>	selects a bank of chips
<15:09>	selects a specific column of the memory chips
<08:02>	selects a specific row of the memory chips
<01:00>	not used in address selection

2.4.2 Function Field

The decoded function field (CMI bits <27:25>) determines the type of memory cycle. There are three types of memory cycles: read, long-word write and byte write. Table 2-3 shows the function field decoding.

Table 2-3 Function Field

Bits			Memory Cycle Type
27	26	25	
0	d	d	read
1	0	d	long-word write or byte write

NOTE: "d" means "don't care."

2.4.3 Mask Field

The mask field consists of the CMI bits <31:28> of the address/control format. Each 1 in the mask permits a specific byte in the 32-bit data word to be written into memory. Refer to Figure 2-3.

The mask field is decoded for each memory cycle as shown in Table 2-4.

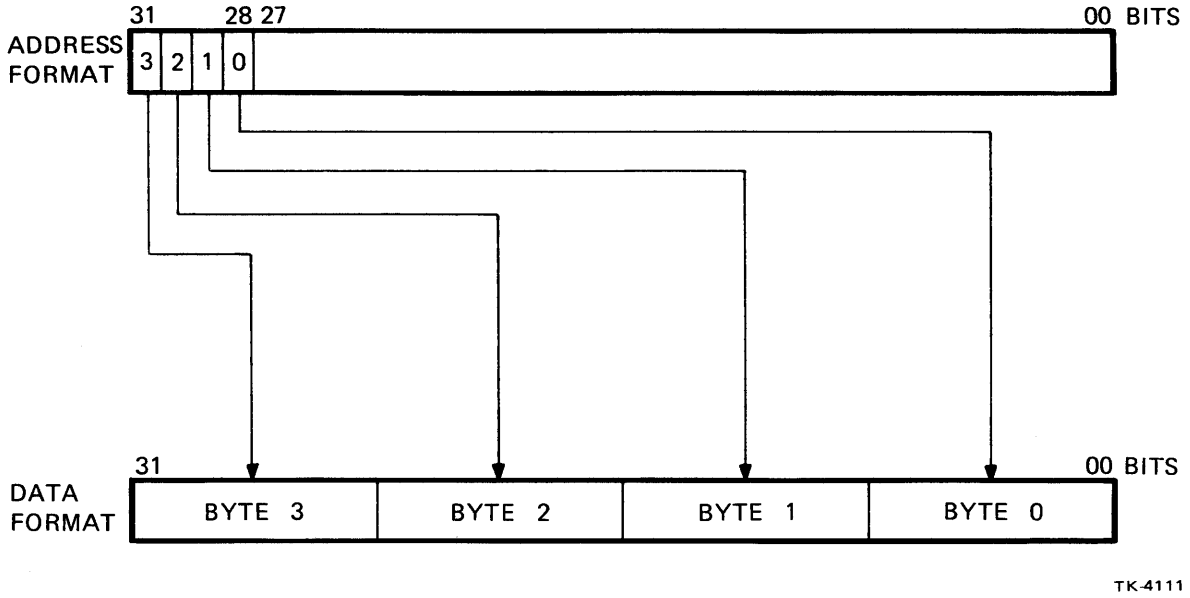

Figure 2-3 Mask Field Interpretation

Table 2-4 Mask Field

Bits				Memory Cycle
31	**30**	**29**	**28**	
d	d	d	d	read (always 4 bytes)
1	1	1	1	long-word write
[1≤ # 1s ≤3]				byte write

NOTE: "d" means "don't care."

2.5 READ AND WRITE OPERATIONS

The following descriptions of read and write operations are meant to serve as an aid to understanding how the memory controller functions. First, the descriptions are a synopsis of the read and write operations. Second, the signals are followed in parentheses by the number of the paragraph in Chapter 3 that describes them.

Two methods can be used to correlate the memory controller's detailed description with the L0011-0-1 logic prints. First, the descriptive paragraphs in Chapter 3 reference specific diagrams containing L0011-0-1's sheet numbers and component numbers (E numbers). Second, the fourth letter in the signal's mnemonic correlates with the L0011-0-1 sheet numbers.

Note that the following descriptions describe only the major events of a read and write operation. Therefore to determine the actual timing of these events, use the timing diagrams (not included in this manual).

2.5.1 Memory Read

A master device communicates with the memory controller via the CMI bus. During this communication period, other devices must be prevented from using the CMI bus. First, the master device asserts CMI DBBZ L (bus busy) for one bus cycle. Then, when the memory controller detects that there is a read operation to be performed, the memory controller asserts CMI DBBZ L (3.17) for the number of bus cycles required to complete the read operation.

In synchronization with CMI B CLK (3.12), the master device applies ADDRESS (CMI bits <23:02>) and CONTROL (CMI bits <31:25>) to the CMI bus for one bus cycle so that ADDRESS and CONTROL can be latched into the memory controller. Thus, CMCF LATCH IAR L (3.5): (1) latches CMI DATA <31:25> H into the intermediate address register decoding circuit (3.5), which generates the read operand, (2) latches CMI DATA <23:17> H into the memory array address generation circuit (3.2), which selects a memory array, and (3) latches CMI DATA <23:02> H into the data handling circuit (3.14), which stores the physical address.

At this point the controlling memory cycle sequence circuit (3.9) receives a ROM starting address so that it can generate the control timing signals to complete the read operation. The ROM starting address CMCB RSA <3:1> H (3.8) is created by the application of CMCL MEM LATCH BUS CYC REQ H (3.7) and CMCL RD OP H (3.8) to the generate cycle type request circuit (3.8).

Meanwhile, the physical address is applied to the memory array. First, CMCF MAR LATCH L (3.10) latches the ADDRESS into the memory address register of the address selection circuit (3.11). Next, INT BUS RAS TIM L (3.4) strobes the row address, and, in turn, INT BUS CAS TIM L strobes the column address into the memory array. Finally, CMCF INT BUS DR EN L (3.21) enables the tri-state drivers on the memory array so that the 32-bit word and the seven check bits contained in the selected address are read out onto the internal bus.

Here the word read out must be verified for a read error. Thus, the 7 check bits and 32 bits of data from the internal bus are latched into the memory error correction circuit (3.15) by CMCF LW LATCH CB REG L (3.15) and CMCE MEC LATCH DATA IN H (3.15), respectively. The memory error correction circuit (3.15) checks for single- and double-bit errors; it corrects single-bit errors but leaves double-bit errors unaltered.

After it is checked and corrected, the READ is output from the memory error correction circuit (3.15) by CMCE MEC LATCH OUTPUT L, which applies the READ data to the data handling circuit (3.14) via the internal bus.

To complete the read operation, READ and STATUS are applied to the CMI bus. CMCH CMI DR EN H (3.14) enables the data handling circuit (3.14) to drive the READ, while CMCF DR STATUS H (3.16) enables STATUS to be driven. At this point both READ and STATUS are driven onto the CMI bus.

2.5.2 4-Byte Write

A master device writes to the memory controller via the CMI bus. To prevent other devices from using the CMI bus when the memory controller is being written to, the master device asserts CMI DBBZ L (bus busy) on the CMI bus for one bus cycle. If, at the next bus cycle, a previous fast-speed write operation is still being processed, the memory controller asserts CMI DBBZ L (3.17) for the number of bus cycles needed to complete the write operation.

In synchronization with CMI B CLK L (3.12), the master device asserts ADDRESS, which specifies the location in the memory array to be written into. It then asserts CONTROL, which places the write operand and byte mask onto the CMI bus so that ADDRESS and CONTROL can be latched into the memory controller. Therefore, CMCF LATCH IAR L (3.5) latches the ADDRESS and CONTROL into the memory controller using the following scheme. First, CMI DATA <23:17> H is latched into the memory array address generation circuit (3.2) to determine which memory array is to be selected, while CMI DATA <32:02> H is latched into the data handling circuit (3.14) to store the physical address. Second, CMI DATA <31:25> H is latched into the intermediate address register decoding circuit (3.5) to generate the byte mask code and the write operand.

Next, the memory cycle sequence circuit (3.9) begins to generate the control timing signals to complete the 4-byte write operation. However, to start the generation of the control timing signals, the memory cycle sequence circuit (3.9) must receive the ROM starting address (CMCB RSA <3:0> H) for the byte write operation. CMCB RSA <3:0> H (3.8) is created from CMCL LATCH BUS CYC REQ H (3.3), CMCL WR OP H (3.8) and CMCL ALL BYTES L (3.5).

The WRITE (the four bytes to be written into memory) is then transferred to the memory error correction circuit (3.15) so that check bits can be generated for it. Therefore, the WRITE is transferred from the CMI bus through the memory data register of the data handling circuit (3.14), via the internal bus to the memory error correction circuit (3.15) by CMCE <3:0> DR EN H (3.15). CMCE MEC LATCH DATA IN H (3.15) then latches the WRITE into the memory error correction circuit (3.15) to generate the check bits for the WRITE. Next, the check bits and WRITE are output from the memory error correction circuit (3.15) onto the internal bus by CMCF HW OUTPUT CB SYN H and CMCE LW/HW OUTPUT BYTE <1:0> H, respectively.

At this point the WRITE is written into memory. Previously, ADDRESS from the data handling circuit (3.14) was latched into the memory address register of the address selection circuit (3.11) by CMCF MAR LATCH L (3.10). INT BUS RAS TIM L (3.4) strobes the row address and INT BUS CAS TIM L strobes the column address into the memory array. Therefore, with a selected address and the assertion of INT BUS WR TIM L, WRITE is written into the memory array from the internal bus. Finally, CMCH DR STATUS H (3.17) drives STATUS onto the CMI bus.

2.5.3 Byte Write

A master device writes to the memory controller via the CMI bus. To prevent other devices from using the CMI bus when the memory controller is being written to, the master device asserts CMI DBBZ L (bus busy) on the CMI bus for one bus cycle. If, at the next bus cycle, a previous fast-speed write operation is still being processed, the memory controller asserts CMI DBBZ L (3.17) for the number of bus cycles needed to complete the write operation.

In synchronization with CMI B CLK L (3.12), a master device applies ADDRESS (CMI bits <23:02>) and CONTROL (CMI bits <23:25>) to the CMI bus for one bus cycle so that ADDRESS and CONTROL can be latched into the memory controller. Therefore, CMCF LATCH IAR L (3.5): (1) latches CMI DATA <31:25> H into the intermediate address register decoding circuit (3.5) so as to latch the byte mask and generate the write operand, (2) latches CMI DATA <23:17> H into the memory address generation circuit (3.2) to select a memory array, and (3) latches CMI DATA <23:02> H into the data handling circuit (3.14) to store the physical address.

At this point the controlling memory cycle sequence circuit (3.9) receives a ROM starting address in order to generate the control timing signals to complete the write operation. The ROM starting address CMCB RSA <3:0> H (3.8) is created by the application of CMCL MEM LATCH BUS CYC REQ H (3.7) and CMCL WR OP H (3.8) to the generate cycle type request circuit (3.8).

Meanwhile, the physical address is applied to the memory array. First, CMCF MAR LATCH L (3.10) latches the ADDRESS into the memory address register of the address selection circuit (3.11). INT BUS RAS TIM L (3.4) then strobes the row address, and, in turn, INT BUS CAS TIM L strobes the column address into the memory array. Finally, CMCF INT BUS DR EN L (3.21) enables the tri-state drivers on the memory array so that the four bytes of data and the seven check bits contained in the selected address are read out onto the internal bus.

Next, the newly formed word that is to be written into memory is created. First, the seven check bits and four data bytes from the internal bus that were read from the memory array are latched into the memory error correction circuit's input registers (3.15) by CMCF LW LATCH CB REG L (3.15) and CMCE MEC LATCH DATA IN H (3.15), respectively. The four bytes in the input register are routed within the memory error correction gate arrays and stored in the memory error correction circuit's output register. The bytes from the memory error correction's output registers that are to be preserved are applied to the internal bus again by CMCE HW/LW OUTPUT BYTE <1:0> H. Similarly, the master device applies the new bytes to be written to the CMI bus. These new bytes are transferred through the data handling circuit (3.14) onto the internal bus by CMCE <3:0> DR EN H.

New check bits must now be generated for the newly formed word that is on the internal bus. The new data word is latched into the memory error correction circuit (3.15) and the new check bits are generated for the data word.

Finally, the new word and its check bits are written into memory, with STATUS applied to the CMI bus. Thus, with the assertion of INT BUS WR TIM L (3.21), the four bytes and the check bits are written into memory. In addition, the STATUS is applied to the CMI bus by CMCH DR STATUS H (3.17).

2.5.4 CSR/ROM Read or Write

A master device communicates with the memory controller via the CMI bus. Thus, other devices must be prevented from using the CMI bus during this communication period. First, the master device asserts CMI DBBZ L (bus busy) for one bus cycle. When the memory controller detects there is a new read cycle to be performed, it asserts CMI DBBZ L (3.17) for the number of bus cycles needed to complete the data transaction. But if a fast-speed write operation is detected, and if the previous write is still in progress, CMI DBBZ L (3.17) is asserted by the memory controller.

In synchronization with CMI B CLK L (3.12), the master device asserts ADDRESS, which specifies if CSR0, CSR1, CSR2 or a bootstrap ROM is being addressed. The master device then asserts CONTROL, which determines a read or a write operation to be placed onto the CMI bus so that these applied signals can be latched into the memory controller. Therefore, CMCF LATCH IAR L (3.5) latches ADDRESS (CMI DATA <23:02> H) into the data handling circuit (3:14) and CONTROL (CMI DATA <31:25> H) into the intermediate address register decoding circuit (3.5).

Next, it must be determined if a specific CSR or bootstrap ROM is selected. First, bits <23:11> are decoded to determine if either a CSR or a bootstrap ROM is addressed; if so, CMCN HI ROM and CSR L (3.6) become asserted. However, address bit 10 determines either a CSR or a bootstrap ROM selection.

To select a specific CSR, address bits <8:4> are first checked to verify that they are all 0s. If a CSR read request is pending, address bits <3:2> then select either CSR0 or CSR1. However, if a CSR write request is pending, address bits <3:2> produce either CMCM CSR 0 WR CY L (3.6) or CMCM CSR 1 WR CY L (3.6).

Similarly, to select a specific bootstrap ROM and a specific ROM location, address bits <9:2> are decoded. First, address bits <9:8> are decoded to determine which one of the four bootstrap ROMs is selected. Then, address bits <7:2> designate the upper six bits of the ROM's internal 9-bit address, while the three lower bits of the ROM address are derived from the controlling memory cycle sequence circuit (3.9). The lower three bits (CMCB RSA <3:1> H)(3.9) are sequentially incremented by the controlling memory cycle sequence circuit (3.9) so that eight consecutive 4-bit data words can be read out to form a 32-bit word.

At this point the controlling memory cycle sequence circuit (3.9) receives a ROM starting address so that it can generate the control timing signals to complete the read or write operation. The ROM starting address is produced in the following way. By decoding CONTROL, the intermediate address decoding circuit (3.5) produces either CMCL RD OP H (3.5) or CMCL WR OP H (3.5). While decoding the address, the CSR and ROM decoding circuit (3.6) and the generating next cycle request circuit (3.7) produce either a CSR request (CMCN CSR REQ H) (3.7) or a ROM request (CMCN ROM REQ H) (3.7). The CSR or ROM request and the read or write request are applied to the generating cycle type circuit (3.8), which produces a specific ROM starting address (CMCB RSA <3:0> H) for either a CSR read, CSR0 write, CSR1 write or a ROM read.

The control timing signals that complete a CSR read or ROM read perform the following. First, CMCJ MDL OUTPUT CONTROL <2:1> L (3.14) are applied to the data handling circuit (3.14) to determine if either CSR0, CSR1, CSR2 or a bootstrap ROM is read onto the CMI bus. During each CSR read, CMCF CSR2 TO D BUS EN L applies the contents of CSR2 to the internal bus, but CMCJ MDL OUTPUT CONTROL <2:1> L decides if those contents are applied to the CMI bus. While in ROM read, CMCE MDL <3:0> LATCH REG <2:1> H latches the 32-bit ROM data word into the bootstrap ROM word assembly register in sequential 4-bit slices. By the assertion of CMCH CMI DR EN H, the selected data is applied to the CMI bus.

To write to either CSR0 or CSR1, the control timing signals are used in the following way. CMCE MDL <3:0> LATCH REG <2:1> H enables CSR1 and CSR0 on a per-byte basis so that data can be written into them. A WRITE from the CMI bus is applied to CSR0 and CSR1 by the assertion of CMCM CSR WR CY L (3.14); the WRITE is latched into either CSR0 or CSR1 by CMCE MDL <3:0> LATCH REG <2:1> H (3.14). Finally, CMCF DR STATUS H (3.16) drives STATUS onto the CMI bus.

2.6 REFRESH

A delay-line oscillator determines when a refresh operation is performed. Hence, every 220 ns, a delay-line oscillator produces CMCC OSC H (3.13) and CMCC DEL OSC L (3.13). First, CMCC OSC H (3.13) increments a 6-stage refresh frequency counter that contains the refresh row address. Second, the refresh frequency counter's 6-stage output, CMCC DEL OSC H (3.13) and the deasserted CMCA KILL REF L (3.12) produce the request CMCA REF OSC REQ H.

CMCA REF OSC REQ H (3.13) performs two functions. Initially, it produces the row address strobe INT BUS RAS TIM L (3.4) by being applied to the generating RAS and CAS circuit (3.4) via the generating next cycle request circuit (3.7). At the end of each refresh cycle, CMCA REF OSC REQ H (3.13) also increments the refresh frequency counter.

The controlling memory cycle sequence circuit (3.9) receives a ROM starting address to generate the control timing signals that are used to complete the refresh operation. With the assertion of CMCF REF OR CYC 0 H and CMCB REF REQ H (3.13), the generating cycle type circuit (3.8) produces the ROM starting address CMCB RSA <3.0> H (3.8).

Finally, the row address is applied to the memory arrays. CMCD INIT OR REF H, the inversion of CMCM REF CY L (3.13), applies the refresh row address from the refresh frequency counter to the memory arrays via the address selection circuit (3.11).

CHAPTER 3
DETAILED DESCRIPTION

3.1 INTRODUCTION

The memory controller, a single extended hex-multilayer board, is the interface between the CMI bus and the memory arrays. The memory controller performs reads, long-word writes, byte writes, bootstrap ROM reads, error detection, and single-bit error correction. The memory controller also contains circuitry that performs refresh operations to maintain the contents of the MOS memory arrays.

The following description is based on Figure 3-1, Sheets 1–3. (Note that only the most significant signals are shown.)

3.1.1 Addressing

Bits <23:17> from the CMI bus address are applied to the memory array address generation circuit (Sheet 1). The memory array address generation circuit decodes address bits <23:17> to determine which memory array (Sheet 3) is selected by applying INT BUS ADD MEM SEL <7:0> to a memory array board.

Fingerprints and memory present signals from the internal bus are input to the memory array address generation circuit (Sheet 1). The memory present signals indicate which slots contain memory array boards, while the fingerprints indicate which memory arrays are or are not fully populated. CMCL MEM PRES H is applied to both the memory cycle request generation circuit (Sheet 1) and the generating RAS and CAS circuit (Sheet 1).

The generating RAS and CAS circuit produces CMCE COL ADD MUX L, which controls the multiplexing of the row and column addresses into the address selection circuit (Sheet 2). The generating RAS and CAS circuit produces the row address strobe (INT BUS RAS TIM L), and the column address strobe (INT BUS CAS TIM L), which strobes the row and column addresses respectively into the memory array.

These address strobes are produced when: (1) the memory controller is in cycle 0 (CMCB CYC 0 NEXT H), (2) the addressed memory location is present (CMCL MEM PRES H), (3) there is a new bus cycle request (CMCL DBBZ NOW H and CMCL DBBZ WAS L) and a legitimate cycle type (CMCL IAR NO CYC H), and (4) the proper control timing signals (CMCB TI RA <7:0> H) are applied from the controlling memory cycle sequence circuit (Sheet 2).

CMI address bits <17:02> are transferred through the data handling circuit (Sheet 2) to the address selection circuit (Sheet 2). The address bits <17:02> are multiplexed onto the internal bus (A <06:00>) to form the row and column addresses of the selected memory array.

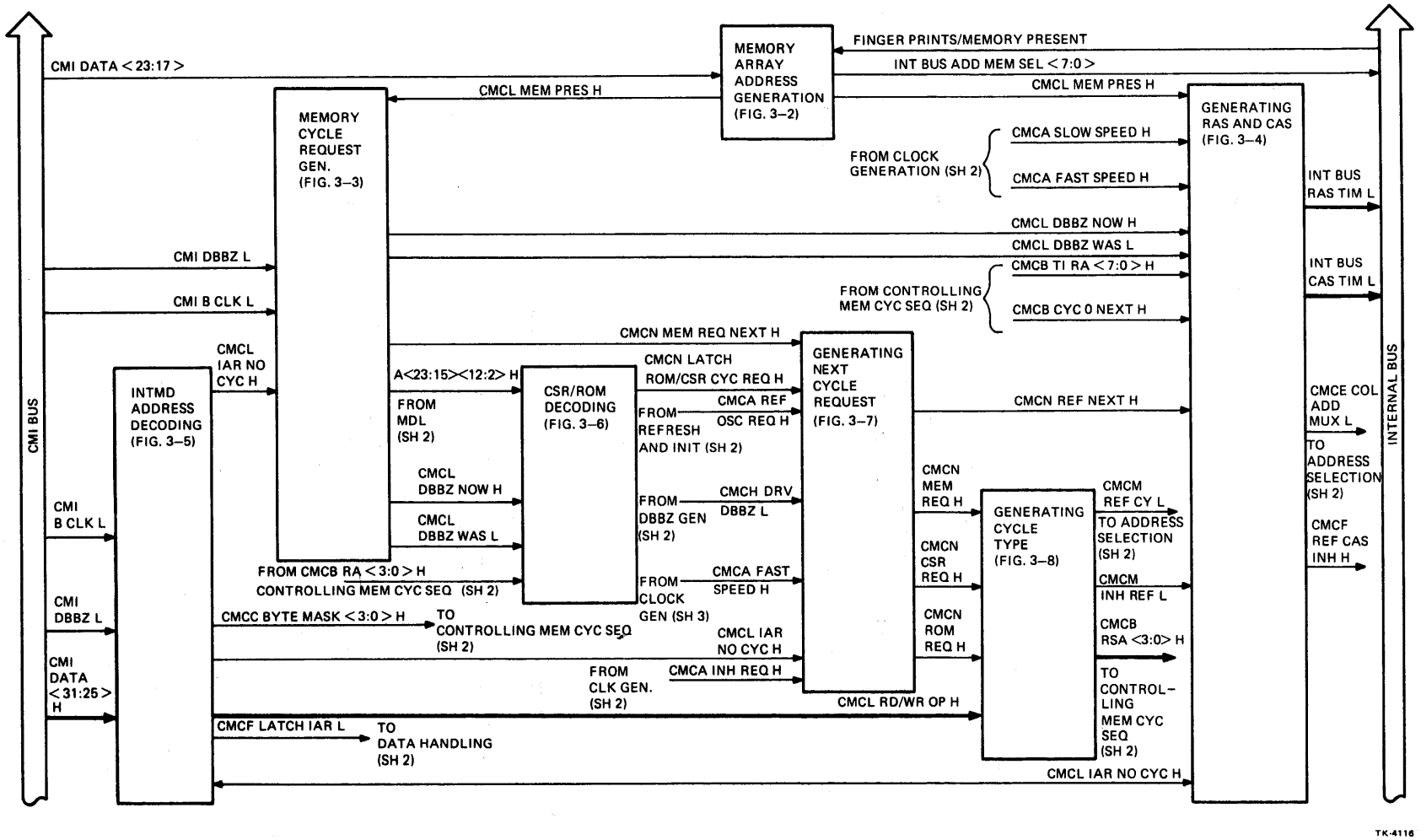

Figure 3-1 Memory Controller (Sheet 1 of 3)

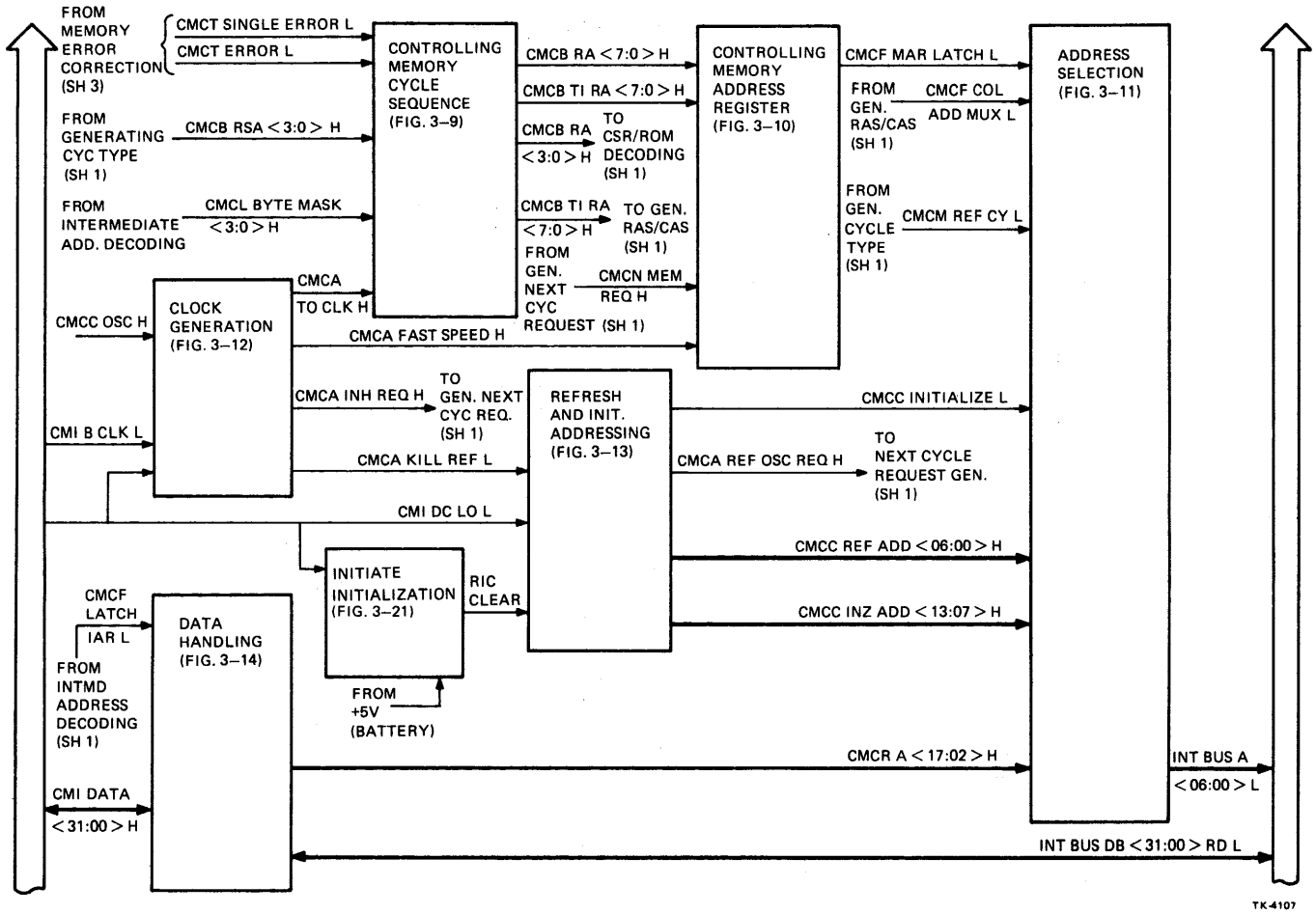

Figure 3-1 Memory Controller (Sheet 2 of 3)

