Honeywell

LEVEL 6

HARDWARE

TYPE CMC 9009/9010 MAIN MEMORY MANUAL

TYPE CMC9009/9010
MAIN MEMORY MANUAL
(Interim)

Document No. 71010278-100 Order No. FQ36, Rev. 0

This manual supports memory controller board types BF2MYE and BF2MYP.

RECORD OF REVISIONS

REVISION	DATE	AUTHORITY	AFFECTED PAGES
-100	Jan. 1978	BLCD 77935	--

Hardware Publications, M&TO, Billerica, MA 01821

FQ36

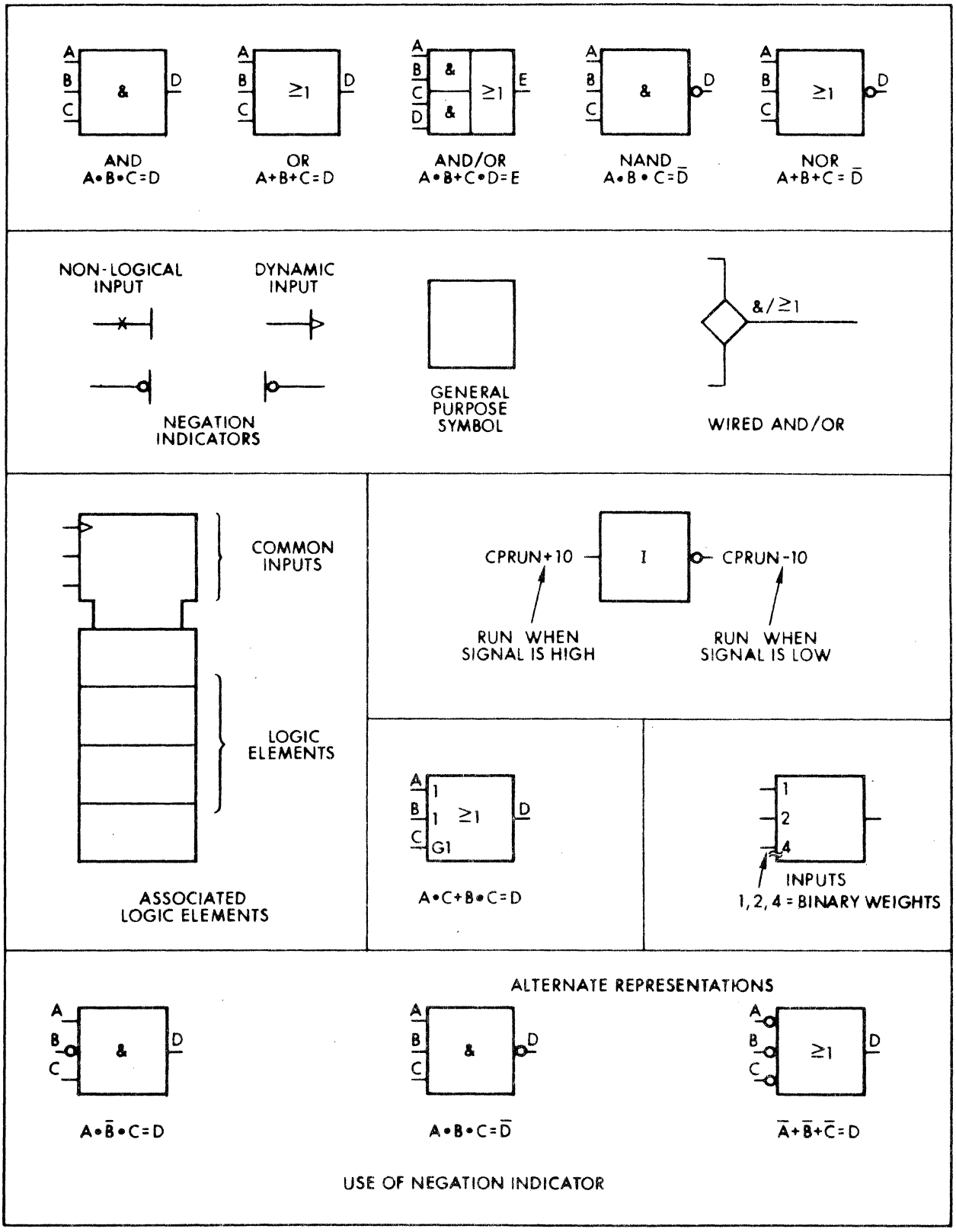
CONTENTS

CONTENTS

ILLUSTRATIONS

TABLES

LOGIC SYMBOLOGY



1 INTRODUCTION

This manual describes the functionality and provides the theory of operation information for the Level 6 128K single-word fetch and 128K double-word fetch main memory controllers.

1.1 GENERAL DESCRIPTION

The 128K single-word fetch or 128K double-word fetch main memory controller is a Random Access Memory (RAM), which utilizes N-channel metal oxide semiconductor technology as the basic storage medium. It is physically contained on a single two-layer etched board with accompanying Memory-Pacs, and is optional for the Model 40 Central Processor Unit (CPU). The purpose of these memory controllers is to provide an expanded main storage facility, defined as attachable 32-kiloword (32K-word) Memory-Pacs for the Level 6 computer system.

The 128K main memory subsystem, which contains the memory controller and the 32K-word Memory-Pacs, communicates with the CPU and other processors and controllers through an interface over a common bus, called the Megabus* (see Figure 1-1).

The Megabus routes data, addressing, and control commands to the memory controller, allowing the unit to perform single-word or double-word read, single-word or byte write, or diagnostic operations. The memory controller responds by delivering control, data, and/or error signals to the bus for dissemination to the appropriate subsystems associated with the CPU. Figure 1-2 shows the interface signals between the memory subsystem and the Megabus.

*Trademark of Honeywell Information Systems Inc.