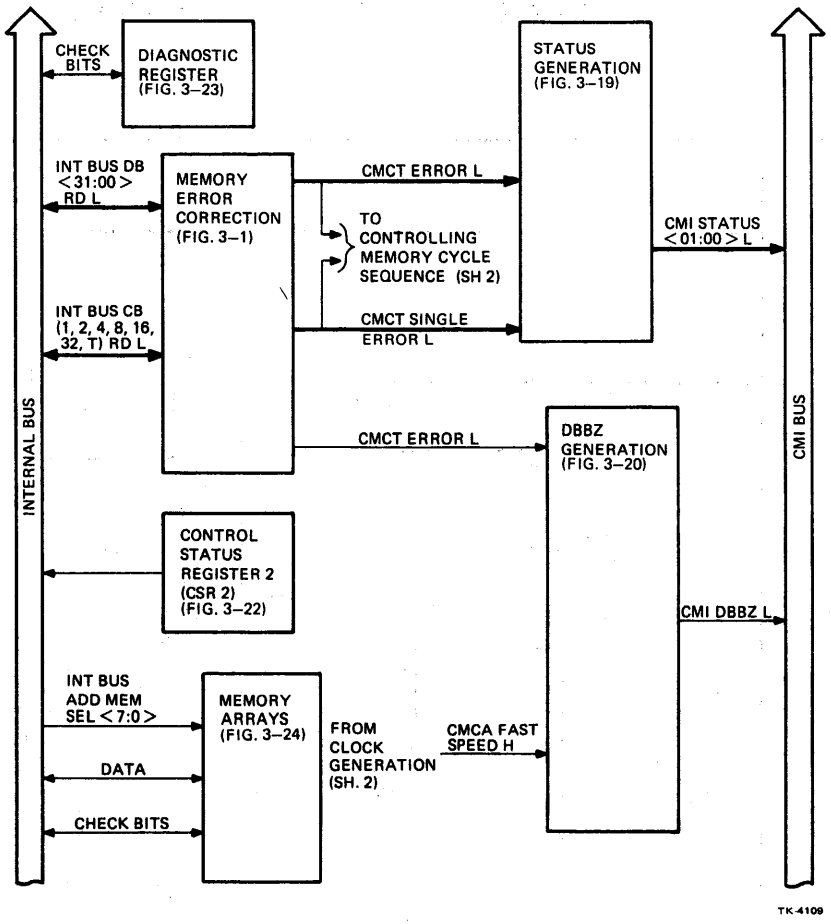

Figure 3-1 Memory Controller (Sheet 3 of 3)

3.1.2 Cycle Type Generation

When there is a new bus cycle request determined by CMI DBBZ L, bits <31:25> of the address format are clocked (CMI B CLK L) into the intermediate address decoding circuit (Sheet 1). The decoded bits <31:25> cause the intermediate address decoding circuit (Sheet 1) to produce the byte mask (CMCC BYTE MASK <3:0> H), a read or a write operand (CMCL RD/WR OP H), and a legitimate cycle type signal (CMCL IAR NO CYC H).

The memory cycle request generation circuit (Sheet 1) monitors the CMI bus busy signal CMI DBBZ L at bus clock time (CMI B CLK L) to determine if there is a new bus cycle request. In other words, CMI DBBZ is currently asserted but was not asserted the previous bus cycle.

The generating next cycle request circuit (Sheet 1) determines which cycle request is next. The next cycle is determined by the assertion of one of the following cycle request inputs to the generating next cycle request circuit: CMCN MEM REQ NEXT H, CMCN LATCH ROM CYC REQ H, CMCN LATCH CSR CYC REQ H or CMCA REF OSC REQ H.

CMCN MEM REQ NEXT H is derived from the memory cycle request generation circuit when there is a new bus cycle request, while both CMCN LATCH ROM CYC REQ H and CMCN LATCH CSR CYC REQ H are derived from the CSR and ROM decoding circuit. Furthermore, the CSR and ROM decoding circuit monitors for a new bus cycle request and also for a CSR or ROM to be addressed. Upon a refresh request, a refresh and initialization addressing circuit applies CMCA REF OSC REQ H. Moreover, the generating next cycle request circuit also verifies that there is a legitimate cycle type decoded in the intermediate address decoding circuit (CMCL IAR NO CYC H), and that there is also no cycle inhibit request (CMCA INH REQ H) from the clock generation circuit (Sheet 2).

A cycle request (CMCN MEM REQ H, CMCN CSR REQ H or CMCN ROM REQ H) from the generating next cycle request circuit (Sheet 1) is applied to the generating cycle type circuit (Sheet 1). Also, a read or write operand (CMCL RD/WR OP H) from the intermediate address decoding circuit (Sheet 1) is applied to the generating cycle type circuit (Sheet 1).

The generating cycle type circuit (Sheet 1) applies a refresh cycle (CMCM REF CY L) to the address selection circuit (Sheet 2). Furthermore, the inhibit refresh (CMCM INH REF L) is applied to the generating RAS and CAS circuit (Sheet 1). By using the cycle request, the generating cycle type circuit produces the ROM starting address (CMCB RSA <3:0> H), which is applied to the controlling memory cycle sequence circuit (Sheet 2).

By using the ROM starting address (CMCB RSA <3:0> H) from the generating cycle type circuit, the byte mask (CMCL BYTE MASK <3:0> H), the error flags from the intermediate address decoding circuit (Sheet 1), and the controlling memory cycle sequence circuit (Sheet 2) produce the control timing signals (CMCB T0 RA <7:0> H and CMCB TI RA <7:0> H), which control the sequential cycles of the cycle types. Moreover, CMCB RA <3:0> H is applied to the CSR and ROM decoding circuit (Sheet 1) to step the bootstrap PROMs sequentially, while CMCB TI RA <7:0> H is applied to the generating RAS and CAS circuit (Sheet 1). Both CMCB RA <7:0> H and CMCB TI RA <7:0> H are applied to the controlling memory address register circuit (Sheet 2), which produces CMCF MAR LATCH L.

3.1.3 Address Selection

CMCF MAR LATCH L latches the memory address (CMCR A <17:02> H) into the address selection circuit (Sheet 2) to select a memory array address. More specifically, the row, then the column address, are multiplexed by CMCF COL ADD MUX L into the address selection circuit that applies the address selection signals INT BUS A <06:00> L to the memory array (Sheet 3) via the internal bus.

3.1.4 Data Transfer and Corrections

The 32-bit read or write data words are transferred to or from the CMI bus via the data handling circuit (Sheet 2). Error detection and single-bit error corrections are performed by the memory error correction circuit (Sheet 3). The memory error correction circuit checks the whole 32-bit word (INT BUS BD <31:00> RD L) for bit errors, and also generates the check bits (INT BUS CB 1,2,4,8,16,32,T RD L) for the 32-bit data word.

The memory error correction circuit (Sheet 3) produces two error-indicating signals (CMCT ERROR L and CMCT SINGLE ERROR L). These signals are applied to the status generation circuit (Sheet 3), which applies single-bit error status to the CMI bus. CMCT ERROR L is also applied to the DBBZ generation circuit (Sheet 3) and to the controlling memory cycle sequence circuit (Sheet 2).

3.1.5 Bus Busy

The assertion of CMI DBBZ L prevents other devices from using the CMI bus. Therefore, during the bus cycle following the application of the address/command data to the CMI bus, the DBBZ generation circuit (Sheet 3) applies the CMI bus busy signal CMI DBBZ L to the CMI bus. The assertion of CMCT ERROR L causes the DBBZ generation circuit to extend the assertion of CMI DBBZ L for one bus cycle so that the memory controller has time to perform a single-bit error correction.

3.1.6 Clock Generation

The clock generation circuit (Sheet 2) monitors CMI B CLK L. When CMI B CLK L is not present, the clock generation circuit switches from fast-speed mode (bus clock) to slow-speed mode (internal clock). Thus the clock generation circuit produces the cycle request inhibit signal CMCA INH REQ H. Next, CMCA INH REQ H is applied to the generating next cycle request circuit (Sheet 1) to prevent a cycle request while the memory controller is switching clocks. CMCA KILL REF L is applied from the clock generation circuit to a refresh and initialization circuit (Sheet 2) to prevent a refresh cycle while the memory controller is switching clocks.

3.1.7 Initialization and Refresh

The initialize initialization circuit (Sheet 2) starts the initialization cycle. This cycle is performed when main ac power is applied and the memory system's battery backup is not present or has been extended beyond the life of the batteries. Therefore, the refresh and initialization circuit (Sheet 2) produces the refresh address (CMCC REF ADD <13:07> H) and the initialization address (CMCC INZ ADD <13:07> H), which are applied to the address selection circuit (Sheet 2). Also, the refresh and initialization circuit contains an internal clock that produces the refresh request CMCA REF OSC REQ H and also serves as the slow-speed mode clock.

3.1.8 CSR2 and Diagnostic Register

Control/status register 2 (CSR2) (Sheet 3) is a read-only register that contains a memory present map and a starting address. The memory present map indicates the amount of memory present in the VAX-11/750 and the locations in the memory backplane into which the memory array boards are inserted. The starting address, usually 0, is also defined in CSR2.

The diagnostic register (Sheet 3) contains seven check bits that are used in diagnostic mode to verify the functionality of the error correction logic (ECC).

3.2 MEMORY ARRAY ADDRESS GENERATION

The memory array address generation circuit consists of a memory configuration checking circuit, a memory present checking circuit, an address input latch, a starting address buffer, a memory array selecting circuit and an output latch. Refer to Figure 3-2.

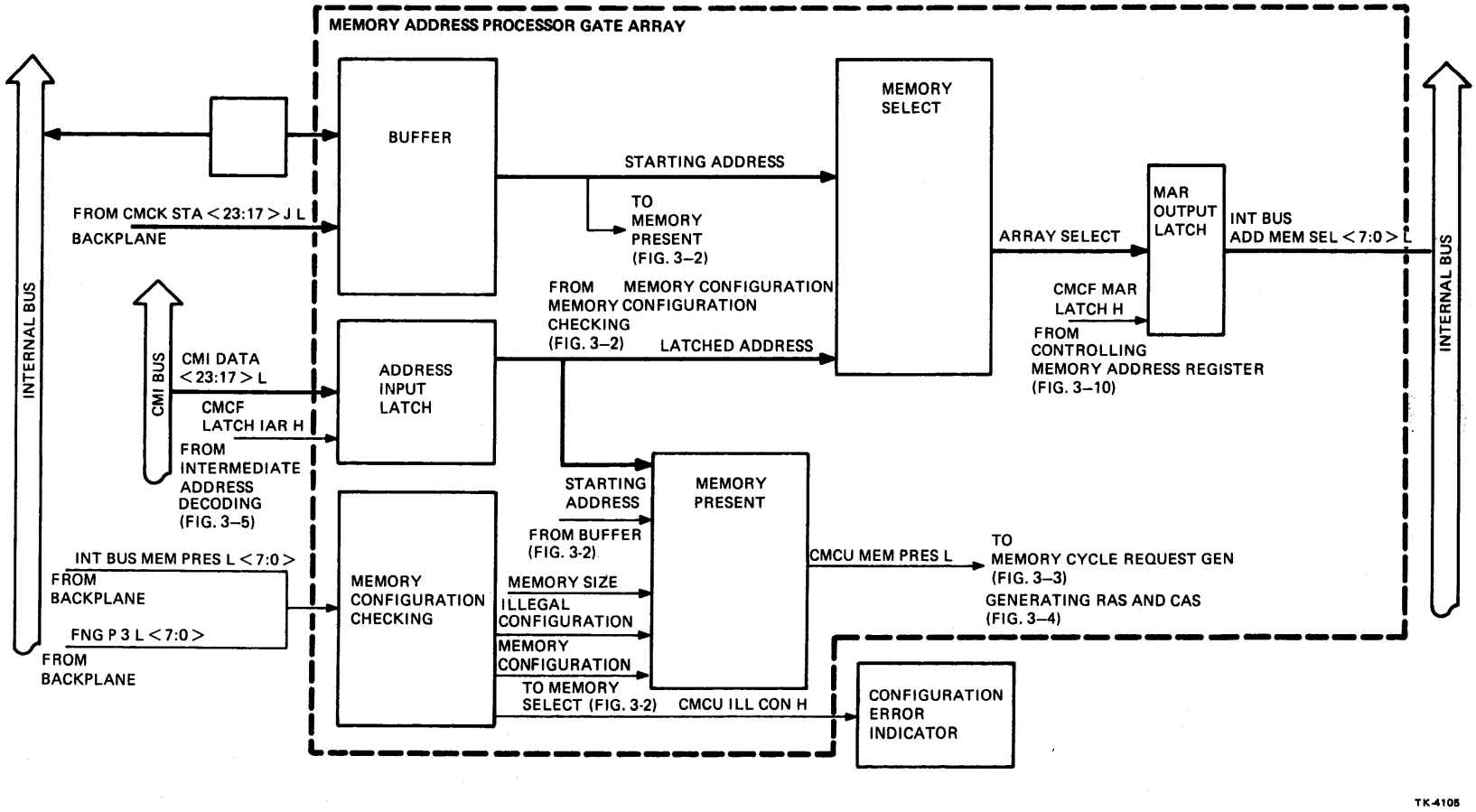

Figure 3-2 Memory Address Generation

This circuit decodes a 7-bit CMI bus address (CMI DATA <23:17> L) that is used to enable one of eight memory array boards. The address is decoded when CMI DATA <23:17> H is on the CMI bus and CMCF LATCH IAR L is asserted. The memory array address generation circuit then checks for the existence of an addressed memory location and examines if the memory array boards are correctly configured (that is, plugged into the right slots).

3.2.1 Memory Configuration Checking

Sixteen signals (INT BUS MEM PRES L <7:0> and FNGP 3 L <7:0>) are applied to the memory configuration checking circuit. These signals provide status data for the memory array boards, with INT BUS MEM PRES L indicating the presence of a memory array board. FNGP 3 L <7:0> indicates which memory array boards are and are not fully populated.

The memory configuration checking circuit applies information on memory size and illegal configurations to both the memory present checking circuit and the memory array selecting circuit. The signal CMCU ILL CON H (illegal configuration) is applied to the memory controller's red configuration error LED indicator. A legal memory configuration is one whose memory array(s) (up to eight) are asserted, starting with slot 11 and proceeding sequentially to slot 18. Also, partially populated 128K memory arrays may not precede one that is full.

3.2.2 Address Input Latch

The upper seven bits of the address (CMI DATA <23:17> H) from the CMI bus are latched into the address input latch when CMCF LATCH IAR L is asserted. The LATCHED ADDRESS out of the input latch is then applied to the memory present checking circuit and also to the memory array selecting circuit.

3.2.3 Memory Present Checking

Four signals are applied to the memory present circuit to determine if the addressed address exists. The input latch applies the LATCHED ADDRESS to the memory present circuit while the memory configuration checking circuit applies the MEMORY SIZE and ILLEGAL CONFIGURATION signal information; the buffer applies the starting address. When the conditions above are met, the memory present circuit's output becomes CMCU MEM PRES L.

3.2.4 Memory Array Selecting

A jumper-defined starting address (CMCK STA <23:17> J L) is applied from the backplane to the internal bus with the assertion of CMCF CSR2 TO D BUS EN L and to the buffers connected to the memory array selecting circuit and the memory present checking circuit. Next, the buffered starting address, MEMORY CONFIGURATION (from the memory configuration checking circuit), and the LATCHED ADDRESS are applied to the memory array selecting circuit, whose outputs (INT BUS ADD MEM SEL <7:0> L) are latched into the output latch by asserting CMCF MAR LATCH H. The internal bus signals INT BUS ADD MEM SEL <7:0> L from the output latch then select one of the eight memory array boards, that is, if the address is defined as valid.

3.3 MEMORY CYCLE REQUEST GENERATION

The memory cycle request generator consists of a DBBZ status monitoring circuit, a memory cycle clock flop, memory cycle request next flip-flops and a memory cycle in queuing circuit. Refer to Figure 3-3.

This circuit generates memory cycle requests when the memory controller is addressed and CMI DBBZ L is asserted (but was not asserted the previous bus cycle) on the CMI bus. Therefore, after proper circuit conditions have been checked, this circuit generates a CMI DBBZ L signal on the CMI bus. However, CMI DBBZ L is not asserted during a fast-speed write if there is no fast-speed write in queue. Thus, when asserted, this signal enables the memory controller to hold the CMI bus for memory data transfers.

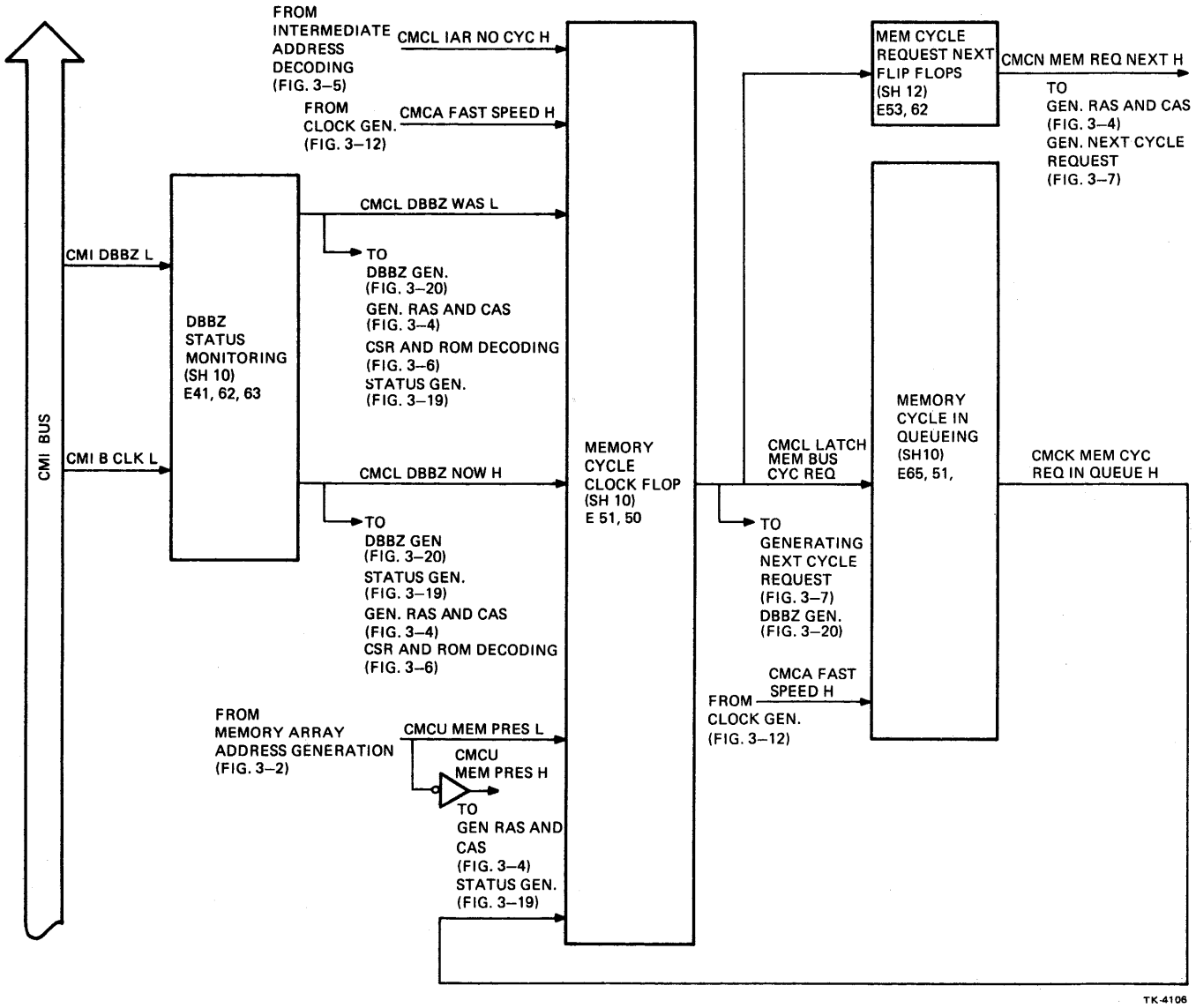

Figure 3-3 Memory Cycle Request Generation

3.3.1 DBBZ Status Monitoring

The DBBZ status monitoring circuit determines when the CMI bus is busy, that is, when the CMI DBBZ L signal is asserted. Initially, CMI DBBZ L is clocked into a flip-flop by CMCA BUS CLK H, and thus becomes CMCL DBBZ NOW L. One delay gate later, CMCL DBBZ NOW H is clocked by CMCF DEL CLK L into another flip-flop, whose output is clocked by CMCA BUS CLK H into another. This results in CMCL DBBZ WAS H, enabling the DBBZ status monitoring circuit to determine if CMI DBBZ L is asserted at the present bus cycle and was not asserted at the previous one.

3.3.2 Memory Cycle Clock Flop

The two outputs (CMCL DBBZ NOW L and CMCL DBBZ WAS H) of the DBBZ status monitoring circuit and also three signal conditions (memory is present, a legitimate cycle type is present, and the memory bus cycle is not in queue in fast-speed mode), are applied to the memory cycle clock flop. When all five of the above signal conditions are met, CMCL LATCHED MEM BUS CYC REQ H is clocked by CMCA BUS CLK H into a flip-flop.

While in slow-speed mode at time T1, CMCL LATCHED MEM CYC REQ H is clocked by CMCA T1 CLK H into the first flip-flop of the memory cycle request next flip-flops. At time T0, the first flip-flop's output of the memory cycle request next flip-flops is clocked by CMCA T0 CLK H into the second flip-flop, resulting finally in CMCN MEM REQ NEXT H.

3.3.3 Memory Cycle In Queuing

The memory cycle in queuing circuit checks for three conditions:

1. Is a new memory bus cycle request pending?
2. Is a memory bus cycle presently being performed?
3. Is the memory controller in fast-speed mode?

When these three conditions are met at bus clock time (CMCA BUS CLK H), the output of the memory cycle in queuing circuit becomes CMCL MEM CYC REQ IN QUEUE H.

3.4 RAS AND CAS GENERATION

The generating RAS and CAS circuits consist of an idle RAS start circuit, a queued RAS start circuit, a refresh RAS start circuit, a slow-speed RAS start circuit, a RAS store, a circuit RAS driver, and a TTL time delay circuit. Refer to Figure 3-4.

This circuit generates the row address strobe (RAS), the column address strobe (CAS), and the control signal CMCF COL ADD MUX L. These signals are generated at the beginning of either a memory cycle request (fast-speed), a queued memory cycle request (fast- or slow-speed), or a refresh cycle request. First, the RAS and CAS signals are used to strobe the row address and column address respectively into the memory arrays. Meanwhile, CMCF COL ADD MUX L is used to switch the row/column address multiplexer so that the column address may be applied to the memory arrays.

3.4.1 Idle RAS Start

The idle RAS start circuit checks for the following conditions:

1. When a memory cycle is requested, the memory controller is in the idle cycle state.

2. There is no request for a refresh cycle; the requested cycle type is legitimate and the addressed address exists.

3. The memory controller is in fast-speed mode.

When all the conditions above are met for one bus cycle, the idle RAS start circuit applies the signal START RAS FROM CYC 0 to the RAS store circuit.