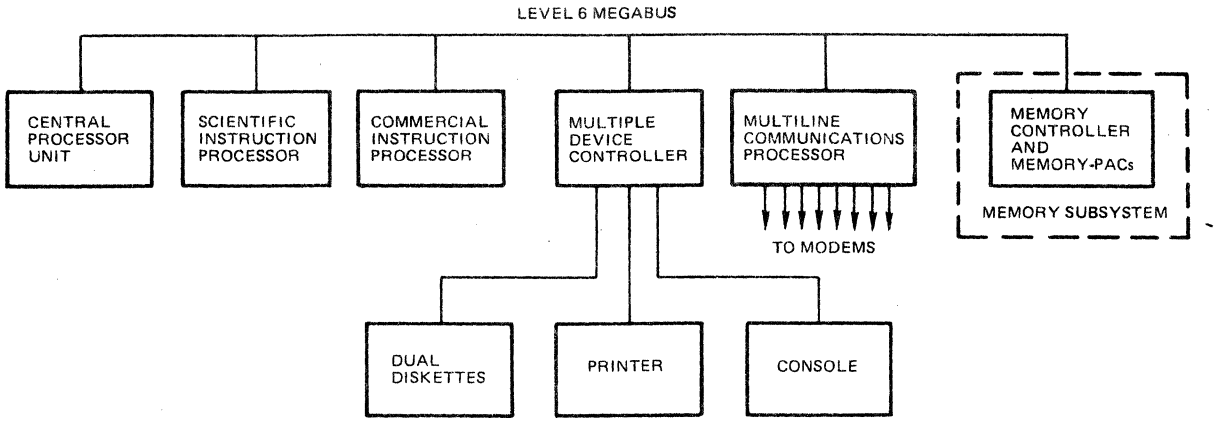


Figure 1-1 128K Memory Subsystem Communication Block Diagram

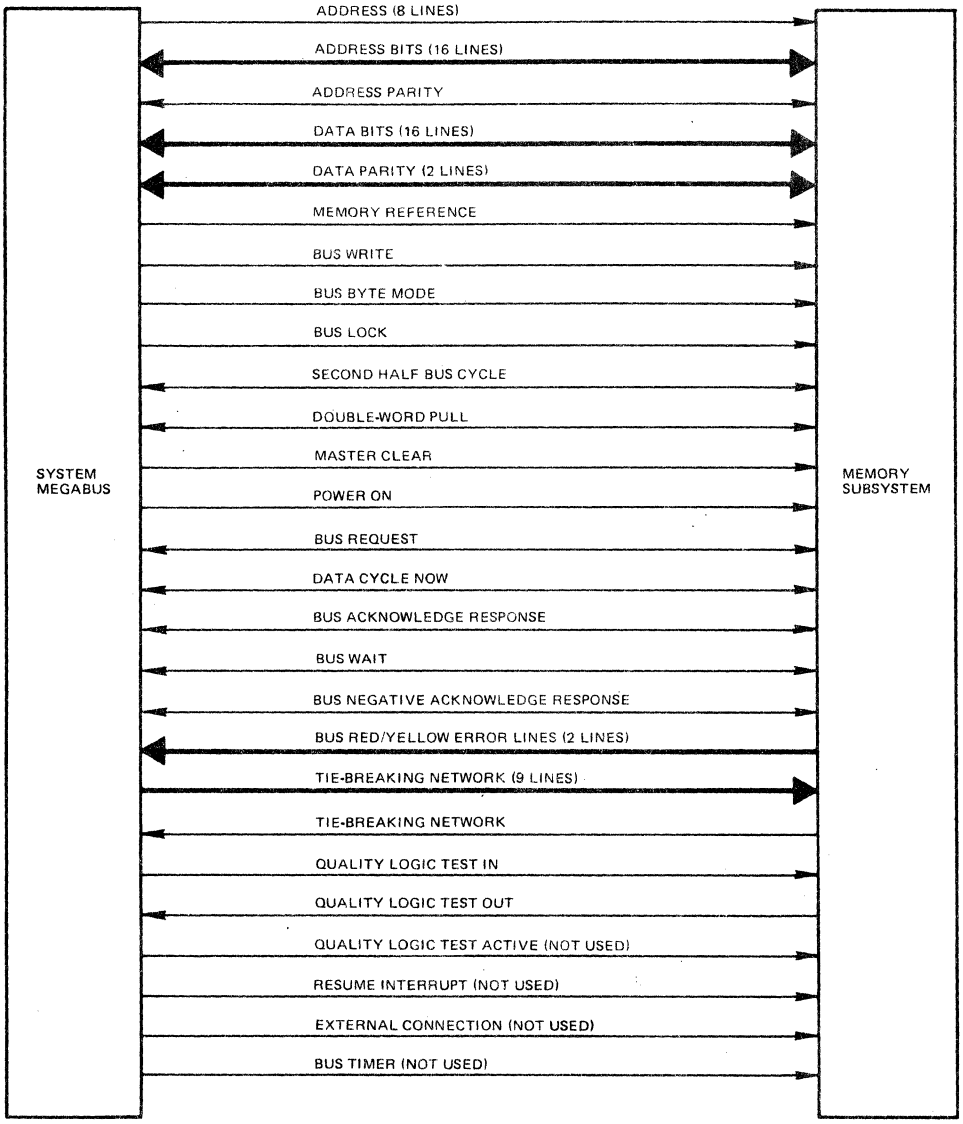


Figure 1-2 Megabus/Memory Subsystem Interface Signals

The memory controller contains all of the support logic that is needed to perform memory timing, refresh operations, data functions, address distribution and decoding, bus interfacing, and memory storage, as well as the logic that is used to protect the memory contents during power outages or brownouts via battery power. Memory can also be configured to detect and correct single-bit errors and to detect multiple-bit errors via the Error Detection and Correction (EDAC) logic.

The 32K-word Memory-Pac is a high-speed, random-access semiconductor board that performs data storage functions or control information functions without restrictions on address sequencing, specific data patterns, or repetition rates (i.e., within the specifications), utilizing 16K MOS RAM technology for the basic storage medium.

The following list provides additional information relative to the general characteristics of the memory subsystem:

- 8-bit byte length
- 2-byte word length
- Memory controller capacity (maximum of four Memory-Pac boards):

  Parity: 128K-words by 18 bits
  EDAC: 128K-words by 22 bits

- Memory-Pac capacity:

  Parity: 32K-words by 18 bits
  EDAC: 32K-words by 22 bits.

1.2 MEMORY CONTROLLER BOARD DESCRIPTION

The major assembly of the memory subsystem is the memory controller, which supports up to four 32K-word Memory-Pacs (128K), as shown in Figure 1-3. Memories that have the EDAC option installed use memory controller board type BF2MYE. Memories that have the Parity option installed use memory controller board type BF2MYP. All of the support logic needed to accommodate 128K words of information is contained on either of these memory controller boards. Address, data, and control signals are transferred between memory and other subsystems attached to the Megabus via the two 50-pin connectors that attach the memory controller board to the Megabus. Located on the component side of the memory controller board are four 80-pin connectors, which provide a mounting area for the 32K-word Memory-Pacs.

1.2.1 Memory Controller Physical Layout

Refer to the Series 60 Level 6 Model 43 System Manual, Order No. FN36, for the layout of a memory controller board with the supporting 32K-word Memory-Pacs and the modular order of positioning the Memory-Pacs when they are added to the memory controller.

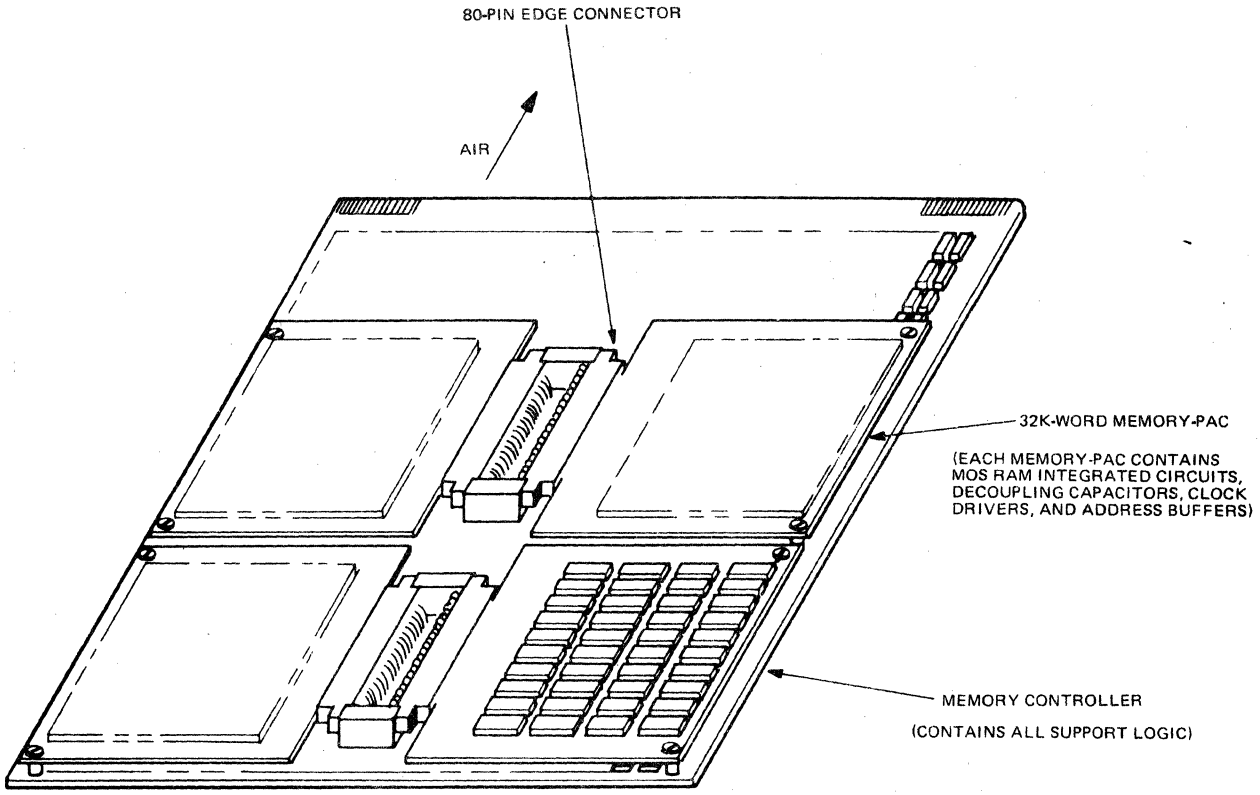


Figure 1-3 Memory Controller and Memory-Pac Subsystem

1.2.2 Memory Controller Switches

Several configuration switches are located on the memory controller board, and are defined in the following discussion. Figure 1-4 shows the location of the memory controller configuration switches. There are six rocker type switches and one toggle switch contained on the memory controller board. One 8-position rocker switch, labeled F19, is used to set the memory controller board's address position. Table 1-1 lists the logical state to which the F19 switch should be set to accommodate memory addressing capabilities. Three 10-position rocker switches, labeled E16, E17, and E18, are used to set the configuration of the memory controller. Table 1-2 lists the logical state to which the E16, E17, and E18 switches should be set. Two 2-position rocker switches, labeled F04S and H04X, are used to set the EDAC bypass and double-word fetch options. Table 1-2 lists the logical state to which the F04S and H04X switches should be set. One 2-position memory test toggle switch, labeled P02N, is utilized during memory Test and Verification (T&V) routines. On the parity controller board, memory test toggle switch P02N is used in conjunction with the state of the Bus Byte Mode signal to set or reset the data parity bits. On the EDAC controller board, toggle switch P02N is used to force all of the EDAC encoder (check) bits to Zeros. The T&V routines have specified coded data patterns, which are written into memory to simulate the picking and dropping of bits. When reading out of memory from an EDAC system, the decoded portion of the EDAC logic points to and corrects the proper bit position, verifying the proper EDAC operation.