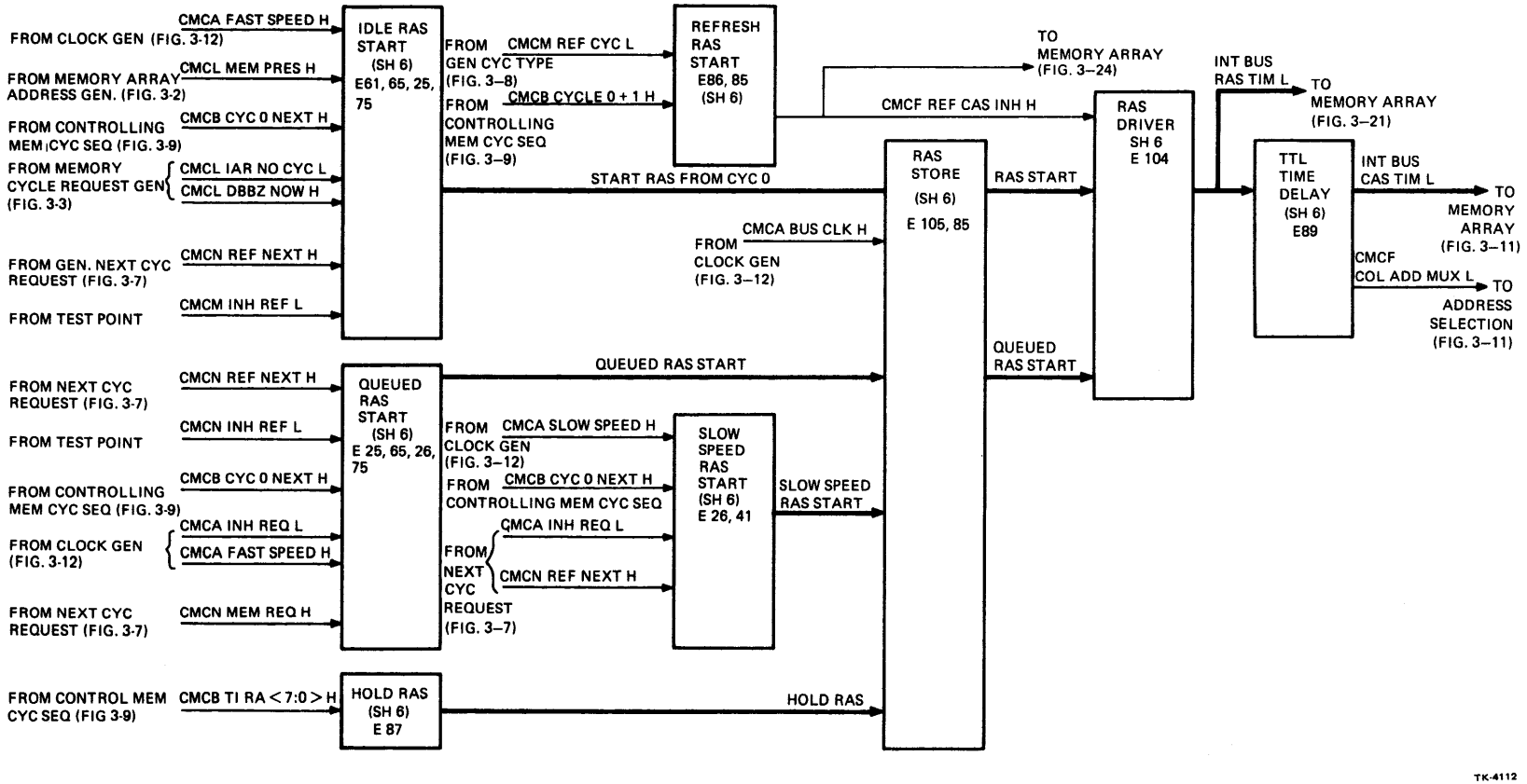

Figure 3-4 RAS and CAS Generation

3.4.2 RAS Store and Driver

The output signal START RAS FROM CYC 0 of the RAS start circuit is clocked by CMCA BUS CLK H into the RAS store, whose output RAS START is applied to the RAS driver, which produces the row address strobe signal (INT BUS RAS TIM L) for one cycle. Next, the signal INT BUS RAS TIM L is asserted for the proper number of cycles by the hold RAS circuit. The controlling memory cycle sequence circuit's output signals CMCB T1 RA <7:0> H are then applied to the hold RAS circuit. This causes HOLD RAS to be asserted.

INT BUS RAS TIM L is applied to the memory arrays on the internal bus and also directly to the TTL time delay circuit. CMCF COL ADD MUX L and INT BUS CAS TIM L are produced respectively 45 ns and 75 ns after INT BUS RAS TIM L is applied to the TTL time delay circuit.

3.4.3 Queued RAS Start

The queued RAS start circuit checks for the following.

1. The memory cycle request is in queue.
2. Cycle 0 is next.
3. No refresh cycle request is pending.
4. The memory is in fast-speed mode.
5. No inhibit request is pending.

When all the conditions above are met, the queued RAS start circuit produces QUEUED RAS START. This signal is applied to the RAS store circuit, after which it functions exactly as the START RAS CYC 0 signal.

3.4.4 Slow-Speed RAS Start

If the memory controller is in slow-speed mode, the slow-speed RAS start circuit is used to generate the signal INT BUS RAS TIM L. Moreover, the slow-speed RAS start circuit checks for the following conditions.

1. Cycle 0 is next.
2. No inhibit request is pending.
3. A memory request is next.
4. A refresh cycle is not next.

When all the slow-speed conditions are met, the slow-speed RAS start circuit produces SLOW SPEED RAS START, which functions exactly as START RAS FROM CYC 0.

3.4.5 Refresh RAS Start

During a refresh cycle request, the refresh RAS start circuit is used to generate the row strobe. When a refresh cycle request is pending and the memory controller is in cycle 0 or 1, the refresh RAS start circuit generates the signal CMCF REF CAS INH H. This signal is applied to the RAS driver circuit and, thus, the signal INT BUS RAS TIM L is produced. Note that the signal INT BUS RAS TIM L is applied to the TTL time delay circuit, which causes the signals CMCF COL ADD MUX L and INT BUS CAS TIM L to be produced. The application of INT BUS INH CAS TIM L to the memory arrays causes INT BUS CAS TIM L to be gated into each memory array.

3.5 INTERMEDIATE ADDRESS REGISTER DECODING

Command cycle enable logic, a command cycle latch, a command cycle decoder, and a next cycle hold latch comprise the intermediate address register decoding circuit. Refer to Figure 3-5.

This circuit decodes the upper seven bits (CMI DATA <31:25> H) of the CMI bus to generate the byte mask and a read or a write operand. Therefore, during each address bus cycle, the CMI bits <31:25> (CMI DATA <31:25> H) are decoded if CMI DBBZ L determines there is a new data transaction pending.

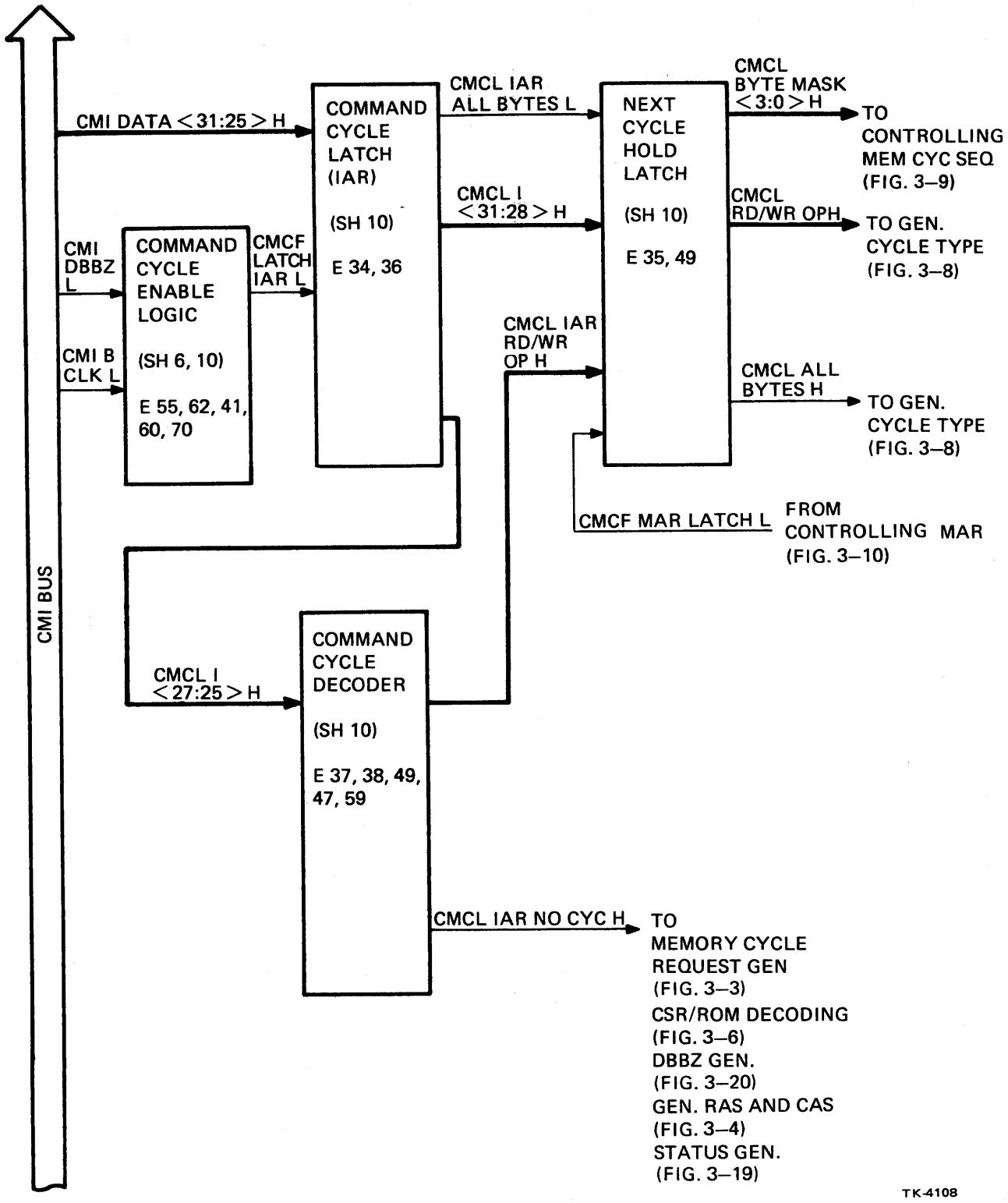

Figure 3-5 Intermediate Address Register Decoding

3.5.1 Command Cycle Latch

At the positive transition of the bus clock (CMI B CLK L), the CMI bus signal CMI DBBZ L is asserted by the bus master and applied to the command cycle enable logic, which then asserts CMCF LATCH IAR L. The assertion of CMCF LATCH IAR L causes the upper seven CMI bits (CMI DATA <31:25> H) to be stored in the command cycle latch, thus producing three output signals: CMCL IAR ALL BYTES L, CMCL I <31:28> H and CMCL I <27:25> H.

3.5.2 Command Cycle Hold Latch

CMCF MAR LATCH L enables CMCL IAR ALL BYTES L and also CMCL <31:28> H of the command cycle latch to be latched into the next cycle hold latch circuit, which then produces CMCL BYTE MASK <3:0> H and CMCL ALL BYTES H.

3.5.3 Command Cycle Decoder

The command cycle latch circuit's output CMCL I <27:25> H is applied to the command cycle decoder circuit, which generates the read/write operand (CMCL IAR RD/WR OP H) in addition to CMCL IAR NO CYC H. A deasserted CMCL IAR NO CYC H indicates there is a legitimate cycle code, while CMCF MAR LATCH L latches either a read or a write operand (CMCL IAR RD/WR OP H) into the next cycle hold latch circuit.

3.6 CSR AND ROM DECODING

The fundamental elements of the CSR and ROM decoding circuit are a ROM or CSR selection logic, a bootstrap PROM decoder select circuit, bootstrap PROMs, a CSR read decoder circuit and an MDL operation control circuit. Refer to Figure 3-6.

Address bits <23:04> (CMC R/S A <23:04> H) are applied to this circuit to select either a control/status register or a bootstrap PROM. The address selection of CSRs and PROMs is performed when a legitimate cycle type (CMCL IAR NO CYCLE L), a new bus cycle request (CMCL DBBZ NOW H and CMCL DBBZ WAS L) and the correct address are present.

The CSR and ROM decoding circuit performs two functions. First, it enables data transfers to and from CSR <1:0> or CSR2. Second, it selects the bootstrap PROMs that transmit the bootstrap program data to the data handling circuit.

3.6.1 ROM or CSR Selection Logic

The ROM or CSR selection logic circuit verifies that a bus cycle request is pending (CMCL DBBZ NOW H and CMCL DBBZ WAS L) and that the cycle type is legitimate (CMCL IAR NO CYCLE L).

CMCN HI ROM + CSR L is asserted (with either ROM or CSR selected) when one of the address bit conditions (CMCS/R A <21:11> H) is met. When CMCR A 10 H is asserted, CMCN LATCHED ROM BUS CYC REQ H is asserted (PROM selected). However, when CMCR A 10 H is deasserted and one of the address bit conditions (CMCR A <9:4> H) is met, CMCN LATCHED CSR BUS CYC REQ H is asserted (CSR selected).

3.6.2 Bootstrap PROM Decoder Select

Address bits <9:8> (CMCR A <9:8> H) are applied to the bootstrap PROM decoder select circuit. The four outputs CMCP SEL ROM <3:0> L of the bootstrap PROM decoder select circuit select one of the four bootstrap PROMs.

To address the bootstrap PROMs, the upper address bits <7:2> (CMCR A <7:2> H) from the data handling circuit, plus CMCB RA <3:1> H from the controlling memory cycle sequence circuit, are applied to the bootstrap PROMs while the lower address bits (CMCB RA <3:1> H) are applied and sequentially incremented so as to step through the bootstrap PROMs' addresses.

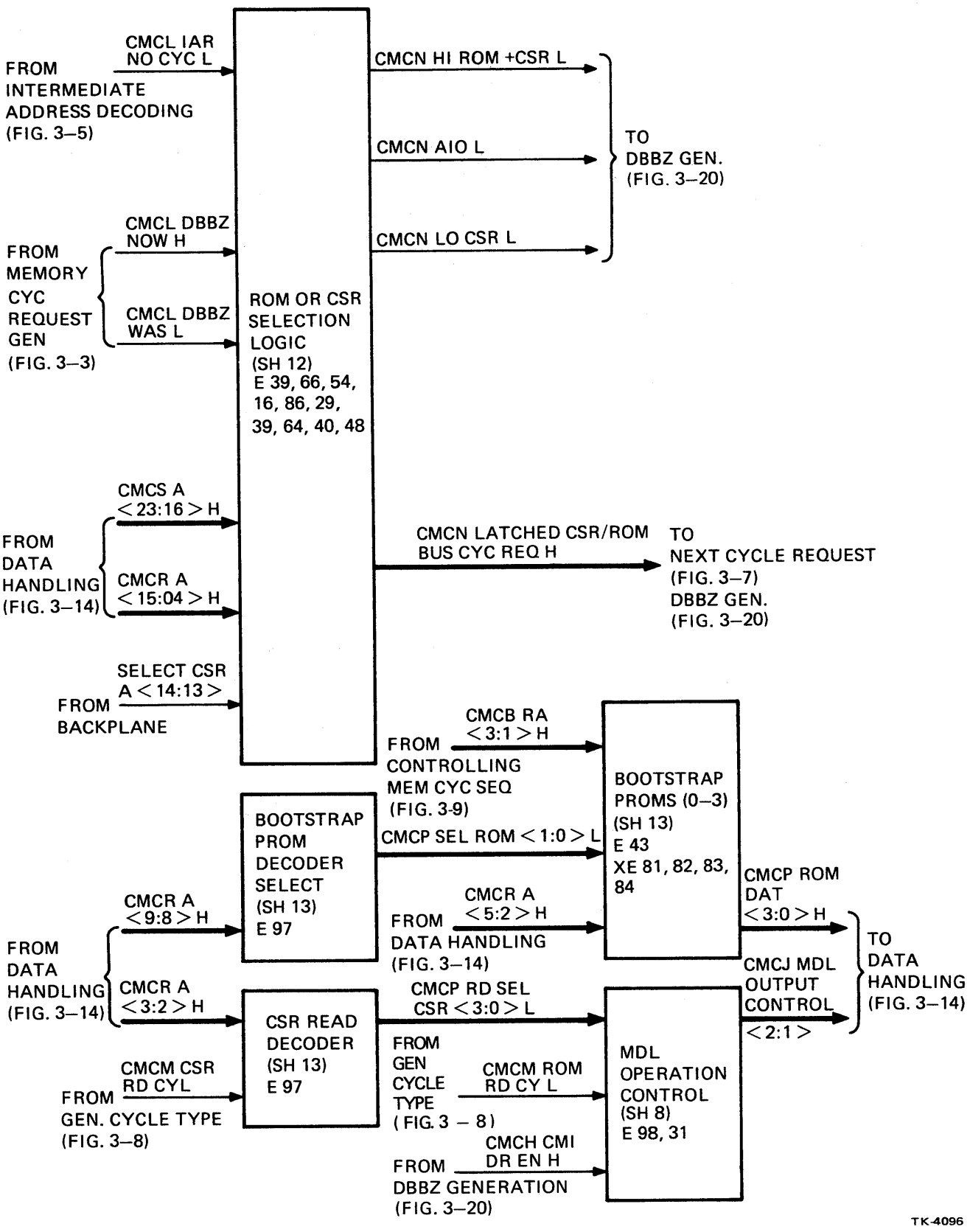

Figure 3-6 CSR and ROM Decoding

The bootstrap program data CMCP ROM DATA <3:0> H from a bootstrap PROM is applied to the data handling circuit. This bootstrap program data is assembled into 32-bit words that are transferred first to the CPU via the data handling circuit, then to the CMI bus.

3.6.3 CSR Read Decoder

The CSR read decoder circuit indicates that a CSR read cycle is to be performed when it receives CMCM CSR RD CYC L. The input signals CMCR A <3:2> H to the CSR read decoder circuit select which CSR is to be read. Meanwhile, the output signals CMCP RD SEL CSR <1:0> L of the CSR read decoder circuit and the output signal CMCM ROM RD CY L (ROM read cycle type) of the generating cycle type request circuit are applied to the MDL operation control circuit.

The two outputs (CMCJ MDL OUTPUT CONTROL <2:1>) of the MDL operation control circuit are applied to the data handling circuit to determine if a ROM read, a CSR read, or a CSR write is pending.

3.7 GENERATING NEXT CYCLE REQUEST

Seven flip-flops and one multiplexer comprise the generating next cycle request circuit. Refer to Figure 3-7.

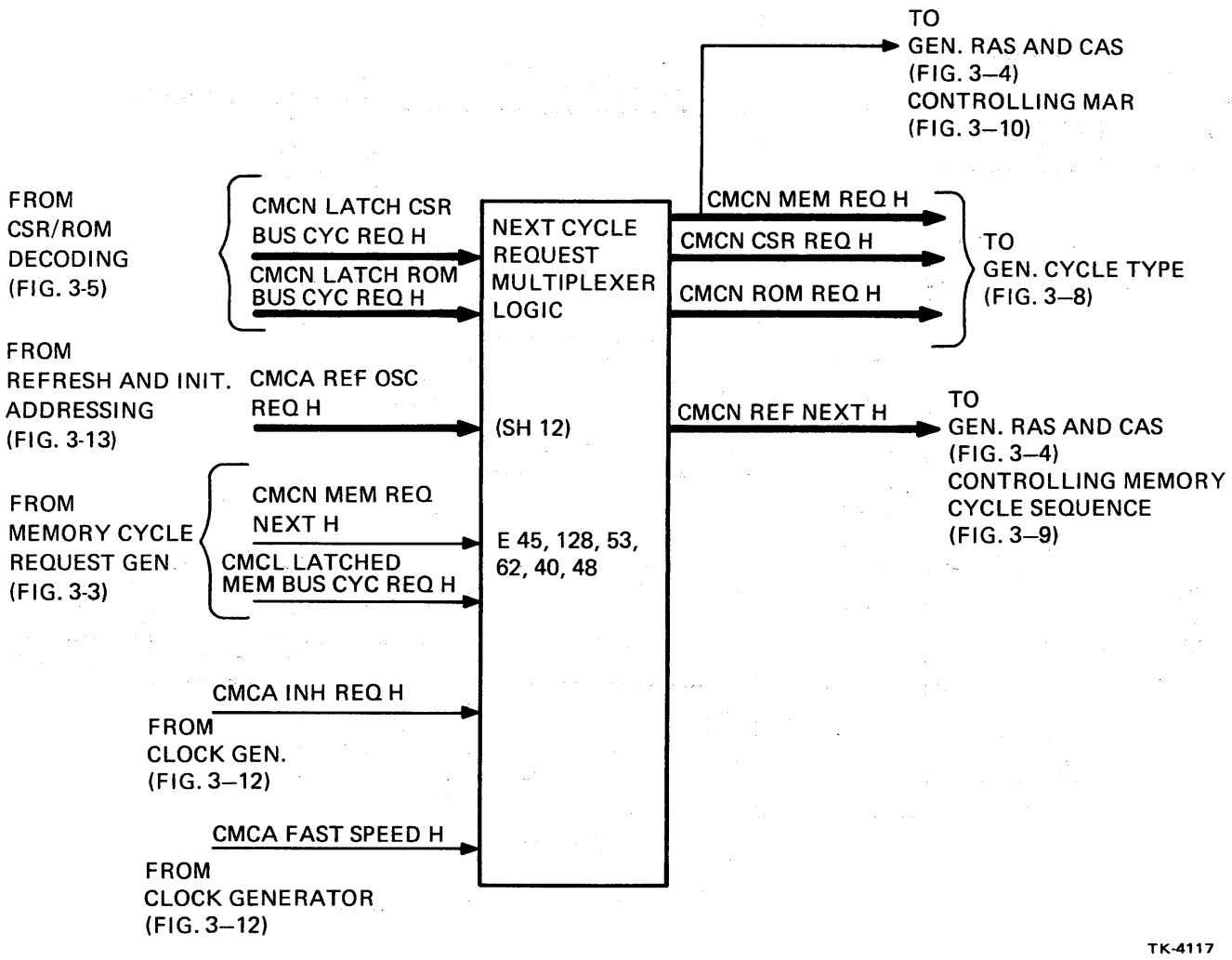

Figure 3-7 Generating Next Cycle Request

This circuit multiplexes the cycle requests so the cycle request is in synchronization with the memory controller's clock. CMCA T0 CLK H, the memory controller's clock, is derived from either the bus clock (CMI B CLK L) or the internal oscillator (CMCC OSC H). Four flip-flops synchronize the cycle requests with time T0 for slow-speed operation (internal clock), while in fast-speed mode (CMCA FAST SPEED H asserted), the cycle request is clocked with the bus clock (CMCA BUS CLK H) as it is applied to the multiplexer.

The three multiplexer's outputs CMCN MEM REQ H, CMCN CSR REQ H and CMCN ROM REQ H are applied to the generating cycle type request circuit so as to generate the ROM starting addresses for the controlling memory cycle sequence circuit. Next, the two multiplexer outputs CMCN MEM REQ H and CMCN REF NEXT H are applied to the generating RAS and CAS circuits (see Figure 3-4) so as to generate RAS and CAS for refresh cycles.

3.8 GENERATING CYCLE TYPE REQUEST

The generating cycle type request circuit consists of a refresh or cycle 0 circuit, an enable refresh circuit, a cycle request logic circuit, and a ROM cycle starting logic circuit. Refer to Figure 3-8.

This circuit generates the ROM starting address (CMCB RSA <3:0> H) for the controlling memory cycle sequence circuit. First, cycle type requests are applied to the generating cycle type request circuit so as to initiate the generation of a ROM starting address. Second, the ROM starting address (CMCB RSA <3:0> H) is applied to the controlling memory cycle sequence circuit, which generates the timing signals for the memory controller. Table 3-1 lists the ROM starting addresses.

Table 3-1 ROM Starting Addresses

ROM Starting Address				Cycle Type
A3	A2	A1	A0	
0	0	0	0	IDLE
0	0	0	1	CSR 1 WR
0	0	1	0	CSR 0 WR
0	0	1	1	CSR RD
0	1	0	0	ROM RD
0	1	0	1	ILL CSR OP
0	1	1	0	INITIALIZE
0	1	1	1	(NOT USED)
1	0	0	0	MEM 4 BYTE WR NO ECC DIS
1	0	0	1	MEM 4 BYTE WR ECC DIS
1	0	1	0	REFRESH
1	0	1	1	MEM BYTE WR
1	1	0	0	MEM RD DIAG OUT OF PG
1	1	0	1	MEM RD DIAG IN PG
1	1	1	0	MEM RD
1	1	1	1	MEM RD ECC DIS

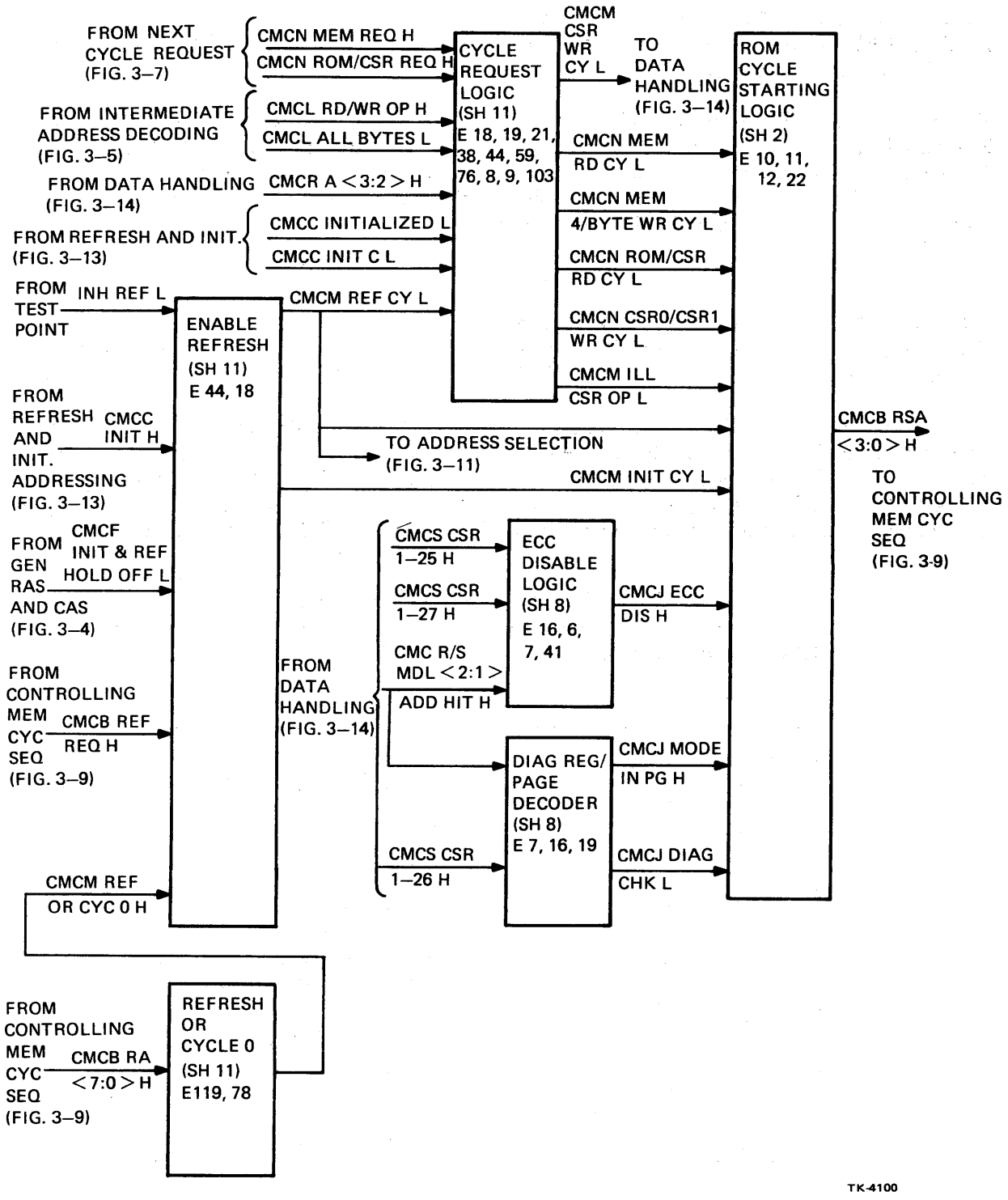

Figure 3-8 Generating Cycle Type Request

3.8.1 Refresh or Cycle 0

The controlling memory cycle sequence circuit applies the ROM address (CMCB RA <7:0> H) to the PROM of the refresh or cycle 0 circuit. The output of the PROM is then applied to flip-flops. One of the flip-flop outputs, CMCM REF OR CYC 0 H, is applied to the enable refresh circuit. This signal enables the refresh operation to have the highest priority of all the cycle operations.