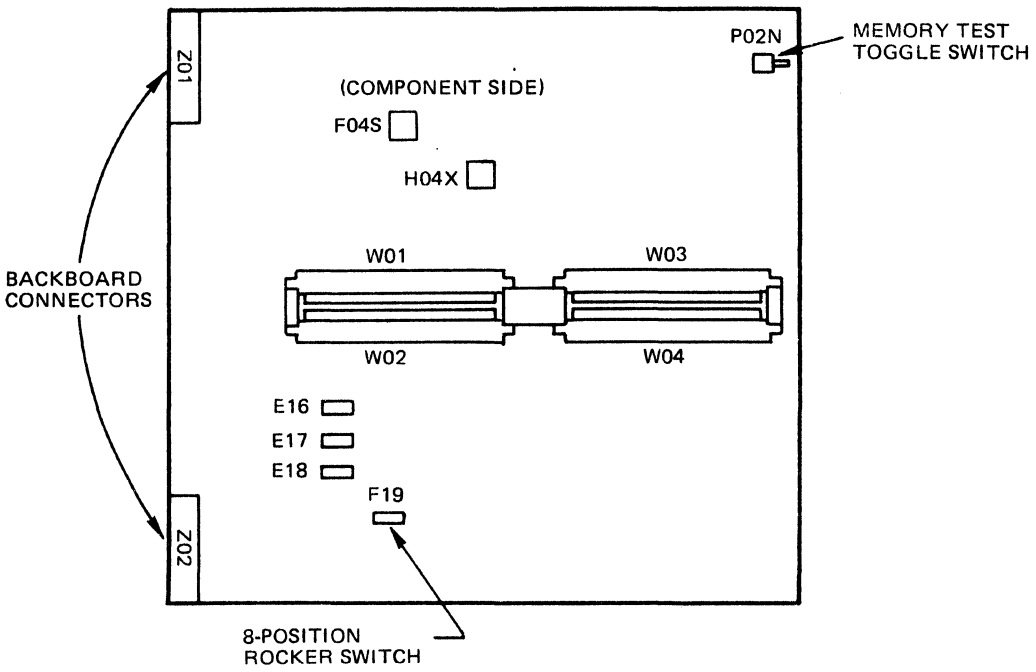


Figure 1-4 Memory Controller Select Switches

Table 1-1 Memory Module Select Switch F19 Settings

MODULE	SWITCH F19							
	1	2	3	4	5	6	7	8
1 (0 through 128K)	0	0	1	1	1	1	1	1
2 (128K through 256K)	0	0	0	1	1	1	1	1
3 (256K through 384K)	0	0	1	0	1	1	1	1
4 (384K through 512K)	0	0	0	0	1	1	1	1
5 (512K through 640K)	0	0	1	1	0	1	1	1
6 (640K through 768K)	0	0	0	1	0	1	1	1
7 (768K through 896K)	0	0	1	0	0	1	1	1
8 (896K through 1,024K)	0	0	0	0	0	1	1	1

0 = OFF; 1 = ON.

Table 1-2 128K Memory Controller Switch Configurations

TYPE	SWITCH DESIGNATION		E16										E17										E18										H04X		F04S	
	NO.	INTERNAL PRODUCT IDENTIFIER, BF2MYE	1	2	3	4	5	6	7	8	9	10*	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1**	2	1	2
EDAC	1	Single-Word	0	1	0	1	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0	0
	2	Double-Word	0	1	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0
TYPE	SWITCH DESIGNATION		E16										E17										E18										H04X		F04S	
	NO.	INTERNAL PRODUCT IDENTIFIER, BF2MYP	1	2	3	4	5	6	7	8	9	10*	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	1	2
Parity	1	Single-Word	0	1	0	1	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	1	0	0
	2	Double-Word	0	1	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1	0	0	1	0	1	0

*For double-word, set to 1 to inhibit Hidden Refresh.

**Must be equal to 1 for EDAC bypass.

HONEYWELL PROPRIETARY AND CONFIDENTIAL

1.3 MEMORY-PAC BOARD DESCRIPTION

Memory-Pacs are added to the memory controller (see Figure 1-3) in increments of 64K words, configured for Double-Word Pull (i.e., two Memory-Pacs), and 32K words, configured for Single-Word Pull (i.e., one Memory-Pac). The Memory-Pacs are configured with a number of MOS RAM integrated circuits for bit storage and a minimum amount of support logic. This support logic includes clock decoding and board select functions that direct data to the proper module, integrated circuit, and location within an integrated circuit.

Systems without the EDAC option installed use Memory-Pac type BS2SH6, which contains 36 MOS RAM integrated circuits and permits the utilization of 32K by 18 bits (i.e., 16 data bits and two parity bits). Systems that have the EDAC option installed use Memory-Pac type BS2SH4, which contains 44 MOS RAM integrated circuits and permits utilization of 32K by 22 bits (i.e., 16 data bits and six EDAC check bits).

For the physical layout of the Memory-Pacs and the modular order of positioning the Memory-Pacs when they are added to the memory controller board, refer to the Series 60 Level 6 Model 43 System Manual, Order Number FN36.

1.4 MEMORY OPTIONS

Additional memory can be added to the memory subsystem in increments of 64K words, configured for Double-Word Pull (i.e., two Memory-Pacs), and 32K words, configured for Single-Word Pull (i.e., one Memory-Pac). The memory can be added as 32K-word Memory-Pacs or as 128K-word controller boards. The maximum memory for the Level 6 Model 43 is one megaword.

1.4.1 Error Detection and Correction

The Level 6 128K single-word and double-word fetch memory can be configured to contain EDAC logic that allows a single-bit read error in any memory word to be detected, corrected, and reported to the unit that is receiving data. Double-bit read errors in any memory word are also detected and reported to the unit that is receiving data, but are not corrected.

1.4.2 Memory Save Power Supply

Memory can be configured to include logic that protects the memory contents during power outages or brownouts. With this option, a battery-powered supply provides the necessary power lines to the timing, refresh, and control circuitry so that refresh cycles can be run uninterrupted when main power fails.

1.5 REFERENCE DOCUMENTS

The following documents can be used in conjunction with this manual for a better understanding of the overall operation of the 128K single-word or double-word fetch main memory:

- Series 60 Level 6 Model 43 System Manual, Document Number 71010316-100, Order Number FN36

- Series 60 Level 6 CMC9009/CMC9010 Memory Reference Manual, Document Number 71010279-100, Order Number FQ37
- Series 60 Level 6 Model 40 Central Processor Unit Manual, Document Number 71010300-100, Order Number FN28
- Series 60 Level 6 Power System Manual, Document Number 71010290-300, Order Number FL34.

II THEORY OVERVIEW

The material in this section provides an overview description for all elements of the Level 6 128K single-word and double-word main memory subsystem. These elements are as follows:

- Hardware
- Megabus operation
- Megabus formats.

2.1 MEMORY SUBSYSTEM HARDWARE OVERVIEW

The memory subsystem is comprised of the memory controller and 32K-word Memory-Pacs. The memory initiates activity cycles in response to requests made by other subsystems on the Megabus. Figure 2-1 is a major block diagram of the 128K memory subsystem, showing all of the major data transfer paths between the memory subsystem registers.

2.1.1 Timing Generation

The timing generator starts the memory timing sequence and sets a Memory Busy condition. If the cycle started is a read operation, the timing generator enables a signal, which triggers a second half read response in the bus control logic. This response attempts to acquire the bus so that the information requested on the first half read can be transmitted to the controller that initiated the cycle. If the request from the other subsystem was for two words (double-word fetch), the request remains for an additional second half bus cycle.

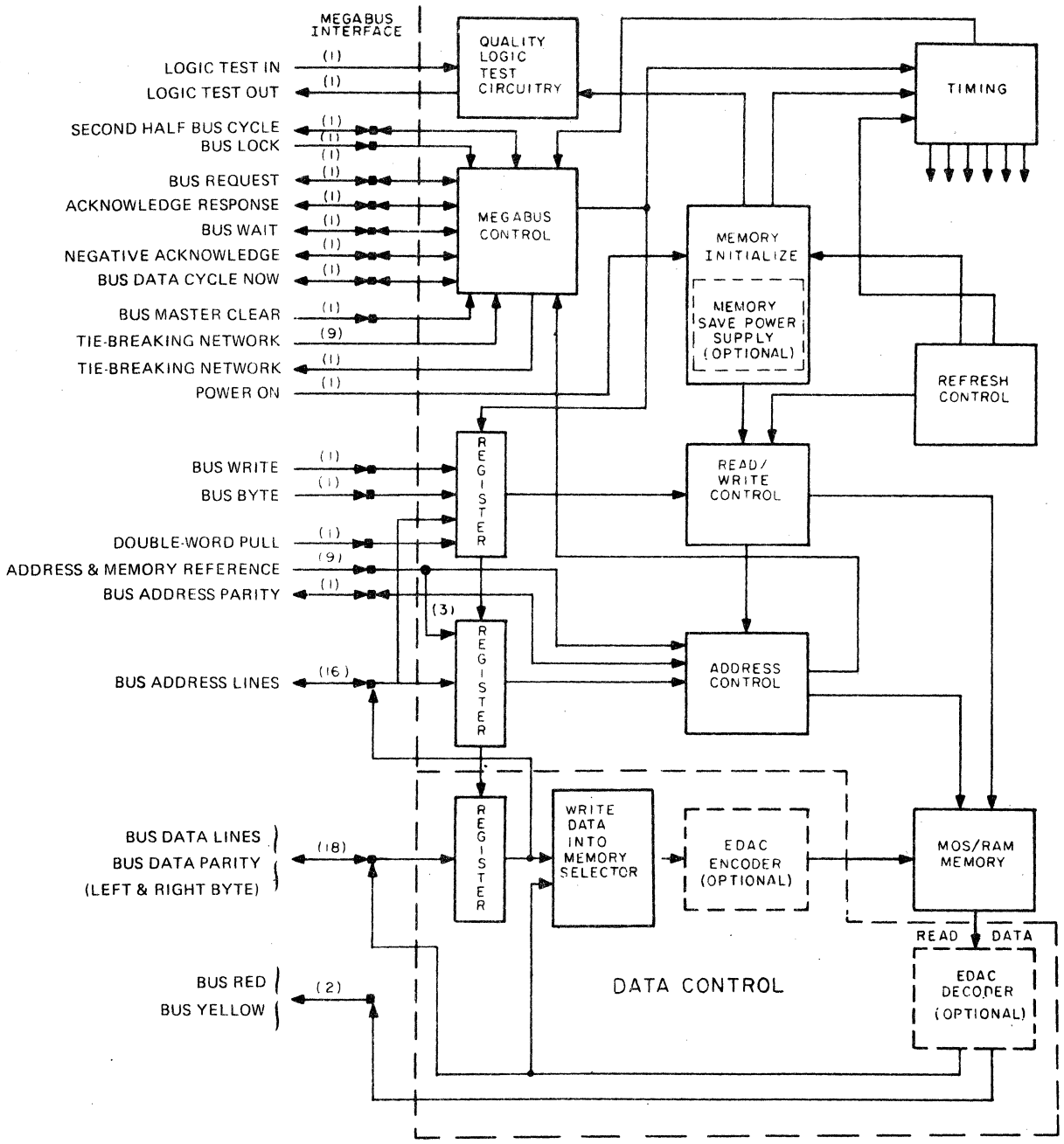


Figure 2-1 Memory Major Block Diagram

Internal memory cycles can be started by refresh commands or by initializing memory. The timing generator supplies the memory controller with all of the signals used to control memory access and cycle times. These signals are: Chip Timing, Memory Busy, Data Cycle Now, the memory request for a Second Half Bus Cycle, and partial write information.