3.8.2 Enable Refresh

When CMCM REF OR CYCLE H and CMCB REF REQ H are asserted, the deasserted CMCF INIT AND REF HOLD OFF L is applied to the enable refresh circuit. The status of CMCC INITIALIZE H decides which output of the enable refresh circuit is asserted.

When CMCC INITIALIZE H is asserted, the output CMCM INIT CY L is applied to the ROM cycle starting logic circuit. In contrast, when CMCC INITIALIZE L is deasserted, CMCM REF CY L is applied to the cycle request logic circuit and also to the ROM cycle starting logic circuit.

3.8.3 Cycle Request Logic

The four cycle type requests CMCN MEM REQ H, CMCN ROM REQ H, CMCN CSR REQ H and CMCM REF CY L are applied to the cycle request logic circuit so as to decide which cycle type is to be performed. To determine if a read, long-word write, or byte write function is pending, CMCL RD OP H, CMCL WR OP H, and CMCL ALL BYTES L are applied to the cycle request logic.

The address CMCR A <3:2> H from the data handling circuit is applied to the cycle request logic circuit to address either CSR0 or CSR1. The assertion of either CMCC INITIALIZE L, CMCM REF CY L or CMCC INIT C L prevents the cycle request logic from asserting a read or write cycle type to the ROM cycle starting logic. When the read-only CSR2 is illegally written to, the cycle request logic circuit asserts the signal CMCM ILL CSR OP L, which clears the CMI bus signal CMI DBBZ L.

3.8.4 ROM Cycle Starting Logic

From the cycle request logic circuit, the seven cycle types and CMCM ILL CSR OP L are applied to the ROM cycle starting logic circuit. Also, the enable refresh circuit applies CMCM REF CY L and CMCM INIT CY L to the ROM cycle starting logic circuit.

The three inputs (CMCJ DIAG CHK L, CMCJ ECC DIS H and CMCJ MODE IN PG H) that are applied to the ROM cycle starting logic circuit and affect the memory controller's diagnostic operations are derived from the diagnostic register/page decoder circuit. First, by asserting CMCJ DIAG CHK L, the memory controller substitutes check bits when data is read out of the memory. Second, the error correction logic (ECC) does not function when CMCJ ECC DIS H is asserted. Third, the memory controller's diagnostic features operate in one page of memory with CMCJ MODE IN PG H asserted.

The 4-bit ROM starting address output of the ROM cycle starting logic circuit is applied to the controlling memory cycle sequence circuit in order to produce microcode and timing for the cycles of each cycle type.

3.9 CONTROLLING MEMORY CYCLE SEQUENCE

A next address multiplexer, a lower address ROM control circuit, an upper ROM control circuit and a T1 CLK ROM address latch comprise the controlling memory cycle sequence circuit. Refer to Figure 3-9.

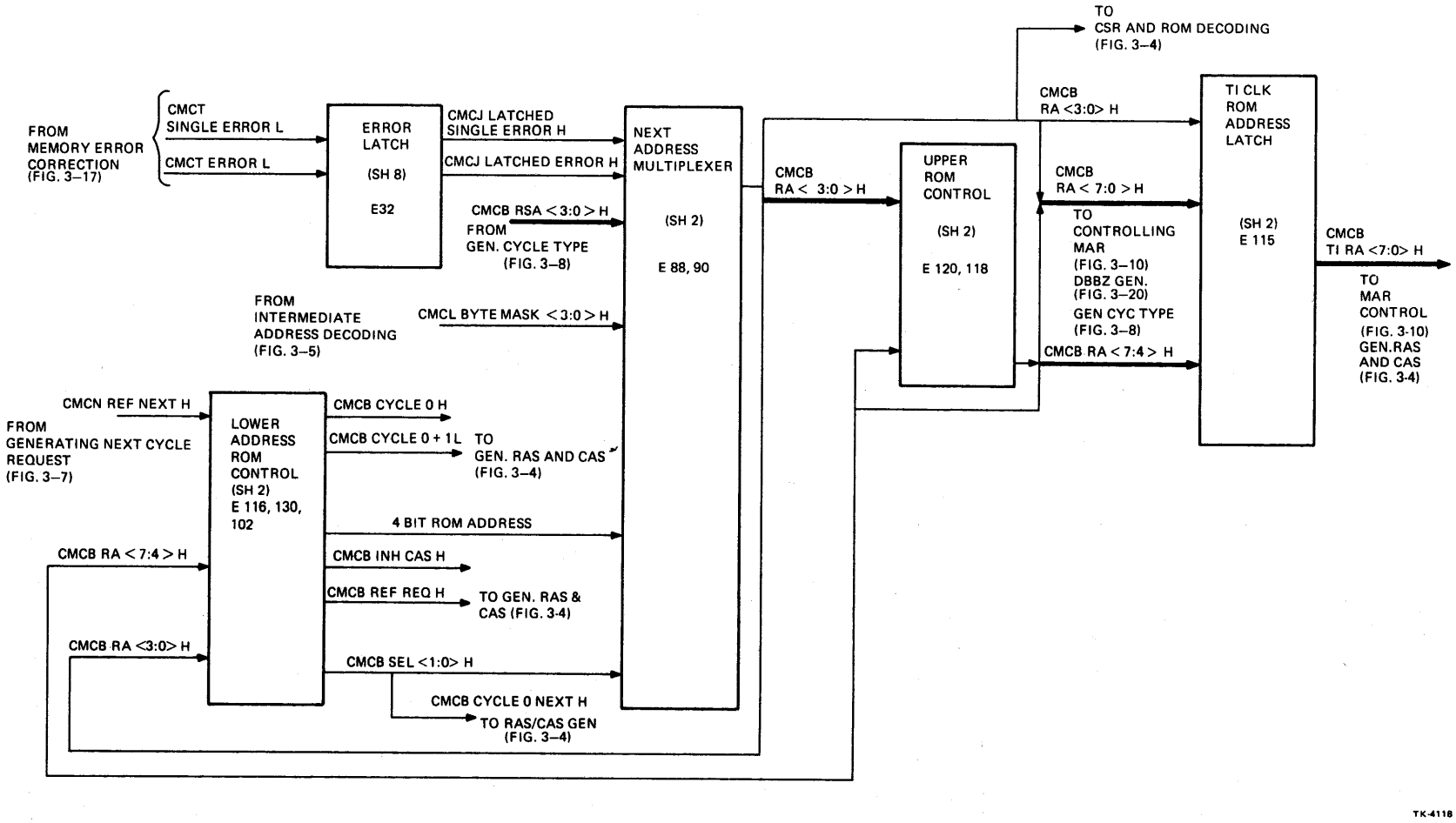

Figure 3-9 Controlling Memory Cycle Sequence

This circuit generates the timing for the cycles of each cycle type. First, each cycle type begins from an idle state. Second, each is initiated by the application of the ROM starting address CMCB RSA <3:0> H to the memory cycle sequence circuit (see Figure 3-9) from the generating cycle type request circuit (see Figure 3-8). Finally, the controlling memory cycle sequence circuit generates the timing signals CMCB RA <7:0> H and CMCB T1 RA <7:0> H, which are applied to 14 ROMs that provide control signals for the memory controller.

3.9.1 Next Address Multiplexer

Four signals (the ROM starting address, CMCB RSA <3:0> H; the byte mask, CMCL BYTE MASK <3:0> H; the 4-bit ROM address from the lower address ROM control circuit; and the error status, CMCJ LATCH ERROR H and CMCJ LATCHED SINGLE ERROR H) are applied to the next address multiplexer circuit. CMCB SEL <1:0> H then selects which of the four signals is to be multiplexed by the next address multiplexer.

In the idle state, the generating cycle type request circuit (see Figure 3-8) applies the ROM starting address CMCB RSA <3:0> H to the next address multiplexer circuit. The output of the next address multiplexer circuit ROM address CMCB RA <3:0> H is then applied to both the upper ROM control circuit and the lower address ROM control circuit. Also, the next address multiplexer circuit allows branching of the microcode for error response and byte masking.

3.9.2 Lower Address ROM Control

The ROM address CMCB RA <7:4> is applied to the lower address ROM control circuit from the upper ROM control circuit and the next address multiplexer circuit. The lower address ROM control circuit then outputs a 4-bit ROM address and a data selection (CMCB SEL <1:0> H), which are both applied to the next address multiplexer circuit.

The lower address ROM control circuit also generates the following: a refresh cycle request (CMCB REF REQ H), an inhibit column address strobe (CMCB INH CAS H), and cycle status signals. Furthermore, the cycle status signals indicate if the memory controller is in cycle 0 (CMCB CYCLE 0 H), in cycle 0 *or* cycle 1 (CMCB CYCLE 0 + 1 L), or if cycle 0 is next (CMCB CYCLE 0 NEXT H).

3.9.3 Upper ROM Control/T1 CLK ROM Address Latch

The lower ROM address CMCB RA <3:0> H from the next address multiplexer circuit and the upper ROM address CMCB RA <7:4> H from the upper ROM control circuit itself are applied to the upper ROM control circuit. At the next time T0, the upper ROM control circuit produces the timing signals CMCB RA <7:0> H, which are applied to ROMs and to the T1 ROM address latch circuit, which, at time T1, produces the timing signals CMCB T1 RA <7:0> H. In addition, CMCB RA <7:0> H and CMCB T1 RA <7:0> H are applied to 14 ROMs that produce control signals for the memory controller.

3.10 CONTROLLING MEMORY ADDRESS REGISTER

The controlling memory address register circuit consists of a control ROM circuit, a memory address register hold circuit, and a memory address register control circuit. Refer to Figure 3-10.

This circuit generates two address latching signals, CMCF MAR LATCH L and CMCF LATCH AUX MAR H. CMCF MAR LATCH L latches the memory array address into the memory address register, while CMCF LATCH AUX MAR H latches the present address into the auxiliary address register of the data handling circuit. These two latching signals are generated from the timing signals CMCB RA <7:0> H and CMCB TI RA <7:0> H, with both signals originating from the controlling memory cycle sequence circuit (see Figure 3-9).

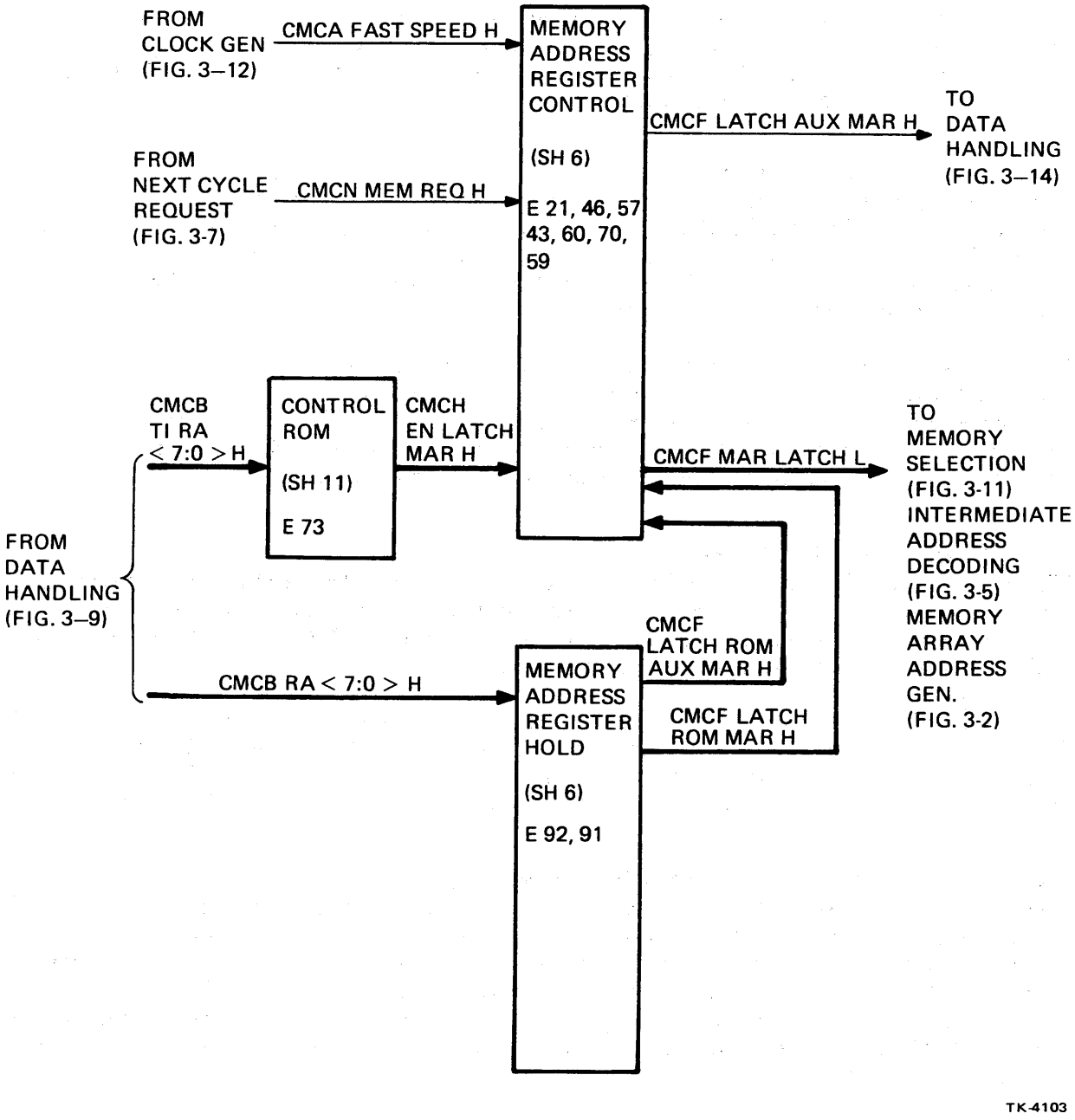

Figure 3-10 Controlling Memory Address Register

Memory Address Register Hold and Memory Address Register Control
The output CMCM EN LATCH MAR H of the control ROM circuit is applied to the memory address register control circuit. When CMCA FAST SPEED H (memory controller in fast-speed mode) and CMCN MEM REQ H (memory cycle request) are both asserted with CMCM EN LATCH MAR H, the signals CMCF MAR LATCH L and CMCF AUX MAR H are both asserted. Thus the memory address is loaded into the memory address register and auxiliary address register, respectively.

The timing signals CMCB RA <7:0> H are applied from the controlling memory cycle sequence circuit (see Figure 3-9) to the memory address register hold circuit, whose output is CMCF LATCH ROM MAR H. CMCF LATCH ROM MAR H is then applied to the memory address register control circuit. CMCF LATCH ROM MAR H is logically equivalent to CMCF MAR LATCH L. Finally, the timing signals CMCB T1 RA <7:0> H of the controlling memory address register circuit (see Figure 3-10) are applied to the control ROM circuit.

3.11 ADDRESS SELECTION

The address selection circuit is composed of an initialization or refresh circuit, a memory address register, an address multiplexer and a row/column address multiplexer. Refer to Figure 3-11.

This circuit determines which address (physical, refresh, or initialization) is to be applied to the memory arrays. First, the row address and then the column address are applied to the memory array by the address multiplexer circuit via the internal bus.

When CMCD INIT OR REF H is deasserted, the physical address CMCR/S A <15:02> H is multiplexed (a 7-bit row address followed by a 7-bit column address) and applied to the internal bus to address a memory array location for a read or a write operation. By the assertion of CMCD INIT OR REF H, the refresh address (bits <06:00>) and the initialization address (bits <13:07>) are applied to the memory arrays to perform the refresh or initialization operation.

3.11.1 Initialization or Refresh

The initialization or refresh circuit checks for an initialization cycle (CMCC INITIALIZE L asserted) or a refresh cycle (CMCM REF CY L asserted). With an asserted input (CMCC INITIALIZE L or CMCM REF CY L) to the initialization or refresh circuit, CMCD INIT OR REF H is asserted and applied to the address multiplexer circuit so as to select a refresh or initialization address. Finally, the output INT BUS REF CYC of the initialization or refresh circuit is applied to all the memory arrays to enable the four banks of RAMs on the memory arrays for a refresh or initialization cycle.

3.11.2 Memory Address Register

CMCF MAR LATCH L latches the physical address (CMCR/S A <17:02> H) from the data handling circuit into the memory address register. Next, the physical address output from the memory address register is applied to the address multiplexer circuit. The memory address register then applies bits 17 and 16 (INT BUS MA <15:14>) of the internal bus to decode the selected bank of MOS chips on the memory array.

3.11.3 Address Multiplexer

The physical address CMCR/S A <15:02> H, the refresh address CMCC REF ADD <06:00> H, and the initialization address CMCC INZ ADD <13:07> H are inputs to the address multiplexer circuit. When CMCD INIT OR REF H is asserted, CMCC REF ADD <06:00> H and CMCC INZ ADD <13:07> H are applied to the row/column address multiplexer circuit; otherwise, the physical address is applied to the row/column address multiplexer circuit.

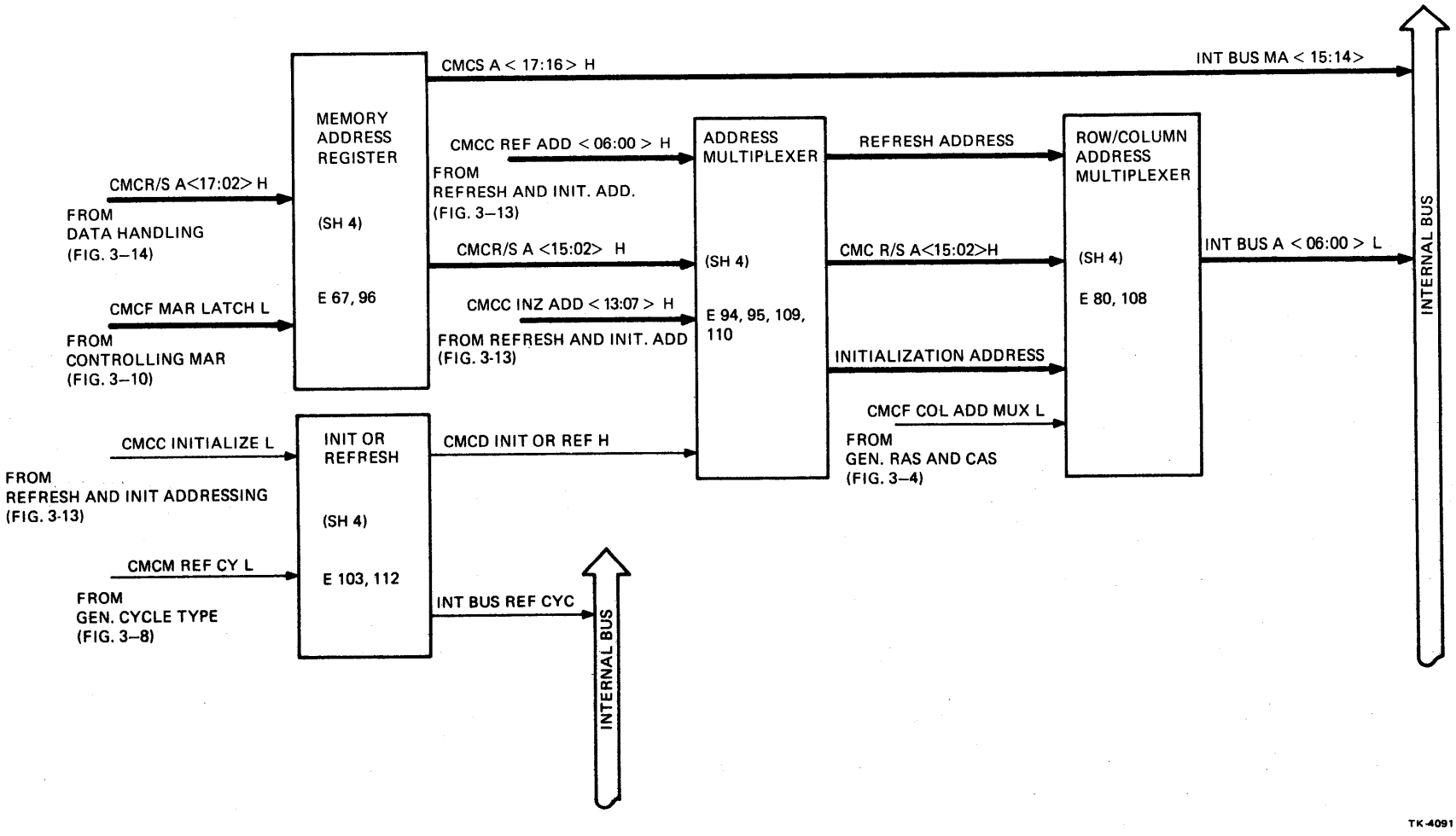

Figure 3-11 Address Selection

3.11.4 Row/Column Address Multiplexer

The row/column address multiplexer circuit multiplexes the address (either the physical address, row address CMCR A <06:00> H and then the column address CMCR/S A <13:07> H, or the refresh/initialization address, refresh address CMCC REF ADD <06:00> H and then the initialization address CMCC INZ ADD <13:7> H) and then applies the address to the internal bus. Table 3-2 explains the multiplexing scheme.

Table 3-2 Address Multiplexing

CMCD INIT OR REF H	CMCF COL ADD MUX L	INTERNAL BUS
deasserted	deasserted	CMCR A <06:00> H
deasserted	asserted	CMCR/S A <13:07> H
asserted	deasserted	CMCC REF ADD <06:00> H
asserted	asserted	CMCC INZ ADD <13:07> H

3.12 CLOCK GENERATION

The clock generation circuit consists of a missing B CLK detector, a slow-speed synchronization flip-flop circuit, an internal clock circuit, a bus clock circuit, a fast-speed synchronization flip-flop circuit and a kill refresh/inhibit cycle logic circuit. Refer to Figure 3-12.

This circuit generates CMI bus-associated timing signals (CMCA T0/T1 CLK H and CMCA BUS CLK H) for the memory controller. Normally, the clock generation circuit generates the memory controller's internal timing from the CMI bus clock (CMI B CLK L). However, when the bus clock (CMI B CLK L) is removed from the CMI bus, the clock generation circuit switches the source of the memory controller's timing from CMI B CLK L to the internal clock (delay-line oscillator). By applying the bus clock (CMI B CLK L) to the CMI bus, the clock generation circuit switches from the internal clock to the CMI bus clock (CMI B CLK L).

3.12.1 Missing B CLK Detector

Every 125 ns the retriggerable 1-μs one-shot is clocked (CMCF DEL CLK L) by the delayed bus clock. If the bus clock is not applied to the one-shot within 1 μs, the one-shot times out and produces CMCA TIMEOUT L. The asserted CMCA TIMEOUT L is then applied to the fast-speed and slow-speed synchronization flip-flop circuits. Furthermore, CMI DC LO L forces the missing B CLK detector into the timed-out state.

3.12.2 Fast-Speed Synchronization Flip-Flops

The asserted CMCA TIMEOUT L is applied to the reset gates of the two flip-flops of the fast-speed synchronization flip-flop circuit and, thus, CMCA FAST SPEED H is deasserted.

3.12.3 Slow-Speed Synchronization Flip-Flops

By applying the asserted CMCA TIMEOUT L to the slow-speed synchronization flip-flop circuit, the memory controller switches from fast-speed mode (bus clock) to slow-speed mode (internal clock). Therefore, the asserted CMCA TIMEOUT L is clocked (CMCC DEL OSC L and CMCC OSC H) into the two flip-flops of the slow-speed synchronization flip-flop circuit to synchronize CMCA SLOW SPEED H with the internal oscillator. The two clock signals (CMCC DEL OSC H and CMCC OSC H) are then derived from the internal clock (delay-line oscillator). Finally, the asserted CMCA SLOW SPEED H is applied to the internal clock circuit and the fast-speed synchronization flip-flop circuit.

Switching from slow-speed mode to fast-speed mode is initiated by applying CMI B CLK L to the missing B clock detector circuit, which causes the output CMCA TIMEOUT L to be deasserted. CMCA TIMEOUT L is then applied to the kill refresh/inhibit cycle logic circuit.