2.1.2 Address Control

The proper address generation and distribution are essential for reliable memory operation. The address control logic contains all of the circuitry involved in initialization, address decoding, read/write selection, and the refreshing of the main memory subsystem.

A decoding format is implemented on the 24-bit address, which enters the memory controller. The memory address bits determine the selection of a particular memory controller, 32K-word Memory-Pac, half Memory-Pac chip selection, 16K memory address, and left or right byte selection for Byte operations.

2.1.3 Memory Initialize

During initial power up, memory is initialized to clear it of erroneous information. Memory is initialized by setting or clearing its contents to a specific state. The read/write lines are forced into a Write mode, and a Wait condition is presented to any unit which tries to access memory. The address register is incremented through all locations of memory. Zeros are written into the data locations while Ones are written into the parity locations. For EDAC systems, the correct check bits are written into the check bit location. The complete process of initializing 128K words of memory requires slightly less than 16 seconds.

Memory can be optionally configured to include a Memory Save Power Supply that supplies voltage to the memory controller for refresh control to protect the contents of memory during power outages or brownouts. This battery power provides some of the timing, refresh, and control circuits with the necessary power lines so that the MOS/RAM integrated circuits are refreshed during power loss.

2.1.4 Read/Write Control

Memory read/write logic transfers Read, Write, and Byte instructions between the Megabus and the addressed location in memory. The read/write circuitry determines whether the read/write line will perform in the capacity of a read, write, or a read followed by a write as in the case during a Byte command. During Byte Write operations, the read/write line is forced to take on the characteristics as dictated by a signal received from the timing generator.

2.1.5 Refresh Control

Due to the storage medium (16K MOS RAMs), logic is required which periodically refreshes the information stored in memory. The refresh logic includes a means of substituting a Refresh cycle for a Megabus-oriented cycle and places memory in an internal Read mode. All clock decoding is overridden, and the same column of every integrated circuit in memory is refreshed. Additional circuits, called Hidden Refresh, are provided for performing a Refresh when configured as a double-word fetch memory.

2.1.6 Data Control

The data control logic enables data to be written into memory from data multiplexers and/or read from memory and transferred to the Megabus transceivers and the data multiplexers. The Level 6 memory subsystem uses either a parity or an EDAC data format, depending on the customer-selected option. Data control is explained in four parts: Read Data Flow, Write Data Flow, EDAC Encoder/Decoder, and Data Error Reporting.

2.1.6.1 Read Data Flow

During a Read cycle, 16 data bits and two parity bits for parity memories or 16 data bits and six check bits for EDAC memories are transferred from the MOS RAM memory and latched to local data registers.

2.1.6.2 Write Data Flow

Two types of write operations are performed by memory: Word Write and Byte Write. During a Word Write operation, 16 data bits and two parity bits are gated from the data bus receivers to the data registers. A Byte Write operation consists of a read of the memory location into which the new byte is to go and a write of the new byte into memory.

2.1.6.3 EDAC Encoder/Decoder

To enhance the reliability of the memory subsystem, the EDAC option is available. The EDAC encoder and decoder circuitry, inserted within the data paths, perform necessary generation and checking functions, which provide correction of a single-bit error and detection of a double-bit error. Two error signals are generated to indicate single-bit or double-bit errors.

2.1.6.4 Data Error Reporting

Memory error reporting to the CPU or other components on the Megabus is via two lines: Bus Red and Bus Yellow. Data error reporting is applicable only to controllers with the EDAC option installed.

2.1.7 MOS RAM Memory

The main component of the memory subsystem is the Memory-Pac. Each memory controller supports a maximum of four Memory-Pacs, which are organized in formats of 32K by 18 or 22 bits, depending on the presence or absence of the EDAC option.

For storage of information, the Memory-Pac utilizes 16K-word MOS/RAM chips, which optimize speed, power, and density. The 16,384 bits per chip are organized in a matrix of 128 rows by 128 columns. The row and column to be accessed are determined by the address control group while information is gated to memory through the data control group. All signals are TTL compatible and distributed to the Memory-Pac via circuitry on the memory controller board.

2.1.8 Megabus Control

The bus control logic contains: (1) circuitry involved in generating and accepting bus cycles, and (2) quality logic test circuitry.

The Megabus is interfaced with memory via transceivers that allow data, address, and certain control signals to be routed to and from memory. However, for certain control signals and tie-breaking signals, other individual circuits (standard TTL) must be used to interface memory with the Megabus.

Each unit (in this case the memory) that interfaces with the Megabus requires a tie-breaking network to resolve priority problems on the bus. The problems arise when two units on the bus generate simultaneous requests. The tie-breaking network allows the request of the highest priority unit first. The priority on the bus is decided by physical position. Memory is at the bottom of the bus and has the highest priority. The CPU is at the top of the bus and has the lowest priority. All other units on the bus are located between the memory and the CPU.

2.1.9 Quality Logic Test Circuitry

Each controller connected to the bus has the Quality Logic Testing (QLT) functionality to ensure the integrity of the system. A successful QLT indicates to the sytem that all units on the bus have power, the cables are in place, there are no missing units on the bus, and no test has failed. The QLT circuitry in the memory controller does not perform a quality logic test, as is done in other controllers, but reflects the net QLT status of all units of lower priority than the memory. During the initialization sequences, memory does not accept cycle requests until it has completed its QLTs.

2.2 MEGABUS OPERATION

The Megabus (see Figure 2-2) provides a common communication path (interface) among all units of the Level 6 System. The Megabus is asynchronous in design, permitting units of varying speeds to operate efficiently on the same system. The 128K single-word or double-word fetch memory subsystem communicates with the CPU and other processors and controllers over the Megabus, receiving commands from the CPU.

2.2.1 Master/Slave Relationship

The Megabus is bidirectional, permitting any two units to communicate with each other at a given time. The transfer of information between units forms a master/slave relationship (i.e., the unit requesting and receiving access to the Megabus becomes the master unit while the unit being addressed becomes the slave unit). If the communication requires a response, the responding slave unit assumes the role of master unit and the requesting unit (previous master unit) becomes the slave unit.

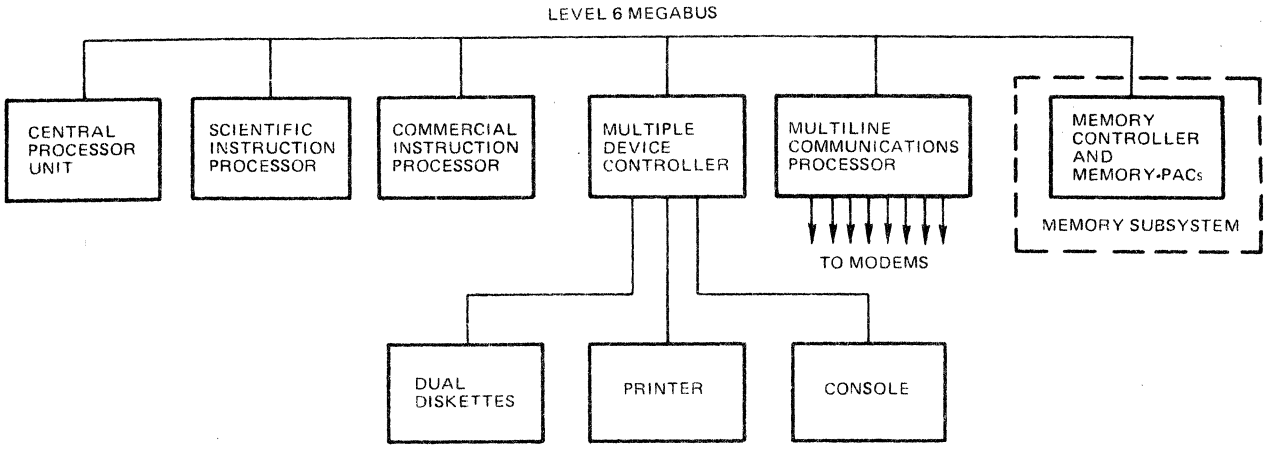


Figure 2-2 128K Memory Subsystem Communication Block Diagram

All information transfers are from the master unit to the slave unit and each transfer is referred to as a Megabus cycle. This cycle is the period of time in which the requester (master unit) asks for use of the Megabus. If no other unit of a higher priority is making a Megabus request, use of the Megabus is granted to the requester (master unit). The master unit then transmits its information to the slave unit and the slave unit acknowledges the communication.

2.2.2 Megabus Usage

Common types of Megabus operations are listed in Table 2-1. These operations require either one, two, or three Megabus cycles. Information transfers that are considered write operations require one Megabus cycle, while transfers that are considered read operations require an additional Megabus cycle for the response (i.e., for Single-Word Read operations, the second Megabus cycle is used to receive the data; for Double-Word Read operations, the second Megabus cycle is used to obtain the first word and the third Megabus cycle is used to receive the second word).

NOTE

Once a Megabus cycle is granted, the types of operations performed between the master unit and the slave unit are a function of the specific functionality of the two units.

Other types of Megabus operations, such as controller-to-controller transfers, are not listed in Table 2-1; however, the Megabus architecture makes no restrictions in this regard.

Table 2-1 Common Types of Megabus Operations

TYPE OF OPERATION	SOURCE	DESTINATION	NUMBER BUS CYCLES
Instruction Fetch (one word)	CPU	Memory	2
Instruction Fetch (two words)	CPU	Memory	3
Operand Fetch (one word)	CPU/CIP	Memory	2
Operand Fetch (two words)	CPU/SIP	Memory	3
Operand Store (word)	CPU/CIP/SIP	Memory	1
Operand Store (byte)	CPU/CIP	Memory	1
DMA Read (word)	Controller	Memory	2
DMA Read (byte)	Controller	Memory	2
DMA Write (word)	Controller	Memory	1
DMA Write (byte)	Controller	Memory	1

2.2.3 Memory Megabus Responses

The memory response signals Bus Acknowledge, Bus Negative Acknowledge, and Bus Wait initiate or deny a request for a Megabus cycle from a master unit. Failure of any one of them to respond causes the CPU to perform the Dead Man Time-Out operation that allows a delay of 5 microseconds to occur before the CPU clears the Megabus.

2.2.4 Memory Bus Request

A Megabus request is made by memory only during a Read Second Half Bus Cycle when memory is responding to the Megabus; i.e., when memory is ready to transmit previously requested data.