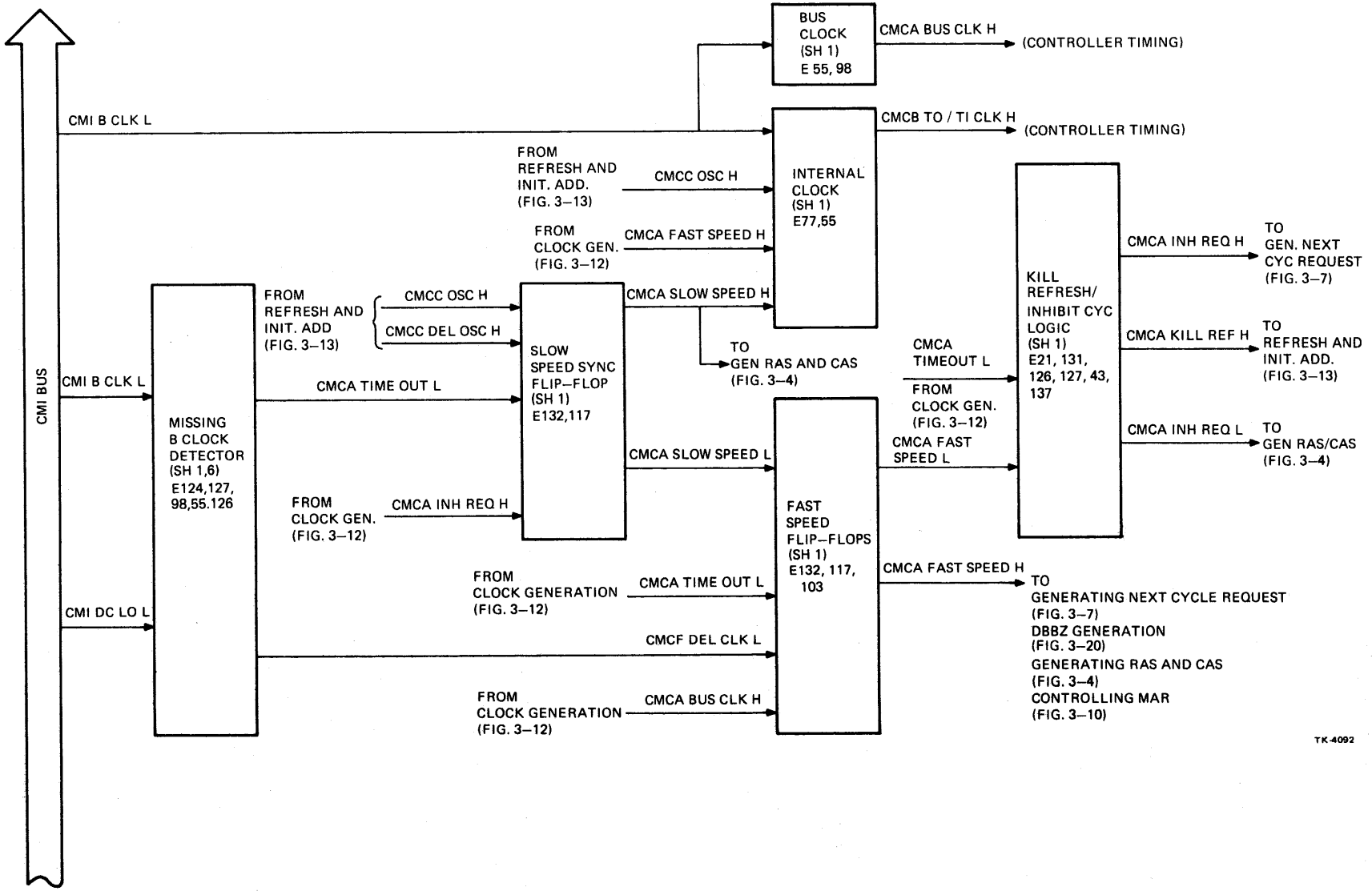

Figure 3-12 Clock Generation

3.12.4 Internal Clock/Bus Clock

In slow-speed mode, CMCA SLOW SPEED H and the delay-line oscillator's output CMCA OSC H are applied to the internal clock circuit that produces CMCA T0/T1 CLK H. However, in fast-speed mode, CMCA FAST SPEED H and CMCA BUS CLK H are applied to the internal clock circuit to produce CMCA T0/T1 CLK H. Finally, the timing signal CMCA BUS CLK H is produced by the application of CMI B CLK L to the bus clock circuit.

3.12.5 Kill Refresh/Inhibit Cycle Logic

A deasserted CMCA TIMEOUT L signal indicates the CMI bus clock is present, while at this time, the kill refresh/inhibit cycle logic verifies the memory controller is not in fast-speed mode (CMCA FAST SPEED L deasserted). When the conditions above are met, the kill refresh/inhibit cycle logic circuit asserts CMCA KILL REF L. Also, CMCA KILL REF L is applied to the refresh and initialization circuit to prevent refresh cycles while the memory controller is switching from slow-speed to fast-speed. Also, CMCA KILL REF H is clocked (CMCA T0 CLK H) into a flip-flop when CMCB CYCLE 0 H (memory controller is in cycle 0) is asserted. CMCA INH REQ H is then produced.

First, the asserted CMCA INH REQ H is applied to the generating next cycle request circuit (see Figure 3-7) to prevent a cycle type request during the slow-to-fast-speed clock exchange. Second, CMCA INH REQ H is applied to the slow-speed synchronization flip-flop circuit to deassert CMCA SLOW SPEED H. Furthermore, CMCA SLOW SPEED L is clocked (CMCF DEL CLK L and CMCA BUS CLK H) into the fast-speed synchronization flip-flop circuit, which then asserts CMCA FAST SPEED H.

3.13 REFRESH AND INITIALIZE ADDRESSING

A delay-line oscillator, a refresh frequency counter, a refresh request flip-flop, a refresh request circuit and address counter comprise the fundamental blocks of the refresh and initialize addressing circuit. Refer to Figure 3-13.

The delay-line oscillator in this circuit provides the timing for the refresh cycle and also substitutes for CMI B CLK L when either the memory controller is on battery backup or the VAX-11/750 is in single-step mode. Every 14 μs the refresh and initialize addressing circuit generates a refresh cycle request so that 1 of the 128 rows of the MOS memory is refreshed. All 128 rows of the dynamic MOS memory arrays must be refreshed within an interval of 2 μs to prevent deterioration of the potential charge of the MOS memory.

3.13.1 Delay-Line Oscillator

Every 220 ns the delay-line oscillator circuit produces CMCC OSC H and CMCC DEL OSC L. Both CMCC OSC H and CMCC DEL OSC L are applied to the clock generation circuit (see Figure 3-12) to provide the timing for slow-speed mode. CMCC OSC H is the clock input to the refresh counter circuit.

3.13.2 Refresh Frequency Counter/Refresh Request Flip-Flop

CMCC OSC H increments the 6-staged refresh frequency counter whose output is clocked by CMCC DEL OSC L into a flip-flop to produce CMCC REF CLK H. CMCC REF CLK H clocks the deasserted CMCA KILL REF L into the refresh flip-flop to produce CMCA REF OSC REQ H.

3.13.3 Refresh Request/Address Counter

CMCA REF OSC REQ H, applied to both the refresh request circuit and the address counter, is clocked by CMCA T0 CLK H into a flip-flop of the refresh request circuit. The output of the refresh request flip-flop is CMCA REF OSC REQ H, which is applied to the generating next cycle request logic circuit to enable a refresh cycle request.

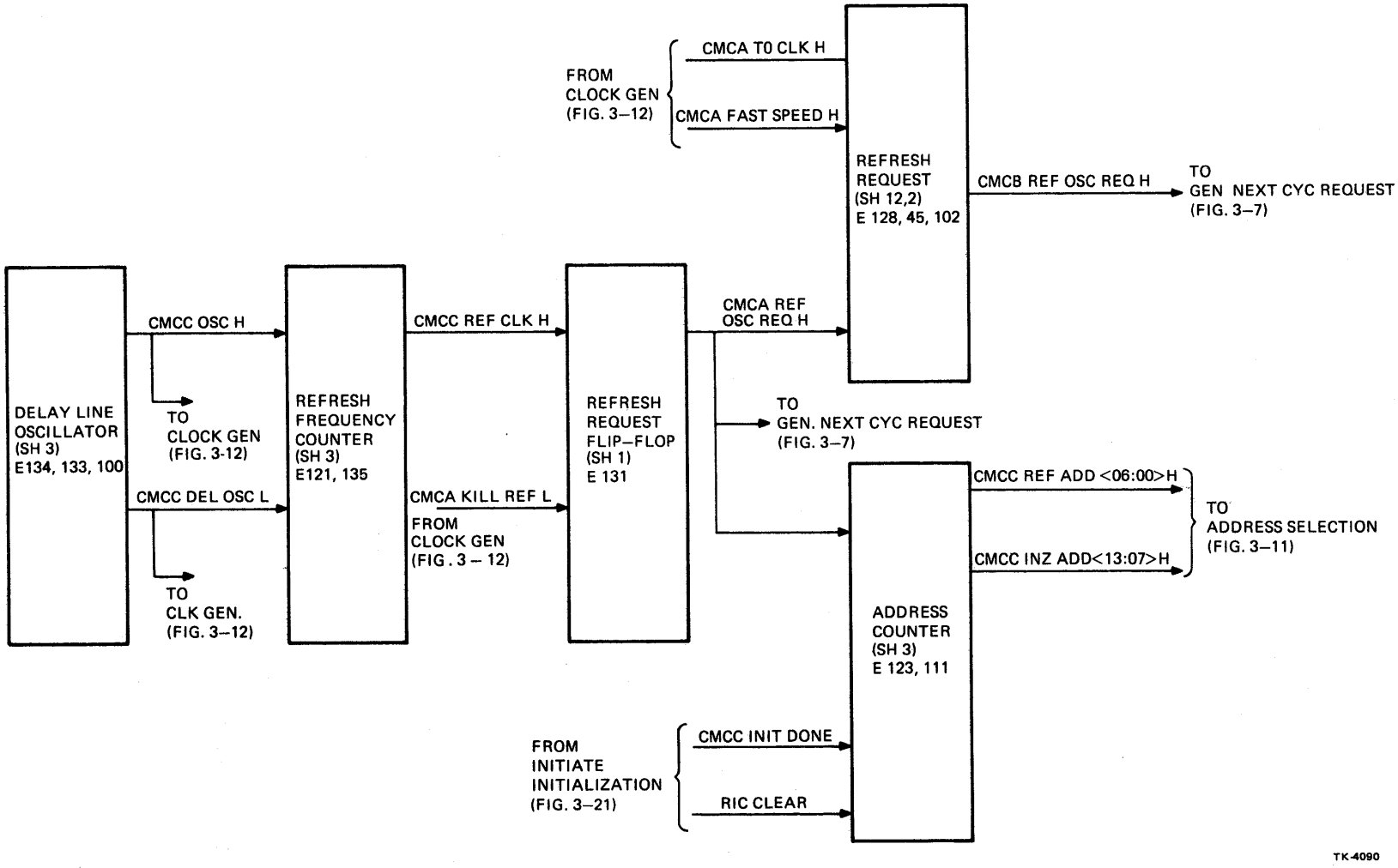

Figure 3-13 Refresh and Initialize Addressing

The address of the address counter (CMCC REF ADD <06:00> H and CMCC INZ ADD <13:07> H) is incremented at the end of each refresh cycle with each deassertion of CMCA REF OSC REQ H. The refresh and initialization address is then applied to the address selection circuit (see Figure 3-11), which multiplexes the address and also applies it to the internal bus.

3.14 DATA HANDLING

The data handling circuit consists of a memory data register, an intermediate address register, an auxiliary register, a CSR input data multiplexer, a CSR0 and a CSR1 register, a CSR0 and a CSR1 comparator, a bus output multiplexer, a bootstrap ROM word assembly register, and a CMI output multiplexer. Figure 3-14 is a composite drawing of the data handling circuit's four memory data loop gate arrays.

This circuit, which is composed of four identical memory data loop gate arrays, transfers the address and data between the CMI bus and the MOS memory arrays. It also contains two control/status registers (CSRs) and two comparators. However, each memory data loop gate array is capable of transferring one byte of data; thus four gate arrays are used to transfer the 32-bit CMI data word.

3.14.1 Memory Data Register

During a write transaction, CMCE LATCH MDR H latches a 32-bit data word from the CMI bus into the memory data register (MDR). The MDR's output is applied to the memory error correction circuit and then to the memory arrays via the internal bus. CMCE <3:0> DR EN H enable the tri-state bus drivers that drive the MDR's contents onto the internal bus.

3.14.2 Intermediate Address Register

First, CMCF LATCH IAR H latches the CMI memory address into the intermediate address register (IAR) until the memory system is not busy. The memory controller is then able to use the address data. The IAR's contents are applied to the ABus output multiplexer, which applies the address to the memory arrays. In addition, the IAR's contents are latched by CMCF LATCH AUX MAR H into the auxiliary address register, which applies the bits <23:08> to CSR0 for error logging.

3.14.3 CSR Input Data Multiplexer

The CSR input data multiplexer selects one of three input sources to be stored in CSR0 and CSR1: bytes 0 and 3 (INT BUS DB <31:24> <6:0> RD L) from the internal bus, all four bytes (CMI <31:00> H) from the CMI bus, or bytes 1 and 2 from the auxiliary address register. Hence, the selected input to the CSR input data multiplexer is determined by IDENT <2:1> and CMCM L CSR WR CY L.

3.14.4 CSR0 and CSR1 Registers

CMCJ MDL OUTPUT CONTROL <2:1> determines the selection of either CSR0, CSR1, the bootstrap ROMs, or the internal bus. However, the CSR input data multiplexer's output is latched into CSR0 or CSR1 by CMCE MDL <3:0> LATCH REG <2:1> H. In contrast, the assertion of CMCC INIT F L clears both CSR0 and CSR1. The contents of CSR0 and CSR1 are applied to the ABus output multiplexer, the CMI output multiplexer, and the CSR0 and CSR1 comparators.

CSR0 contains 32 bits of error data that are reported by the memory controller to the CPU. The error status flags in byte 3 are cleared by the CPU's writing a 1 to CSR0 <31:29>. When the CPU writes a 0 to CSR0 <31:29>, the state of the bits do not change. Furthermore, the RDS error log request (CSR0 bit 31) can be cleared only if the RDS high error rate (CSR0 bit 30) is not set, or if both bits are being cleared during the same write operations.

Figure 3-15 shows the CSR0 bit allocations and Table 3-3 explains them.

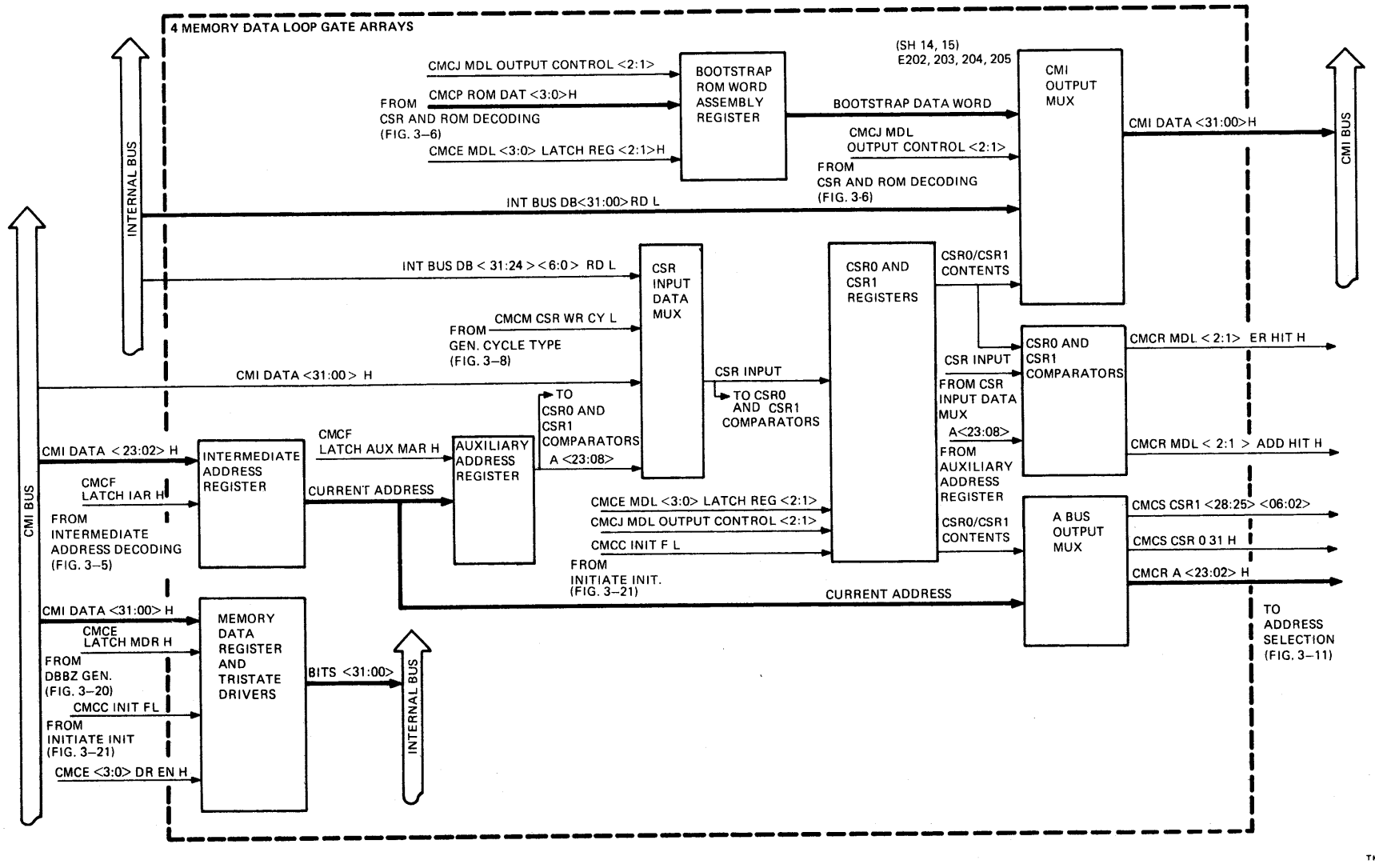

Figure 3-14 Data Handling

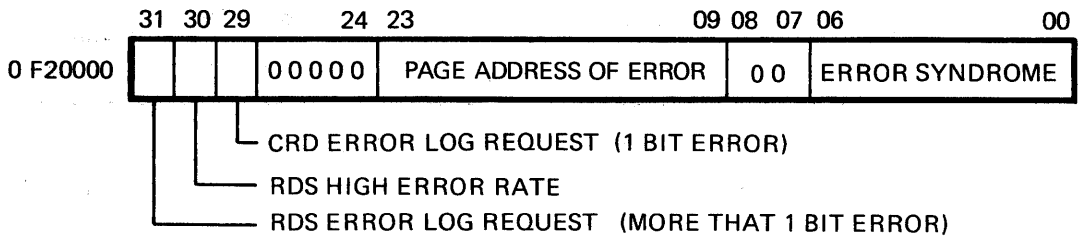

Figure 3-15 CSR0 Bit Allocations

Table 3-3 CSR0 Bit Allocations

Bit Allocation	Definition
CSR0 <06:00>	The error syndrome is stored in CSR0 <06:00>.
CSR0 <08:07>	Not used.
CSR0 <23:09>	Identifies the page (512 bytes) where an error occurred during a memory read cycle. This page address corresponds to the last error that occurs, with the exception that the page address of an uncorrectable error* overwrites the page address of a correctable error. The page address of an uncorrectable error is not overwritten by any other error until CSR0 bits <31:30> are cleared.
CSR0 <24:28>	Not used.
CSR0 bit 29	CRD Error Log Request This bit is set to 1 when a correctable (single-bit) error occurs during a READ operation. A single-bit error that occurs during the read portion of a BYTE WRITE cycle has no effect on CSR0 bit 29. This bit is cleared when set to 1.
CSR0 bit 30	RDS High Error Rate When CSR0 bit 30 is a 1, an uncorrectable error has occurred while the RDS error log request (CSR0 bit 31) was set. The second uncorrectable error page address does not overwrite the first uncorrectable error page address. This bit is cleared when set to 1.
CSR0 bit 31	RDS Error Log Request The occurrence of an uncorrectable error sets CSR0 bit 31 to 1. The address and syndrome of an uncorrectable error overwrites the address and syndrome of a correctable error. This bit is cleared when set to 1.

*An uncorrectable error is a word that contains two bits in error.

CSR1 contains transmitted control information from the memory controller. The diagnostic mode check bits (byte 0) are written by the memory controller via the internal bus. CSR1 bits <28:25> <23:09> <06:00> are set with a 1 and cleared with a 0.

For test purposes, any bit in CSR1 or CSR2 (except CSR0 <31:29>) can be written to a 1 or 0 by the CPU.

Figure 3-16 shows the CSR1 bit allocations and Table 3-4 explains them.

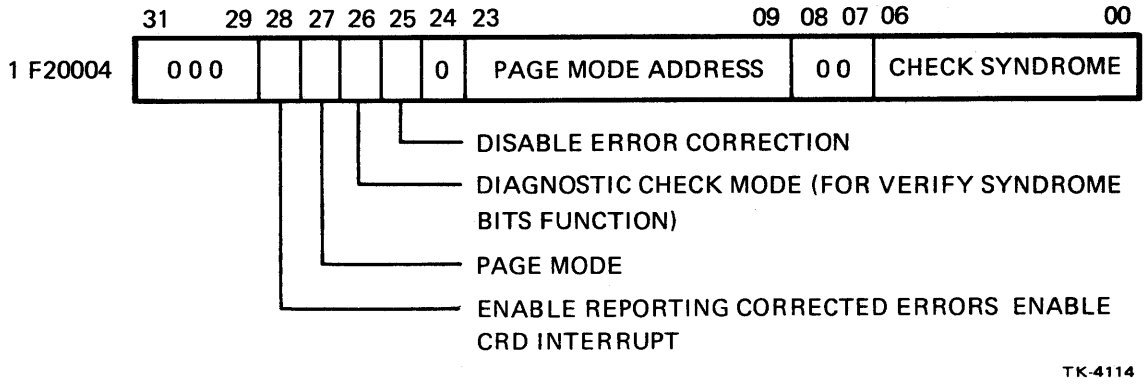

Figure 3-16 CSR1 Bit Allocations

Table 3-4 CSR1 Bit Allocations

Bit Allocation	Definition
CSR1 <06:00>	Check Bits In diagnostic check mode, the contents of CSR1 <06:00> are substituted for the check bits that come from the memory during a read operation. In ECC disable mode, a read operation stores the seven check bits read from the memory array into CSR1 <06:00>. In ECC disable mode, a write operation stores the seven check bits read from the memory array into CSR1 <06:00>.
CSR1 <08:07>	Not used.
CSR1 <23:09>	Page Mode Address If the memory system is in page mode (CSR1 bit 27 is a 1), CSR1 <23:09> contain the address of the 512-byte page to which the diagnostic check mode or the ECC disable mode applies.
CSR1 bit 24	Not used.
CSR1 bit 25	ECC Disable Mode When CSR1 bit 25 is a 1, uncorrectable errors are not detected and single-bit errors are not corrected. Also, error logging in CSR0 does not occur and NO ERROR is reported on the status lines. In ECC disable mode (CSR1 bit 25 is a 1), a read operation stores the seven check bits read from the memory array in CSR1 <06:00>. During a 4-byte write operation, the generated check bits that are written into the memory array are also stored in CSR1 <06:00>. The ECC can be disabled for a 512-byte page or for the entire memory, depending on whether page mode (CSR1 bit 27) is selected. Correct check bits are always stored in the memory array during a write cycle. ECC disable mode and diagnostic check mode cannot be selected at the same time.

Table 3-4 CSR1 Bit Allocations (Cont)

Bit Allocation	Definition
CSR1 bit 26	Diagnostic Check Mode When CSR1 bit 26 is a 1, the contents of CSR1 <06:00> are substituted for the check bits that are read from the memory during a read operation. Diagnostic check mode is constrained to operate on a single-page whose address is stored in CSR1 <23:09>. While operating in diagnostic check mode, read errors that occur in other pages of memory are not logged into CSR0. Diagnostic check mode operates in page mode only.
CSR1 bit 27	Page Mode When CSR1 bit 27 is a 1, ECC disable mode operates on a 512-byte page whose address is contained in CSR1 <23:09>. ECC disable mode operates on a page or the entire memory, depending on whether page mode is selected.
CSR1 bit 28	Enable CRD Error Status When CSR1 bit 28 is a 0, correctable (single-bit) errors are corrected by the ECC logic, but the status lines report no error. The page address of the correctable error is not logged in CSR0 <23:09> and the CRD error log request (CSR0 bit 29) is not set. CSR1 bit 28 has no effect on the logging and reporting of uncorrectable errors.
CSR1 <31:29>	Not used.

3.14.5 CSR0 and CSR1 Comparators

Two kinds of comparisons, page mode address and error address, are performed by the CSR0 and CSR1 comparators.

Page mode address is contained in bytes 1 and 2. Therefore, bytes 3, 0, and bit 0 of byte 1 (bit 0 is not a valid address bit) are disabled in the CSR1 comparator. Furthermore, CSR1 comparator's contents are compared with the current memory operation address stored in the auxiliary address register (AAR). When the contents of CSR1 and AAR are identical, CMCR/S MDL 1 ADD HIT H is asserted.

The error address comparator functions exactly as the page mode comparator except that the syndromes contained in byte 0 are compared (via the CSR input data multiplexer) with syndromes from the memory error correction circuit for the current memory read. However, when the contents of CSR0 and the current memory operation address are identical, CMCR MDL 1 ER HIT H is asserted.

3.14.6 ABus Output Multiplexer

IDENT <2:1> controls the ABus output multiplexer's outputs. The ABus output multiplexer transmits the address (bytes 0, 1 and 2) to the CSR and ROM decoding circuit (see Figure 3-6) for the memory controller to recognize ROM and CSR accesses. Bytes 0 and 1 from the ABus multiplexer are also used to select MOS RAM row and column addresses. In addition, the ABus output multiplexer transmits information (byte 3) from the CSRs to the memory controller. This information (byte 3) is used for control decisions and consists of three error status flags in CSR0 and four mode control bits in CSR1.

3.14.7 Bootstrap ROM Word Assembly Register

The bootstrap ROM word assembly register assembles 4-bit data words (CMCP ROM DATA <3:0> H) from a bootstrap PROM into a single 32-bit word for transmission to the CPU.

CMCJ MDL OUTPUT CONTROL <2:1> enables the bootstrap ROM word assembly register for data transfer. Therefore, each memory data loop gate array contains two 4-bit registers with separate latch control lines. As the ROM is sequenced through eight successive addresses, one of the eight 4-bit ROM registers is actuated by CMCE MDL 1 LATCH REG <2:1> H, which stores four more bits until the whole 32-bit word is assembled.

3.14.8 CMI Output Data Multiplexer

The CMI output data multiplexer selects data from one of four sources for transfer onto the CMI bus. These four sources are the internal bus (INT BUS <31:00> RD L) during a memory array read transaction, the bootstrap ROM word assembly register, CSR0 and CSR1. CMCJ MDL <3:0> OUTPUT CONTROL <2:1> selects and enables the source data onto the CMI bus.

3.15 MEMORY ERROR CORRECTION (MEC)

The memory error correction circuit consists of two memory error correction gate arrays. One of the MECs (low-word) uses bytes 0 and 1 while the other (high-word) uses bytes 2 and 3. The fundamental blocks of the MEC are an input data register, a check bit register, a check bits/syndrome generator, a syndrome decode circuit, a correction network and an output data register. Refer to Figure 3-17.

This circuit detects all single- or double-bit errors and corrects single-bit errors on a 32-bit data field. During a READ the 32-bit data field read out is checked for errors, with single-bit errors corrected. During a WRITE the memory error correction circuit generates the correct check bits for the 32-bit data field that is written into memory. The seven check bits or syndrome bits are generated according to the Modified Haming Code.

Use Figure 3-18 to determine which bit is in error. (All syndrome bits are 0 when there is no error.) An "X" represents a 1. As an example, if syndromes 8, 16 and T are 1s, bit 0 is in error because only bit 0 uses syndromes 8, 16 and T. For another example, bit 1 in error would be represented by syndromes 1, 8 and 16 being all 1s. When only one syndrome is a 1, a check bit error is indicated. If only syndrome 2 equals 1, C2 is in error.

3.15.1 Memory READ

During a READ the memory error correction circuit is initiated by the reading of 32 data bits and 7 check bits from a memory location. The 32-bit data field is latched by CMCE MEC LATCH DATA IN H into the low-word MEC input data register. Using the check bits of bytes 0, 1 and 7, the low-word check bits/syndrome generator produces a 7-bit partial syndrome.

The partial syndromes are applied to the high-word check bits/syndrome generator. Meanwhile, CMCE MEC LATCH DATA IN H latches bytes 2 and 3 into the high-word input data register, whose output is also applied to the check bits/syndrome generator. When the high-word check bits/syndrome generator's output is 0, there is no error in the four bytes of the data field. Consequently, with 0 syndromes, the correct data (bytes 0, 1, 2 and 3) is applied to the internal bus by the assertion of CMCE MEC LATCH OUTPUT L.

Detected syndromes indicate an error in the 32-bit data field. With an error, one or more syndrome bits are asserted; thus, the high-word syndrome decode circuit asserts CMCT ERROR L. Upon the occurrence of a single-bit error, an odd number of syndrome bits and CMCT SINGLE ERROR L are asserted. When an uncorrectable error is detected, the 32-bit data field and seven check bits are rewritten into memory.

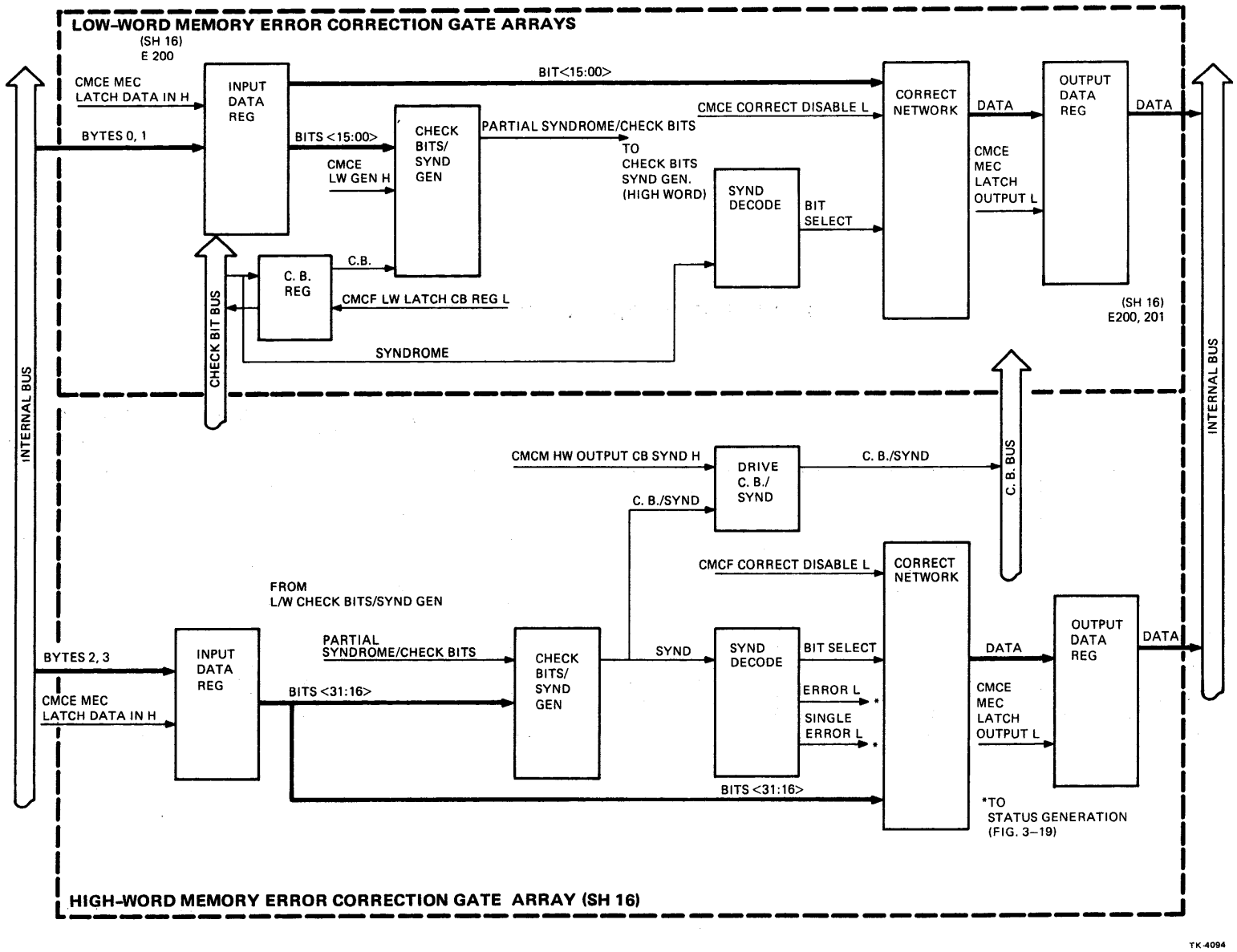

Figure 3-17 Memory Error Correction

	BYTE 0								BYTE 1								BYTE 2								BYTE 3													
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	C1	C2	C4	C8	C16	C32	CT
1		X		X		X		X		X		X		X		X		X		X		X		X		X		X		X		X						
2			X	X			X	X			X	X			X	X			X	X			X	X			X	X			X	X		X				
4					X	X	X	X					X	X	X	X					X	X	X	X					X	X	X	X			X			
8	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X									X	X	X	X	X	X	X	X				X		
16	X	X	X	X	X	X	X	X									X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X					X	
32									X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X						X	
T	X			X		X	X		X			X		X	X		X			X		X	X			X	X		X			X						X

EVEN PARITY IS GENERATED ON 1, 2, AND T.
ODD PARITY IS GENERATED ON 4, 8, 16, AND 32.

TK-4102

Figure 3-18 Modified Hamming Code

Since the syndromes indicate which of the 32 bits of the data word is in error, the high-word syndromes are applied to the low-word syndrome decode circuit via the check bit bus. Bytes 0, 1 and the syndromes are applied to the low-word correction network circuit, while bytes 2, 3 and the syndromes are applied to the high-word correction network circuit. Thus, the correction network circuit can correct the error bit. Finally, the corrected data word is latched by CMCE LW OUTPUT BYTE <1:0> H and CMCE HW OUTPUT BYTE <1:0> H into the low- and high-word output data registers respectively, and the corrected 32-bit data word is applied to the internal bus.

3.15.2 4-Byte WRITE

Bytes 0 and 1 are latched by CMCE MEC LATCH DATA IN H into the low-word input data register while bytes 2 and 3 are latched by CMCE MEC LATCH DATA IN H into the high-word input data register. Next, the check bit register is cleared to 0s with the assertion of CMCE LW GEN H. Since there are no check bits input, the low-word check bits/syndrome generator provides the partial check bits for bytes 0 and 1. At this point the generated low-word check bits are applied to the high-word check bits/syndrome generator.

By using the seven partial check bits and bytes 2 and 3, the high-word check bits/syndrome generator provides the seven check bits for the entire 32-bit data word. The 32-bit data word is then stored in the low-word (bytes 0 and 1) and high-word (bytes 2 and 3) output data registers. Finally, the 32-bit data word and the seven newly generated check bits are applied to the internal bus to be stored in memory.

3.15.3 Byte WRITE

A data word is read out of the memory location that the new data word is to be written into. All four bytes of the data word read out are stored in the low-word and high-word input data registers. The CPU then applies the new byte(s) that is (are) to be written into memory to the low- and high-word MECs via the CMI bus. Which new bytes are to be written into memory is decided by the MECs' inputs CMCE LW OUTPUT BYTE <1:0> H and CMCE HW OUTPUT BYTE <1:0> H, which, in turn, are determined by the byte mask.

The low-word MEC check bit register is cleared to all 0s by CMCE LW GEN H. First, the low-word check bits/syndrome generator generates the partial check bits for bytes 0 and 1. The low-word partial check bits are then applied to the high-word check bits/syndrome generator. By using bytes 2, 3 and the low-word partial check bits, the high-word check bits/syndrome generator produces the check bits for the entire 32-bit data word. These seven check bits are latched into the low-word check bit register via the check bit bus. Under the control of CMCE MEC LATCH OUTPUT L and CMCF LW/HW OUTPUT CB BUS H, the 32-bit word and the seven check bits are written into memory.

3.16 STATUS GENERATION

The status generation circuit consists of a perform cycle type circuit, a memory read done circuit, a slow read done circuit, a status logic circuit, an error latch and a ECC disable logic circuit. Refer to Figure 3-19.

This circuit provides the memory controller's status to the CPU via the CMI bus. The CMI STATUS information is applied to the CPU when CMI DBBZ L is deasserted by the memory controller. The CMI STATUS provides no error, single-bit error, or double-bit error status to the CPU.

3.16.1 Perform Cycle Type

The perform cycle type circuit determines if there is a memory, CSR or ROM cycle type. In addition, it checks if: (1) CMI DBBZ L is asserted in the present bus cycle but was not asserted the previous bus cycle, (2) the addressed memory location is present, (3) the cycle type is legitimate, and (4) there is a ROM or CSR cycle type. When a cycle type is being performed, CMCH DR STATUS H is applied to both the slow read done circuit and the status logic circuit.

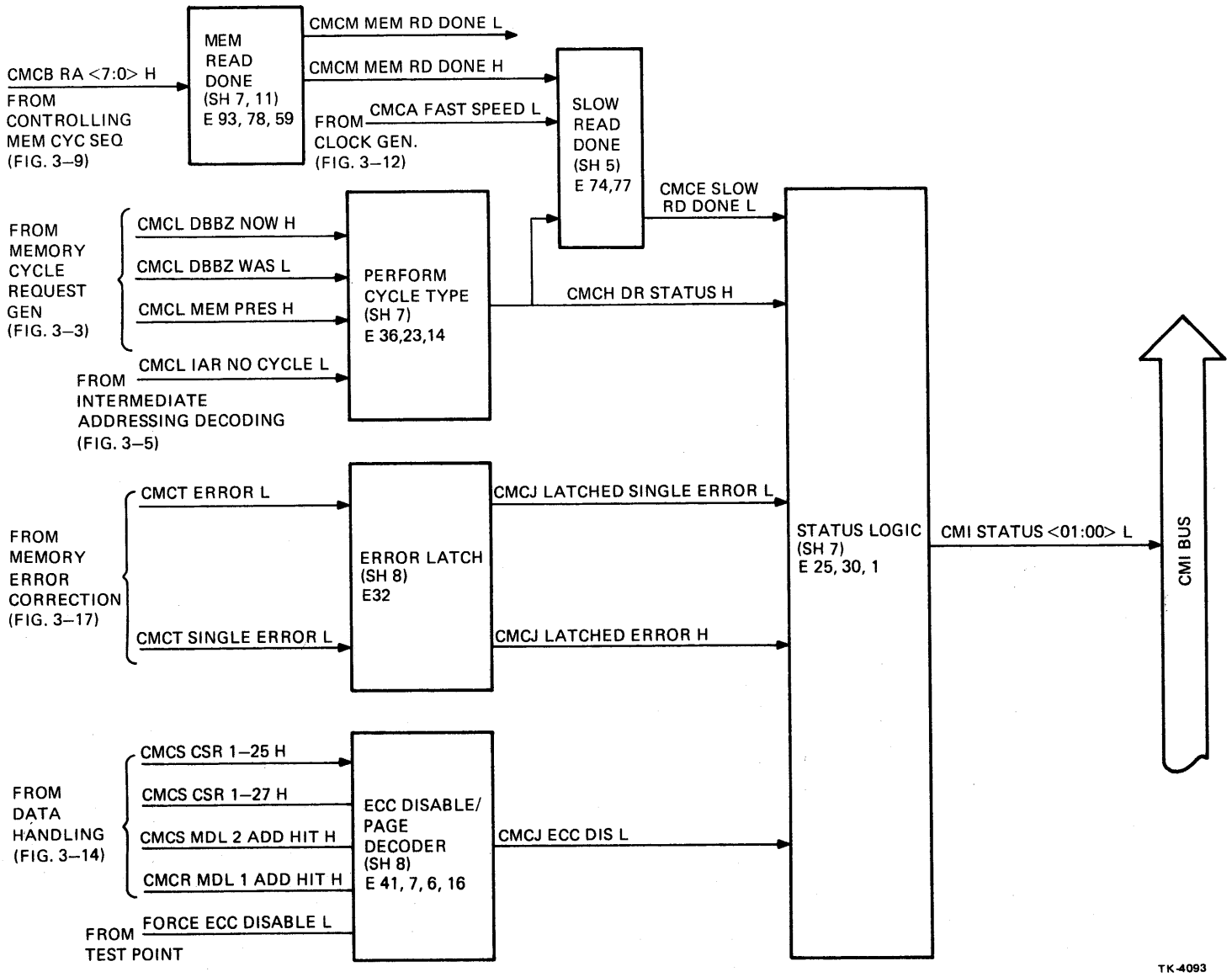

Figure 3-19 Status Generation

3.16.2 Slow Read Done/Memory Read Done

The slow read done circuit is used during slow-speed mode. This circuit verifies that the memory controller is in slow-speed mode (CMCA FAST SPEED L deasserted) with CMCH DR STATUS H asserted, and then applies CMCE SLOW RD DONE L to the status logic circuit. CMCE SLOW RD DONE L is similar to CMCM MEM RD DONE L and is initiated by the control timing signals CMCB RA <7:0> H, which are derived from the controlling memory cycle sequence circuit (see Figure 3-9) and are synchronous to the internal oscillator. Finally, CMCE SHOW RD DONE L is terminated by the deassertion of CMCH DR STATUS H, which occurs synchronously with the CMI B CLK L.

3.16.3 Error Latch

The memory error correction circuit (see Figure 3-17) applies error information to the error latch circuit, which, in turn, applies CMCJ LATCH ERROR H (two error bits) and CMCJ LATCHED SINGLE ERROR L (a single-bit error) to the status logic circuit.

3.16.4 ECC Disable Logic

The ECC disable logic determines if: (1) the error correction circuit (see Figure 3-17) is disabled (CMCS CSR1-25 H asserted), (2) the memory controller is in page mode (CMCS CSR1-27 H asserted), and (3) the page mode address equals the present operational address (CMCS/R MDL <2:1> ADD HIT H asserted). When the memory controller is in ECC disable mode, CMCJ ECC DIS L is asserted and applied to the status logic circuit. This prevents error status information from being transferred to the CPU.

3.16.5 Status Logic

The status logic circuit uses inputs from the perform cycle type circuit (CMCH DR STATUS H), the error latch circuit (CMCJ LATCHED SINGLE ERROR L and CMCJ LATCHED ERROR H), the ECC disable logic circuit (CMCJ ECC DIS L), the memory read done circuit (CMCM MEM RD DONE L) and the slow read done circuit (CMCE SLOW RD DONE L) to determine the timing for the STATUS to be properly applied and then removed from the CMI bus.

Table 3-5 lists the CMI status code.

Table 3-5 CMI Status Code

CMI Status	<01 : 00> L	
No error	L	L
No response	H	H
Single-bit error	L	H
Double-bit error	H	L

L = 1

3.17 DBBZ GENERATION

A ROM control circuit, DBBZ error detection flip-flops, a queued fast-speed write circuit, DBBZ flip-flops, a cycle start circuit and a slow-speed cycle type circuit are the fundamental blocks of the DBBZ generation circuit. Refer to Figure 3-20.

This circuit determines when the memory controller applies CMI DBBZ L to the CMI bus. Moreover, CMI DBBZ L indicates the availability of the CMI bus because the assertion of CMI DBBZ L holds the CMI bus until data transactions are complete.

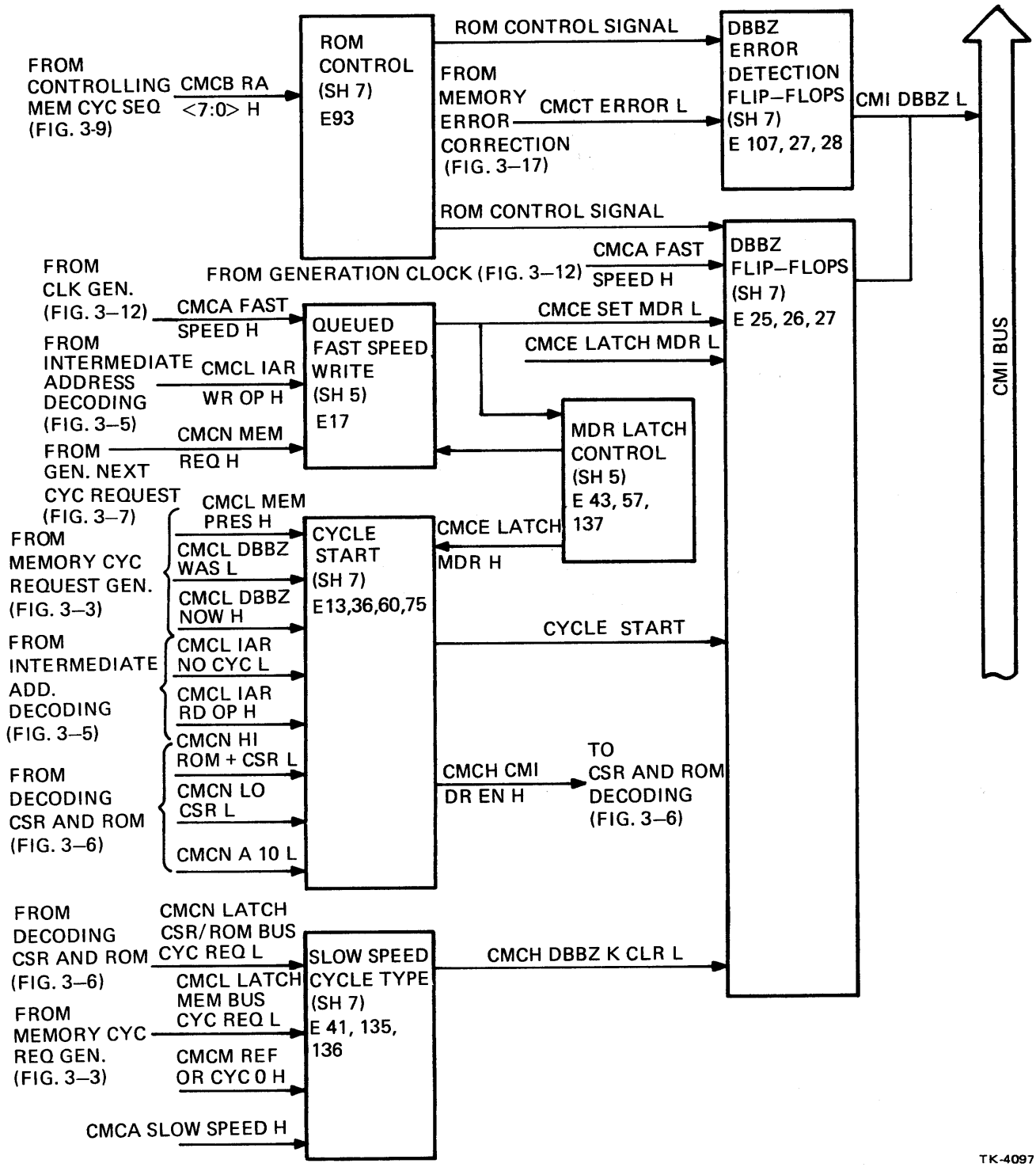

Figure 3-20 DBBZ Generation

During a read operation the memory controller asserts CMI DBBZ L in the cycle following the master device's assertion of CMI DBBZ L. Also, the memory controller continues asserting CMI DBBZ L until the READ data is ready for transmission. However, in a fast-speed write operation, CMI DBBZ L is not asserted if the memory controller was previously in the idle state. On the other hand, CMI DBBZ L is asserted when another fast-speed write operation is requested when the memory controller is busy doing a write operation.

3.17.1 ROM Control/DBBZ Error Detection Flip-Flops

The controlling memory cycle sequence circuit (see Figure 3-9) applies the control timing signals CMCB RA <7:0> H to the ROM control circuit. The ROM control circuit then applies a ROM control signal to the DBBZ flip-flop circuit and also to the DBBZ error detection flip-flop circuit. When these two signals are asserted, the DBBZ error detection flip-flop circuit asserts CMI DBBZ L at bus clock time (CMCA BUS CLK H). If no error is detected, the memory error correction circuit (see Figure 3-17) deasserts CMCT ERROR L, which clears the DBBZ error detection flip-flop circuit. An error extends the assertion of CMI DBBZ L for one more bus cycle.

3.17.2 Cycle Start

The cycle start circuit determines if: (1) a cycle type is being performed by verifying that CMI DBBZ is currently asserted (CMI DBBZ NOW H) but was not asserted the previous bus cycle (CMI DBBZ WAS L), (2) the addressed memory location exists (CMCL MEM PRES H), and (3) the cycle type is legitimate (CMCL IAR NO CYC L). Also, the cycle start circuit verifies that there is either a read operation (CMCL IAR RD OP H), a ROM is addressed, or a CSR is addressed (CMCH HI ROM + CSR L, CMCN LO CSR L and CMCN A IO L). Otherwise, the memory data register (MDR) is latched (CMCE LATCH MDR H), indicating a queued write is in progress. When these conditions exist, CYCLE START is applied to the DBBZ flip-flop circuit, which asserts CMI DBBZ L. When a queued operation is completed, CMCE SET MDR L is asserted, thus clearing CMI DBBZ L.

3.17.3 Queued Fast-Speed Write

The fast-speed write circuit determines if a fast-speed (CMCA FAST SPEED H) memory request (CMCN MEM REQ H) write operation (CMCL IAR WR OP H) is pending. When this condition is met, the queued fast-speed write circuit verifies if the memory data register (MDR) is latched (CMCE LATCH MDR L). If it is not, (CMCE LATCH MDR L deasserted), the queued fast-speed write circuit applies CMCE LATCH MDR L to the DBBZ flip-flop circuit, which prevents assertion of CMI DBBZ L. When the MDR is latched (CMCE LATCH MDR L asserted), the DBBZ flip-flop circuit asserts CMI DBBZ L.

3.17.4 Slow-Speed Cycle Type

The CSR bus cycle request (CMCN LATCH CSR BUS CYC REQ L), ROM bus cycle request (CMCN LATCH ROM BUS CYC REQ L), memory bus request (CMCN LATCH MEM BUS CYC REQ L), and refresh or cycle 0 (CM CM REF OR CYC 0 H) are all applied to the slow-speed cycle type circuit. When all the bus cycle requests are completed in slow-speed mode (CMCA SLOW SPEED H asserted), the slow-speed cycle type circuit applies CMCH DBBZ K CLR L to the DBBZ flip-flop circuit, which clears CMI DBBZ L.

3.18 INITIATING INITIALIZATION

The initiating initialization circuit consists of an initialization and bus control circuit, a controller direct clears circuit, an initialize high circuit, and an initialize and bus operation hold-off circuit. Refer to Figure 3-21.

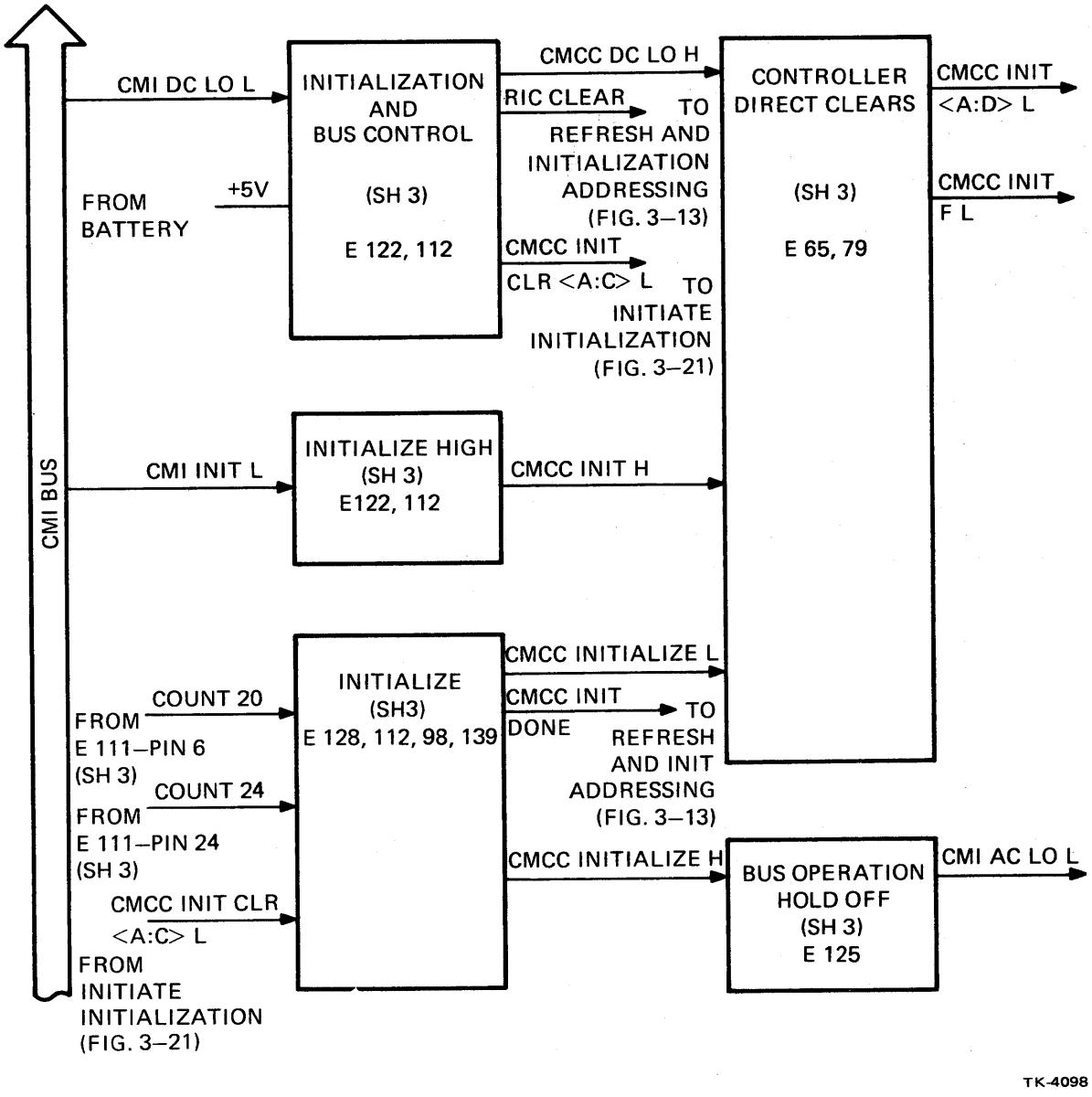

Figure 3-21 Initiating Initialization

This circuit initiates the initialization cycle to assure that the memory controller writes the proper correction code (ECC) and all 0s throughout the VAX-11/750 memory so that after a power-on sequence is completed, the memory responds to any cycle type, without irrelevent ECC error responses. Also, the initiating initialization circuit produces the clear signals CMCC INIT CLR <A:C> L and CMCC INIT <A:F> L, which are applied to the reset gates of flip-flops throughout the memory controller.

The initiating initialization circuit starts an initialization sequence when: (1) the main ac system power is first applied after the memory system has been operated from a battery backup unit beyond the allowable life of the batteries, or (2) no battery is present.

3.18.1 Initialization and Bus Control/Controller Direct Clears

CMI DC LO L from the CMI bus and the +5 V from the battery are applied to the initialization and bus control circuit. Asserting CMI DC LO L indicates that the memory controller's dc voltages are not within the specified limits and the memory system needs to be on battery backup to preserve the MOS memory contents. When the +5 V battery is not present, CMI DC LO L indicates that the battery is not supplying the proper voltage. But if CMI DC LO L is asserted and the dc voltage is restored (CMI DC LO L deasserted), RIC CLEAR is applied to the clear gates of the refresh and initialization counters (see Figure 3-13).

When RIC CLEAR is applied, the clear signals CMCC INIT CLR <A:C> L are asserted. Also, the signals CMCC INIT CLR <A:C> L are applied to the reset gates of flip-flops throughout the memory controller.

The refresh and initialization counters' E111 pin 8 output is deasserted with the clearing of the refresh and initialization counters. The deasserted E111 pin 8 output is then clocked (CMCA TI CLK H) into the initialize high circuit, which applies CMCC INITIALIZE H to the generating cycle type request circuit (see Figure 3-8). Next, the generating cycle type request circuit produces an initialize cycle type (CMCM INIT CY L). When the initialization is over, E98 gates the clock to E111 to prevent the reassertion of CMCC INITIALIZE H.

3.18.2 Bus Operation Hold-Off/Initialize High

The asserted CMCC INITIALIZE H is applied to the bus operation hold-off circuit, which produces CMI AC LO L. The asserted CMI AC LO L prevents all CMI bus data transactions while the initialization cycle is being performed. Also, CMCC INITIALIZE L is applied to the controller direct clear circuit, producing CMCC INIT F L.

After performing two complete passes through all the initialization addresses, the refresh and initialization counters' E111 pin 8 output is asserted, which, in turn, deasserts CMCC INITIALIZE H. Finally, the deasserted CMCC INITIALIZE H ends the initialization cycle.

When CMI INIT L is asserted on the CMI BUS and applied to the initialize high circuit, CMCC INIT H is asserted. The asserted CMCC INIT H is then applied to the controller direct clear circuit, producing CMCC INIT <A:F> L.

3.19 CONTROL/STATUS REGISTER 2

The read-only control/status register 2 (CSR2) provides status information to the memory controller. Figure 3-22 shows CSR2's bit allocations. Tables 3-6 and 3-7 define the bit allocations of CSR2.

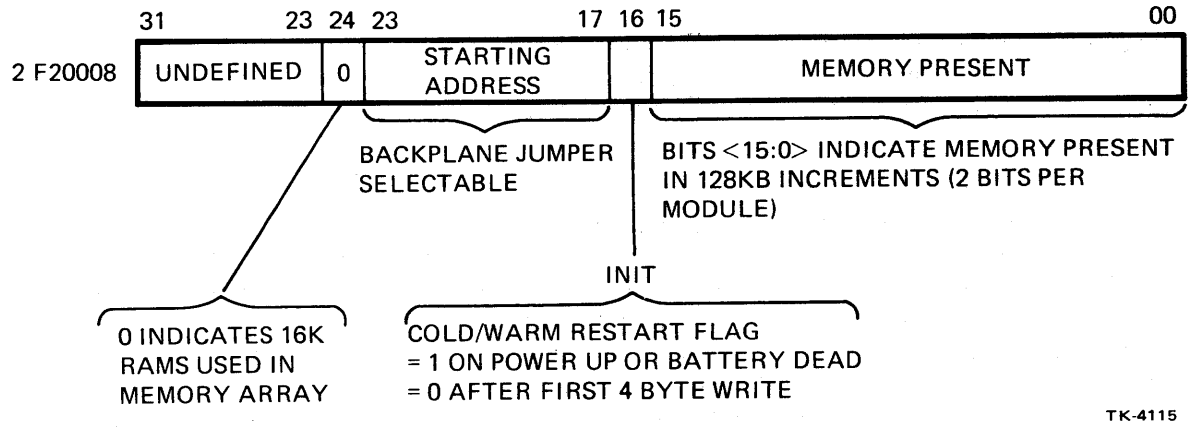

Figure 3-22 CSR2 Bit Allocations

Table 3-6 CSR2 Bit Allocations

Bit Allocation	Definition
CSR2 <15:00>	These bits indicate which of memory slots 11 through 18 are occupied and whether the memory arrays are fully populated or half-populated. Refer to Table 3-7.
CSR2 bit 16	CSR2 bit 16 is used to indicate that the memory controller is on battery back-up beyond the life of the batteries. When CSR2 bit 16 is a 1, the battery back-up has failed and an initial power-up has occurred. The first memory long-word write after the initial power-up then clears CSR2 bit 16.
CSR2 <23:17>	Starting Address
CSR2 bit 24	When CSR2 bit 24 reads as a 0, the memory controller is using 16K MOS RAM arrays.
CSR2 <31:25>	Undefined; could be read as 1s or 0s.

Table 3-7 CSR2 <15:00> Bit Allocation Map

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
0 0	0 0	0 0	0 0	0 0	0 1	0 1	1 1
Slot 18	Slot 17	Slot 16	Slot 15	Slot 14	Slot 13	Slot 12	Slot 11

3.19.1 Memory Present Map

CSR2 bits <15:00> comprise the memory present map. The memory present map represents the amount of memory present in the system and the locations in the memory backplane where the memory arrays are inserted. There are eight possible locations for the memory arrays. Two memory array types are possible: fully populated (256K bytes) and half-populated (128K bytes). Also, the memory present map is divided into 128K segments.

3.19.2 Starting Address

CSR2 bits <23:17> contain the memory starting address, which is usually all 0s. The address is defined by backplane pins; therefore, a backplane pin with a ground jumper has a 1 value, while a pin with no jumper has is a 0 value. See Table 3-8.

The starting address and the memory present map are applied to the memory array address generation circuit (see Figure 3-2).

Table 3-8 Starting Address Jumpers

Starting address jumpers	Backplane pin (slot 11)
17	B86
18	B87
19	B88
20	B89
21	B90
22	B92
23	B93

3.20 DIAGNOSTIC REGISTER

In diagnostic mode the diagnostic register verifies that the ECC logic is functioning properly. The diagnostic register is a redundant register of CSR1's lower seven bits because when the CPU writes to CSR1, identical write data is loaded into both CSR1 and the diagnostic register. Refer to Figure 3-23.

When in diagnostic mode during a read operation, the diagnostic register substitutes the seven check bits instead of reading the seven check bits from memory. Therefore, by using the diagnostic register's seven check bits, the read operation verifies that the ECC logic is functioning properly. When in diagnostic mode during a write operation, the generated check bits are stored in both CSR1 and the diagnostic register. While in ECC DIS mode during a read operation, the read check bits are also stored in CSR1 and the diagnostic register.

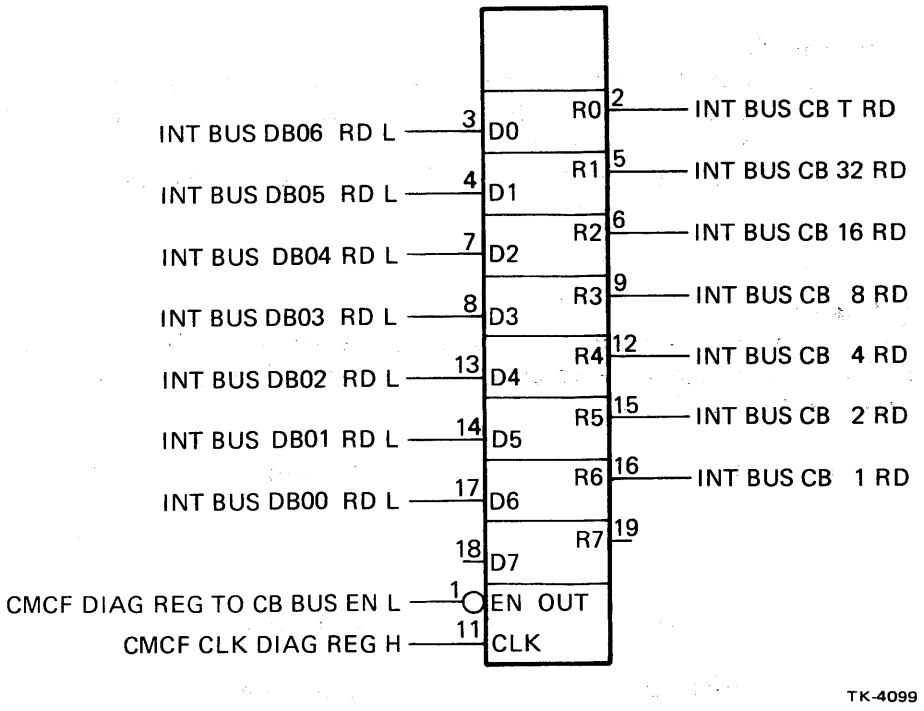

Figure 3-23 Diagnostic Register

3.21 M8728 MOS STORAGE ARRAY

The M8728 storage array module is a hex-multilayer printed circuit board containing 156 16-pin MOS RAM chips and their associated drive circuitry. The memory array consists of receivers, drivers, row select and random access memory. Refer to Figure 3-24. The internal bus is located on connectors A and B. Refer to Figure 3-25.

The MOS chips are arranged in 4 banks of 39 chips, with each bit position of the 39-bit data word consisting of one chip from each bank. Furthermore, the 39-bit word consists of four 8-bit data bytes and seven check bits that are used for error detection and correction.

In any cycle, each chip on the memory array receives an address and INT BUS CAS TIM L. INT BUS RAS TIM L is then decoded and applied to one of the four banks to strobe in the address. However, row addresses and column addresses use the same address lines on the memory array; thus, both are multiplexed onto the same seven address lines by the memory controller.

There is no control included on the memory array. Thus all addresses and control signals are generated and applied via the internal bus by the memory controller.

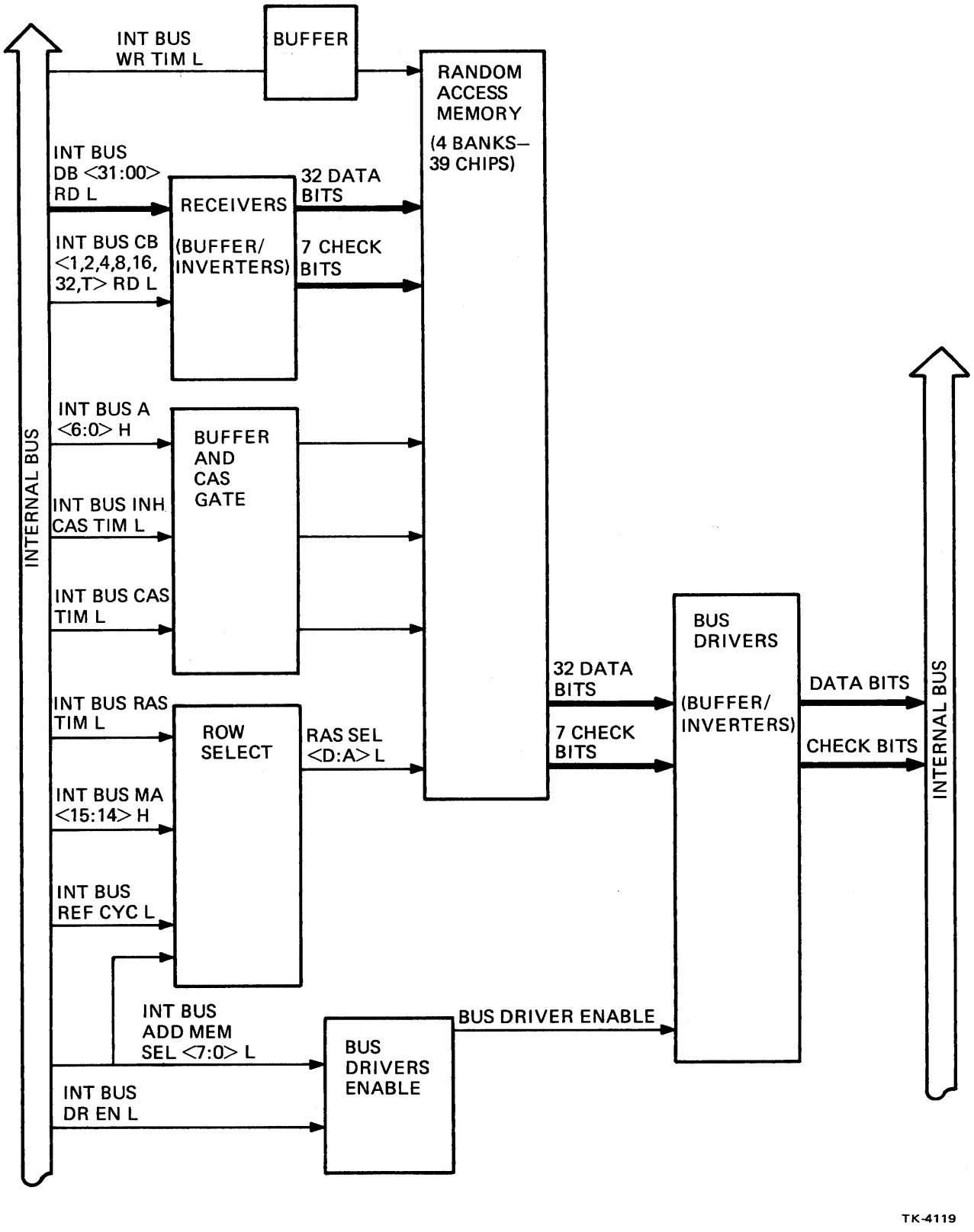

Figure 3-24 Memory Array, Block Diagram

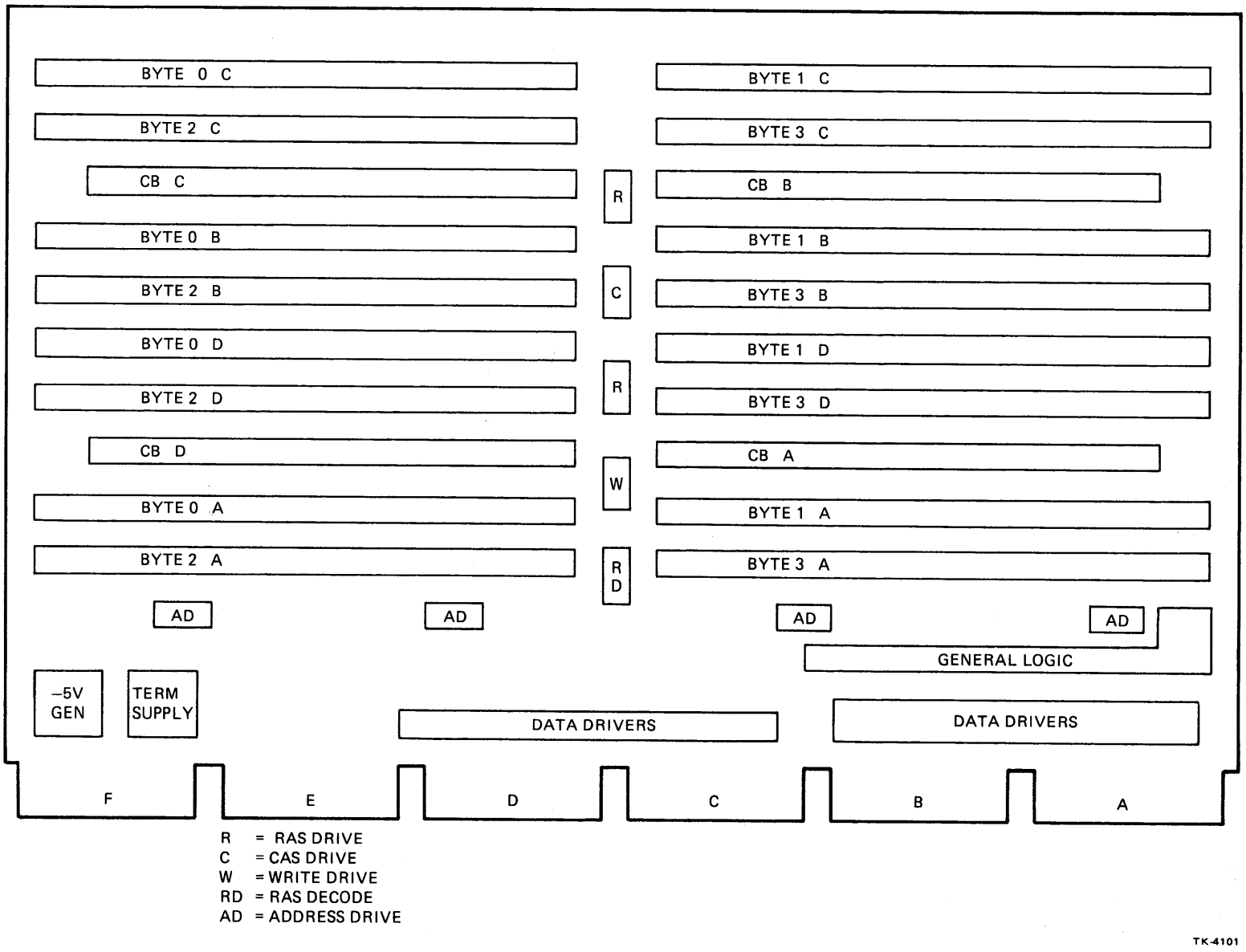

Figure 3-25 M8728 MOS Storage Array

3.21.1 Receivers and Bus Drivers

During a write operation and the write portion of an internal read-modify-write operation, 32 data bits and 7 check bits (INT BUS DB <31:00> RD L) from the internal bus and INT BUS CB (1, 2, 4, 8, 16, 32, T) RD L are applied to the input pins of all 4 banks of chips via the buffer inverter receivers. The memory controller then strobes (INT BUS WR TIM L) the 32 data bits and 7 check bits into the chips corresponding to the selected row.

During a read operation and the read portion of an internal read-modify-write operation, 32 data bits and 7 check bits from the selected row of chips are applied to the internal bus via the bus drivers. When the memory select (INT BUS MEM SEL <7:0> L) is asserted, with the memory controller asserting INT BUS DR EN L, the tri-state bus drivers become enabled when the data is read from the MOS array.

3.21.2 Row Select

During read, write, and refresh operations, the addressed MOS chips are enabled by the address and timing control signals. Therefore, the memory array address generation circuit (see Figure 3-2) of the memory controller applies INT BUS ADD MEM SEL <7:0> L to the memory arrays. INT BUS ADD MEM SEL <7:0> L enables all the functions of the array except the refresh operation.

The address selection circuit (see Figure 3-11) of the memory controller applies INT BUS MA <15:14> to the row select circuit of the memory array. The row select circuit decodes INT BUS MA <15:14>, thus generating the row address strobes RAS SEL <A:D> L, which select one of the four banks of RAMs when INT BUS RAS TIM L is asserted. INT BUS RAS TIM L is derived from the memory controller's generating RAS and CAS circuit (see Figure 3-4).

The address selection circuit of the memory controller applies INT BUS REF CYC L to the row select circuit that forces a memory select. This forced memory select circuit selects all four banks of chips with RAS SEL <D:A> lines; thus, all four banks of chips on the memory array are refreshed simultaneously.

3.21.3 Random Access Memory

The 16K storage cells of the MOS chips are arranged in a matrix, each storage cell having a unique row address and column address. The memory controller's address selection circuit (see Figure 3-11) multiplexes the row and column addresses onto the seven address lines INT BUS A <6:0> H.

When the memory controller's selection circuit asserts INT BUS RAS TIM L (at row address strobe time), the seven bits of the address are strobed into the MOS chip's row address register. Later, INT BUS CAS TIM L (column address strobe time) is asserted and strobes the second seven bits of the address into the MOS chip's column address register.

APPENDIX A
MEMORY ADDRESS PROCESSOR (MAP)

GENERAL DESCRIPTION

The MAP chip performs the following operations.

1. Decodes a 7-bit CMI bus address that enables a memory array.
2. Provides a jumper-designed starting address offset from the backplane.
3. Provides automatic detection of fully and half-populated memory arrays.
4. Provides a status flag that indicates when a valid address is input to access memory space.

Refer to Figure A-1 for the memory address processor pin allocations, and Figure A-2, a block diagram of the memory address processor. Refer to Table A-1 for the MAP gate array signals.

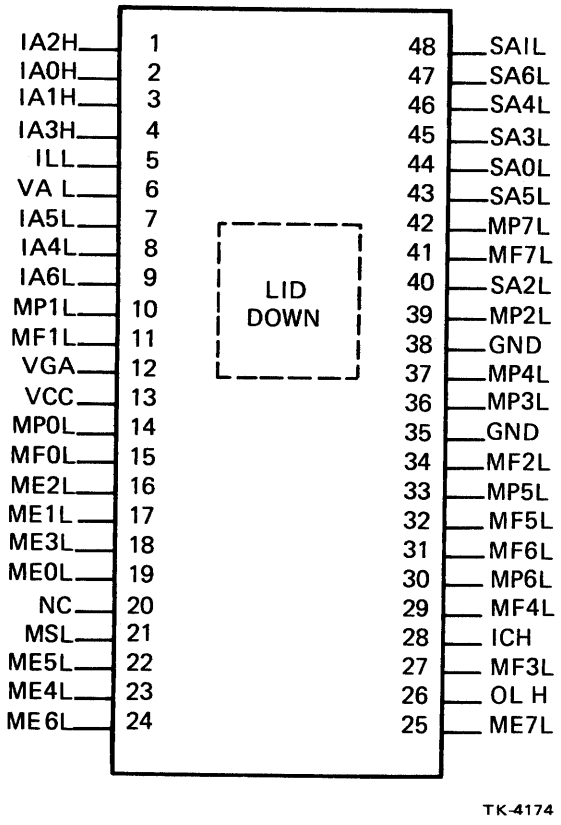

Figure A-1 Memory Address Processor Pin Allocations

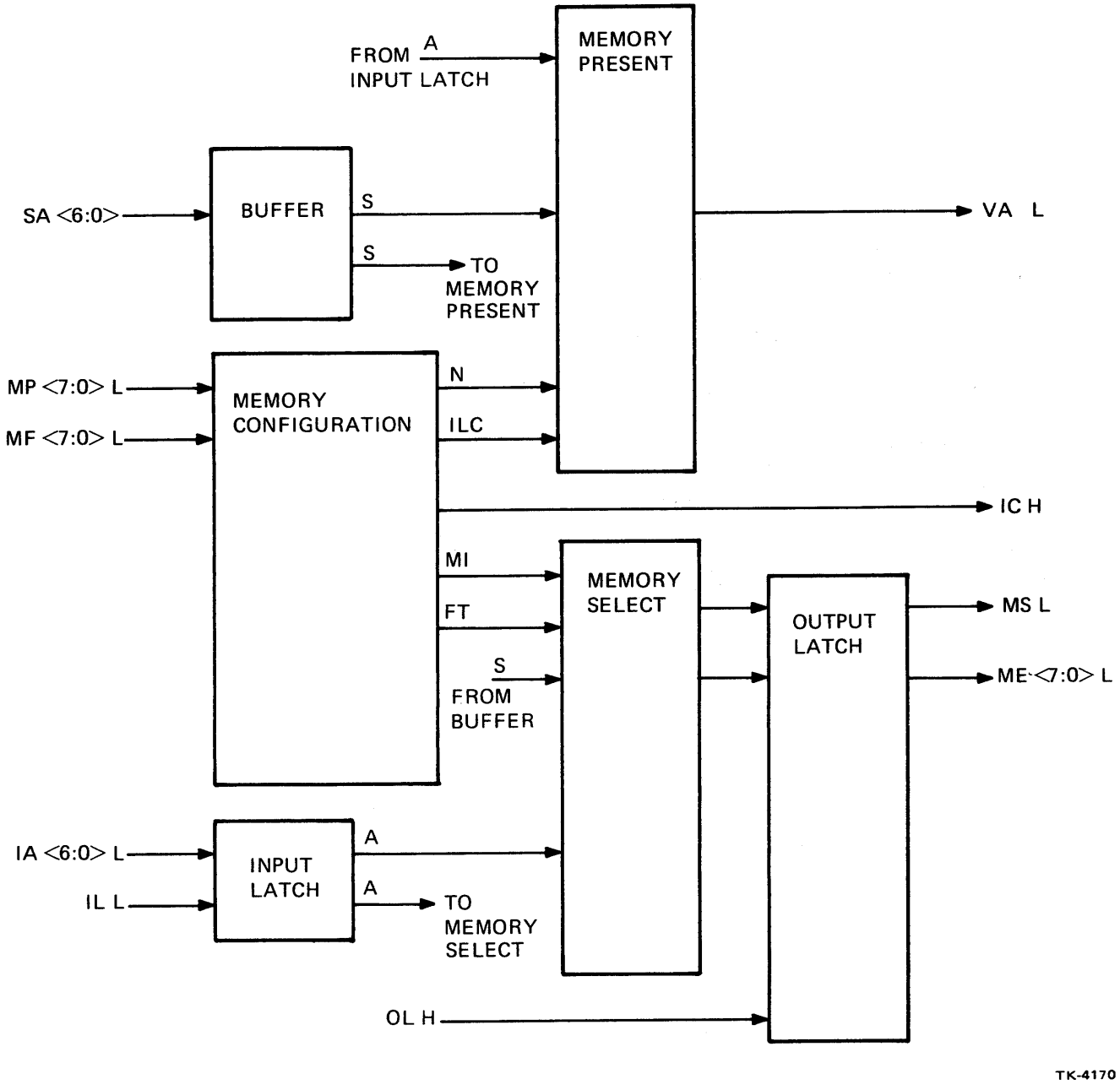

Figure A-2 Memory Address Processor, Block Diagram

Table A-1 MAP Gate Array Signals

Signals	Description
IA <6:0>	CMI DATA <23:17> are the seven most significant bits of the unprocessed address. They are used to select a memory array.
SA <6:0> L	The starting address consists of the seven most significant bits of the address offset, which is determined by backplane jumpers. However, the low-order bits are defined as being all 0s. Moreover, the starting address is defined on 128K-byte boundaries.
MP <7:0> L	Memory present represents the slot locations in the memory backplane where the memory arrays are inserted. There are eight possible locations for the memory arrays (slots <18:11>).
MF <7:0> L	Fingerprint bits (MF <7:0> L) represent the amount of memory present per array. The memory arrays can be fully populated (256K bytes) or half-populated (128K bytes).
IL L	IA <6:0> is continually accepted as long as IL remains deasserted. However, when IL L is asserted, IA <6:0> is latched into the MAP chip. Any changes in IA <6:0> are ignored as long as IL L remains asserted.
OL H	When OL H is deasserted, the outputs ME <7:0> L are derived from IA <6:0> by asserting OL H; the ME <7:0> that is present at the time of the transition occurred is latched. As long as OL H is asserted, any changes in the IA <7:0> are ignored.
IC H	IC H is the illegal configuration flag. This flag is asserted when the received memory present MP <7:0> and the fingerprints MF <7:0> indicate an illegally configured set of memory arrays.
VA L	VA L is the valid address flag. This flag is asserted when the latched IA <6:0> input is an address that accesses available memory. Available memory is a function of the starting address offset (SA <6:0>) and the configuration decoding (MP <7:0> and MF <7:0>).
ME <7:0> L	ME <7:0> L are the memory array enable signals. Each enable signal is applied to a corresponding memory array slot. When a valid address is input, only one memory array enable output is asserted so as to enable only the respective memory array. The memory array enable outputs are latched under the control of OL H. These outputs are not valid for illegally configured memory arrays.

APPENDIX B
MEMORY DATA LOOP (MDL)

GENERAL DESCRIPTION

An MDL chip, four of which are used to handle a 32-bit data field, performs the following operations.

1. Transfers 32 data bits in either direction between the CMI bus and the internal bus.
2. Transfers 21 address bits from the CMI bus to the address selection circuit.
3. Transfers the bootstrap data words from the bootstrap PROM to the CMI bus.

The MDL chip also has the following features.

1. Includes two control/status registers (CSR0 and CSR1) and two CSR comparators, which compare the contents of CSR0 and CSR1 with AUX MAR's contents.
2. Contains an intermediate address register (IAR) and a ROM data latch.
3. Contains a memory data register (MDR).

Refer to Figure B-1 for the memory data loop pin allocations, and Figure B-2, a block diagram of the memory data loop. Table B-1 describes the byte control signals, and Tables B-2, B-3, B-4, B-5 and B-6 are the MDL's signal truth tables.

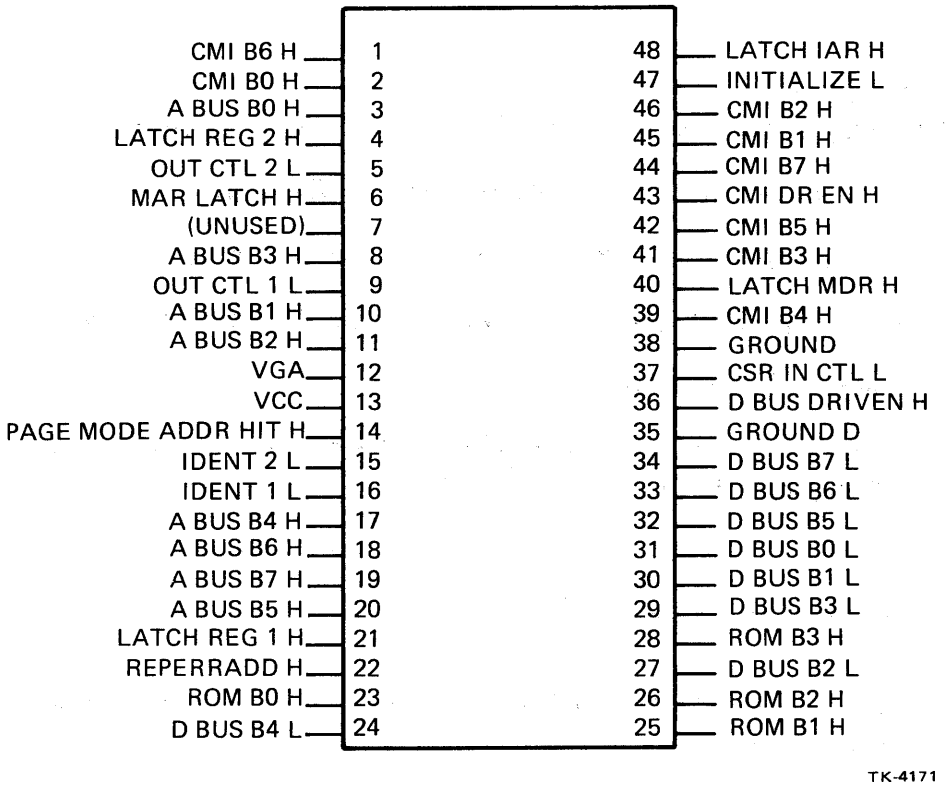

Figure B-1 Memory Data Loop Pin Allocations

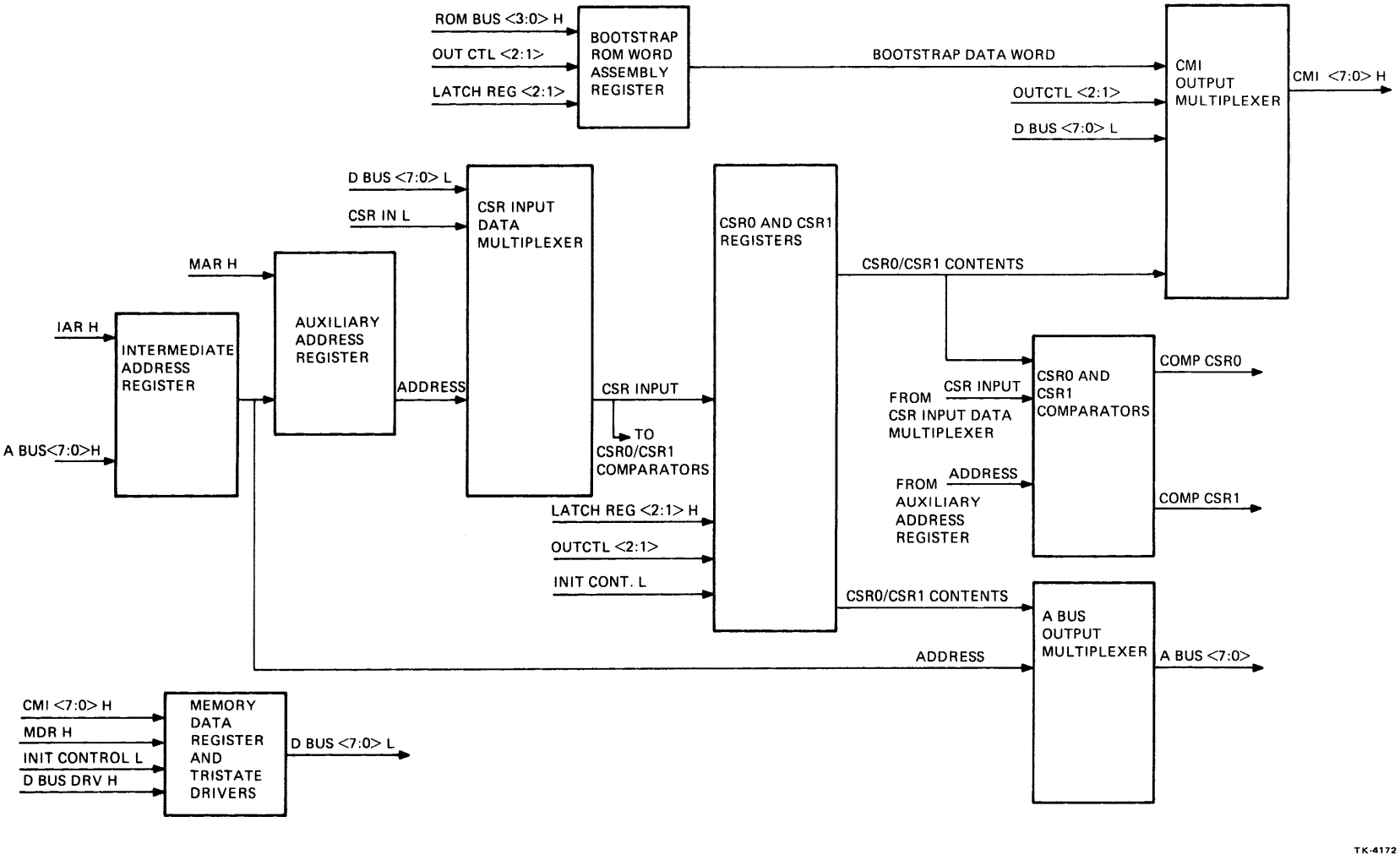

Figure B-2 Memory Data Loop, Block Diagram

Table B-1 Byte Control Signals

Signals	Description
INIT CONT. IN L	All bits of the MDR, CSR0, and CSR1 are forced to 0 when INIT CONT. IN L is asserted.
IDENT <2:1>	The IDENT <2:1> inputs define for which byte a particular MDL is being used by controlling the processes that differ among the four bytes. The IDENT <2:1> lines are decoded according to Table B-2.
CSR IN L	CSR IN L and the IDENT <2:1> lines control the four inputs to the MDLs according to Table B-3.
OUT CTL <2:1>	The two control lines OUT CTL <2:1> determine which of four sources is to be selected and applied to the CMI bus. See Table B-4 to find how these two control lines are decoded.
LATCH REG <2:1> H	LATCH REG <2:1> H controls ROM latch register 1 and ROM latch register 0, or CSR1 and CSR0, depending on the state of LATCH REG <2:1> H. When a data word is being assembled in the ROM buffer latches, these control lines control ROM latches 0 and 1, respectively. When any other function is being performed, these lines control CSR0 and CSR1, respectively. Tables B-5 and B-6 summarize the functions of LATCH REG <2:1> H.
IAR H	By asserting IAR H, CMI DATA <23:02> H is stored in the MDL's internal intermediate address register (IAR). When IAR H is deasserted, the IAR is transparent.
MAR H	The MDL's internal auxiliary address register is latched with the contents of the IAR when MAR H is asserted. When MAR H is deasserted, the auxiliary address register is transparent.
MDR H	CMI DATA <31:00> are latched into the MDL's internal memory data register (MDR) with the assertion of MDR H. The MDR is transparent when MDR H is deasserted.
CMI DR EN H	The data from the internal bus is driven onto the CMI BUS with the assertion of CMI DR EN H.
D BUS DRV H	By asserting D BUS DRV H, the MDR data is driven onto the internal bus.
CSR 0 H	When CSR0's contents and the contents of the auxiliary address register (AAR) are compared and found to be equal, CSR 0 H is asserted.
CSR 1 H	When CSR1's contents and the contents of the auxiliary address register (AAR) are compared and found to be equal, CSR 1 H is asserted.
ROM BUS	ROM BUS is composed of the four bits of data from the bootstrap PROM that are to be applied to the CMI bus via the ROM data latches.

Table B-1 Byte Control Signals (Cont)

Signals	Description
D BUS <7:0> L	D BUS <7:0> L are 32 bidirectional lines connecting the MDL to the memory arrays and MECs.
CMI <7:0> H	CMI <7:0> H are 32 bidirectional lines that transfer data between the CMI bus and the MDL.
A BUS H	A BUS H is composed of the 21 lines used for a memory array's address.
A BUS 1 H	A BUS 1 H is used to control ECC disable mode in the memory controller.
A BUS 2 H	The memory controller is put in diagnostic check mode with the assertion of CMCS A BUS 2 H.
A BUS 3 H	The memory controller is put in page mode with the assertion of A BUS 3 H.
A BUS 4 H	A BUS 4 H controls enable CRD error status mode.
A BUS 7 H	A BUS 7 H indicates an RDS error log request.

Table B-2 IDENT <2:1> Lines

IDENT 2	IDENT 1	BYTE CODE
H	H	0
H	L	1
L	H	2
L	L	3

Table B-3 CSR IN L Line

CSR IN L	Byte Code*	Selected Input
H	0	D BUS <31:00> L
H	1	MAR internal memory address register
H	2	MAR internal memory address register
H	3	D BUS <31:00> L
L	X	CMI DATA <31:00> H

*Refer to Table B-2 for byte code generation.

Table B-4 CMI Source Selection

Output Control 2	Output Control 1	Source Selection
H	H	D BUS <31:00>
H	L	The latched ROM data from ROM BUS <3:0>
L	H	CSR 0
L	L	CSR 1

Table B-5 LATCH REG <2:1> H Lines

Output Control 1	Output Control 2	Latch Register 2	ROM Latch 1 State	CSR State
X	X	H	latched	latched
H	H	L	latched	transparent
L	H	L	transparent	latched
H	L	L	latched	transparent
L	L	L	latched	transparent

Table B-6 CMCE MDL <3:0> LATCH REG 1 H Lines

Output Control 1	Output Control 2	Latch Register 1	ROM Latch 0 State	CSR0 State
X	X	H	latched	latched
H	H	L	latched	transparent
L	H	L	transparent	latched
H	L	L	latched	transparent
L	L	L	latched	transparent

APPENDIX C
MEMORY ERROR CORRECTION (MEC)

GENERAL DESCRIPTION

The MEC chip, which is used in pairs, performs the following operations.

1. Detects all single- and double-bit errors on a 32-bit data field.
2. Corrects single-bit errors on a 32-bit data field.
3. Generates seven check bits or syndrome bits according to the Modified Hamming Code.

Refer to Figure C-1 for the memory error correction pin allocations, and Figure C-2, a block diagram of the memory error correction gate array. Table C-1 describes the MEC gate array signals.

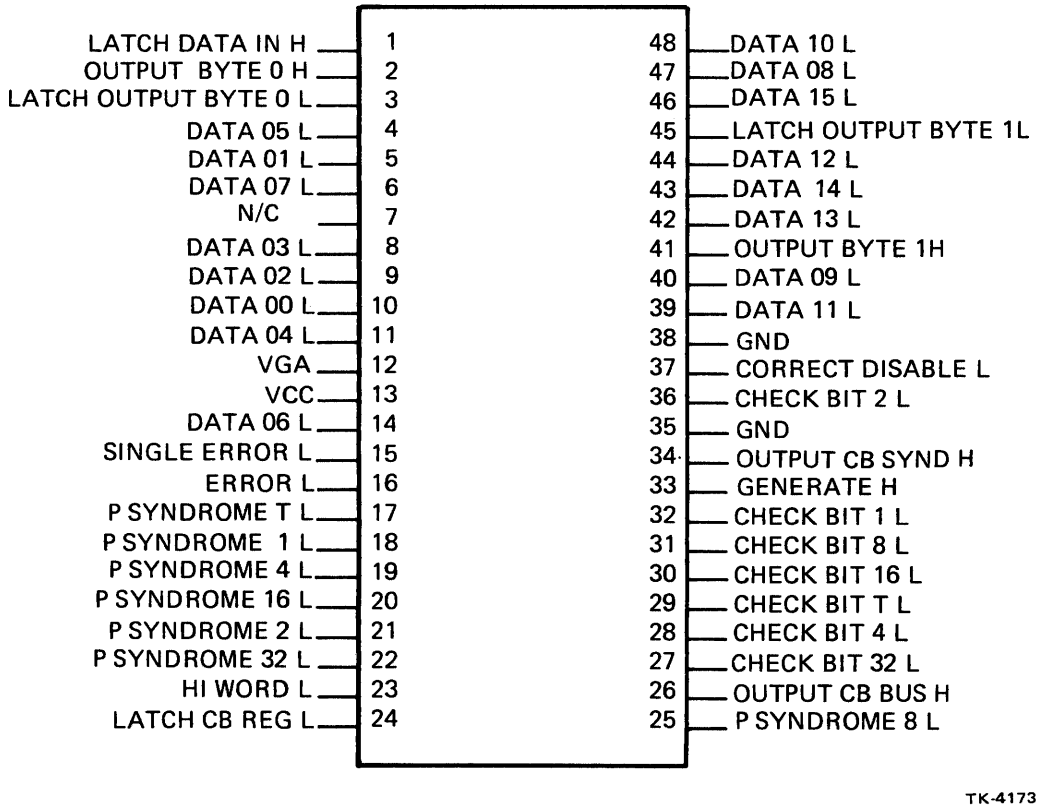

Figure C-1 Memory Error Correction Pin Allocations

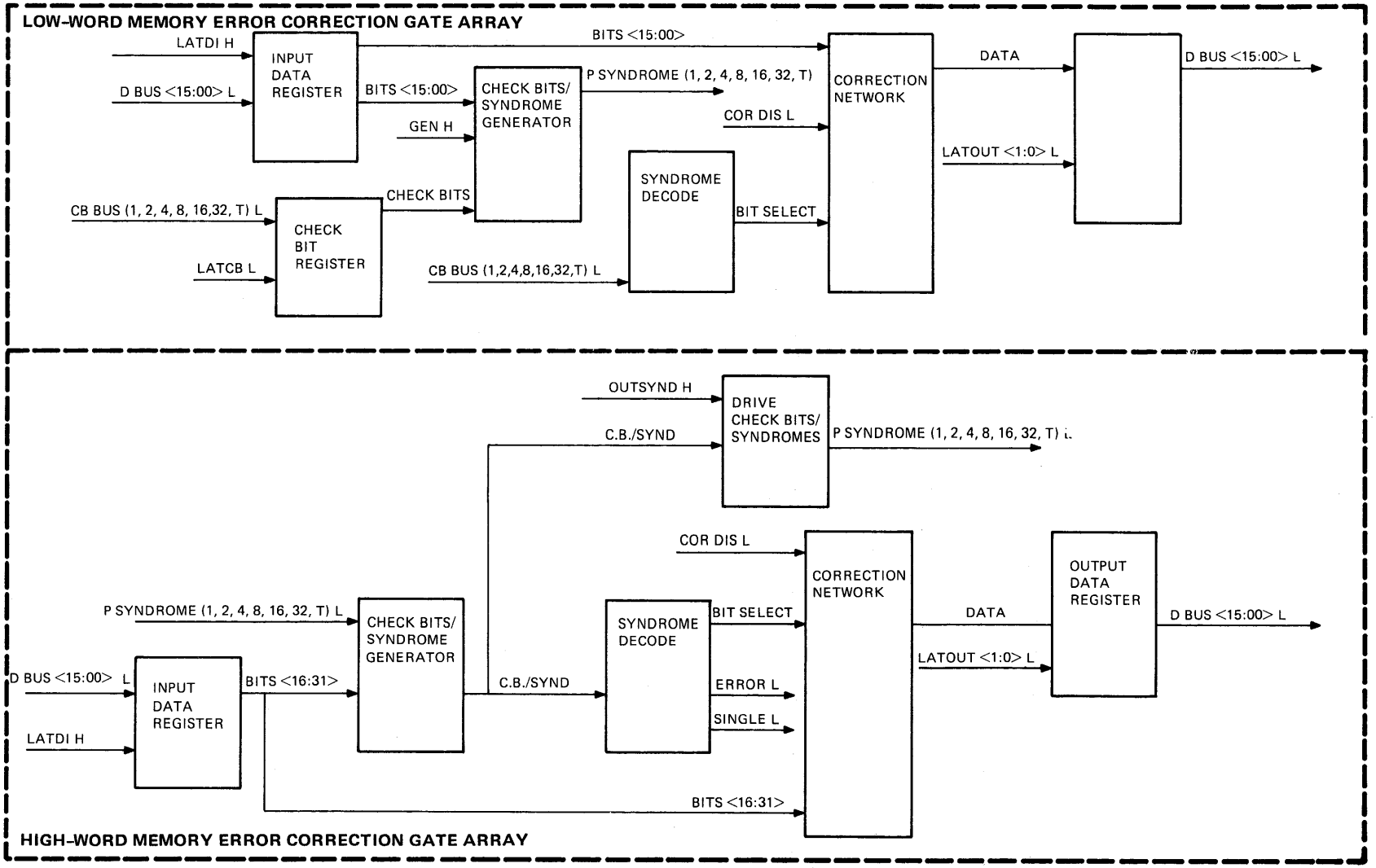

Figure C-2 Memory Correction, Block Diagram

Table C-1 MEC Gate Array Signals

Signals	Description
D BUS <15:00> L	Sixteen bidirectional data lines that connect the memory arrays to the memory data loop gate arrays.
CB BUS L 1,2,4, 8,16,32,T	Seven bidirectional check bit lines. In the low-word MEC, the seven lines are the check bit bus, while in the high-word MEC, the seven lines are the partial syndromes from the low-word MEC.
GEN H	When the MEC is generating check bits, GEN H forces all check bit inputs to 0 in the low-word MEC.
LATDI H	When LATDI H is asserted, the data (D BUS <15:00>) is latched and held in the MECs. The MEC's input latches are transparent when LATDI H is deasserted.
LAT CB L	The CB BUS L CB register is latched with CB BUS L when CMCF LW LATCH CB REG L is asserted. The CB register is transparent when LAT CB L is deasserted.
LATOUT <1:0> L	LATOUT <1:0> L controls the MEC's output latches on a per-byte basis. When LATOUT <1:0> L is asserted, the data is latched; otherwise, the MEC's output latches are transparent.
OUT CB BUS H	Asserting OUT CB BUS H causes the check bit register to drive the check bits onto the check bit bus.
OUT SYND H	OUT SYND H causes the generated check bits or syndromes to be driven onto the check bit bus.
OUT BYT <1:0>	OUT BYT <1:0> are tri-state control signals that enable check bits or syndrome bits onto the check bit bus.
CB BUS L	CB BUS L are either the seven partial syndromes from the low-word MEC, the seven check bits, or the syndromes (depending on whether checking or generating) from the high-word MEC.
ERROR L	ERROR L is asserted when an error is detected.
SINGLE L	When both ERROR L and SINGLE L are asserted, a single-bit error is indicated. If ERROR L is asserted but SINGLE L is not, a multiple error is indicated.

VAX-11/750 Memory System
Technical Description
EK-MS750-TD-001

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 *Technical Documentation Catalog*, 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
444 Whitney Street
Northboro, MA 01532
Attention: Printing and Circulating Service (NR2/M15)
Customer Services Section

Order No. EK-MS750-TD-001

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 and Development
1925 Andover Street (TW/B01)
Tewksbury, MA 01876