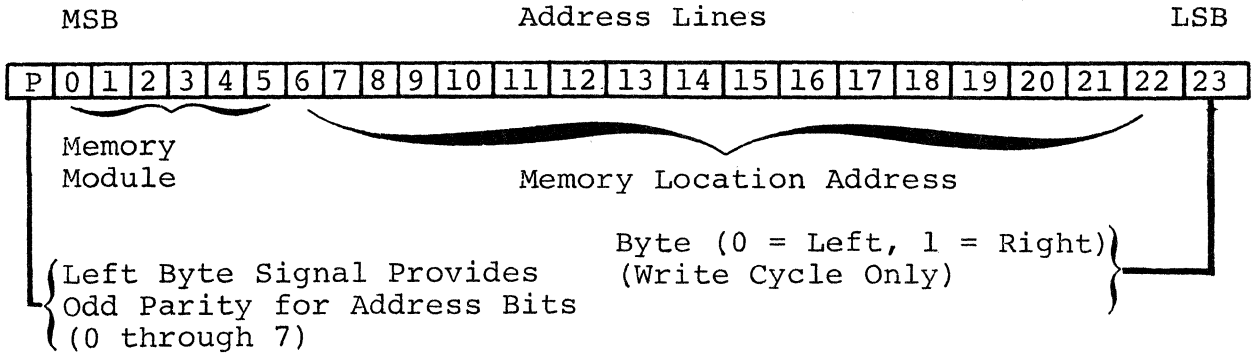
2.3 MEGABUS FORMATS

Figure 2-3 depicts the information format for the data and address lines during various Megabus cycle operations. Each format shown reflects the occurrence of a single Megabus cycle. The following subsections provide a description of the formatting associated with the address and data lines.

2.3.1 Memory Addressing

There are 24 address leads which can have either of two interpretations, depending on the state of an accompanying control lead called Memory Reference. When the Memory Reference signal is true, the Megabus enables 2^{24} bits to be directly addressed in memory.

When Memory Reference is false, memory is not being addressed, the units communicating on the Megabus are transferring control information, data, or interrupts, and they address each other by channel numbers, which are transferred across the address lines.



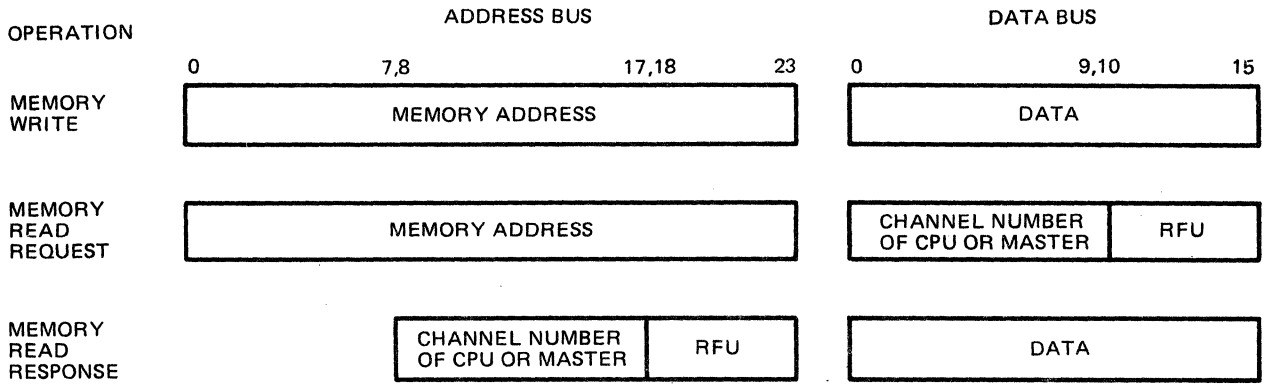


Figure 2-3 Megabus Formats

2.3.2 Memory Write Operation Parameters

The operand store and DMA write parameters consist of a memory address delivered across the address bus and accompanied by data across the data bus. The write operation addresses a particular location in a memory controller and writes either a byte (8 bits) or a word (16 bits) of data into that location. When addressed by some master controller, memory becomes the slave unit and completes the write operation in one Megabus cycle.

2.3.3 Memory Read Operation Parameters

When a Read operation is requested from memory, the master-to-slave communications require the initiation of a response (second Megabus cycle) to complete the operation. The instruction fetch, operand fetch, and DMA read parameters for a Memory Read operation consist of a memory address delivered across the address lines and the requesting unit's channel number delivered across the data lines during the first Megabus cycle. When memory responds (second Megabus cycle), memory becomes the master unit, initiates the response, and delivers the parameters (channel number of the previously requesting unit) across the address lines and information (data) across the data lines to the requesting unit which completes the operation. The duration of this operation depends on slave response time; therefore, the intervening time between the Megabus cycles can be used by two other units, which are not involved in the interchange. If double-word fetch is used, a third read bus cycle is needed (same as second bus cycle).

III
HARDWARE OPERATION

The material contained in this section provides a detailed hardware description for the Level 6 single-word and double-word fetch main memory subsystem. The Series 60 Level 6 CMC9009/CMC9010 Memory Reference Manual (Document Number 71010279-100, Order Number FQ37) provides a functional map for each logic integrated circuit described in this section. These maps can be used to determine the configuration of each integrated circuit location and should be referenced as needed.

3.1 TIMING GENERATION

The timing generator develops and distributes the timing pulses required for successful memory operation through delay line circuits and the associated logic.

The timing generator is started either by a My Acknowledge signal, indicating the start of a Memory cycle requested by another controller, or by a Refresh Command signal, indicating the start of a Refresh cycle. Once started, a series of pulses is distributed to generate all of the necessary timing functions within the memory.

At the center of the timing generator are two 200-nanosecond delay lines driven by a four-way AND/OR input driver (see Figure 3-1). An active cycle is started by function My Acknowledge coming true, which signifies the acceptance by memory of a bus cycle request. The positive-going edge of My Acknowledge propagates the Delay Line Input function. Delay Line Input is returned to one input of the four-wide AND/OR gate, latching the input. After latching (approximately 25 nanoseconds), the state of My Acknowledge is not critical for timing signal generation.

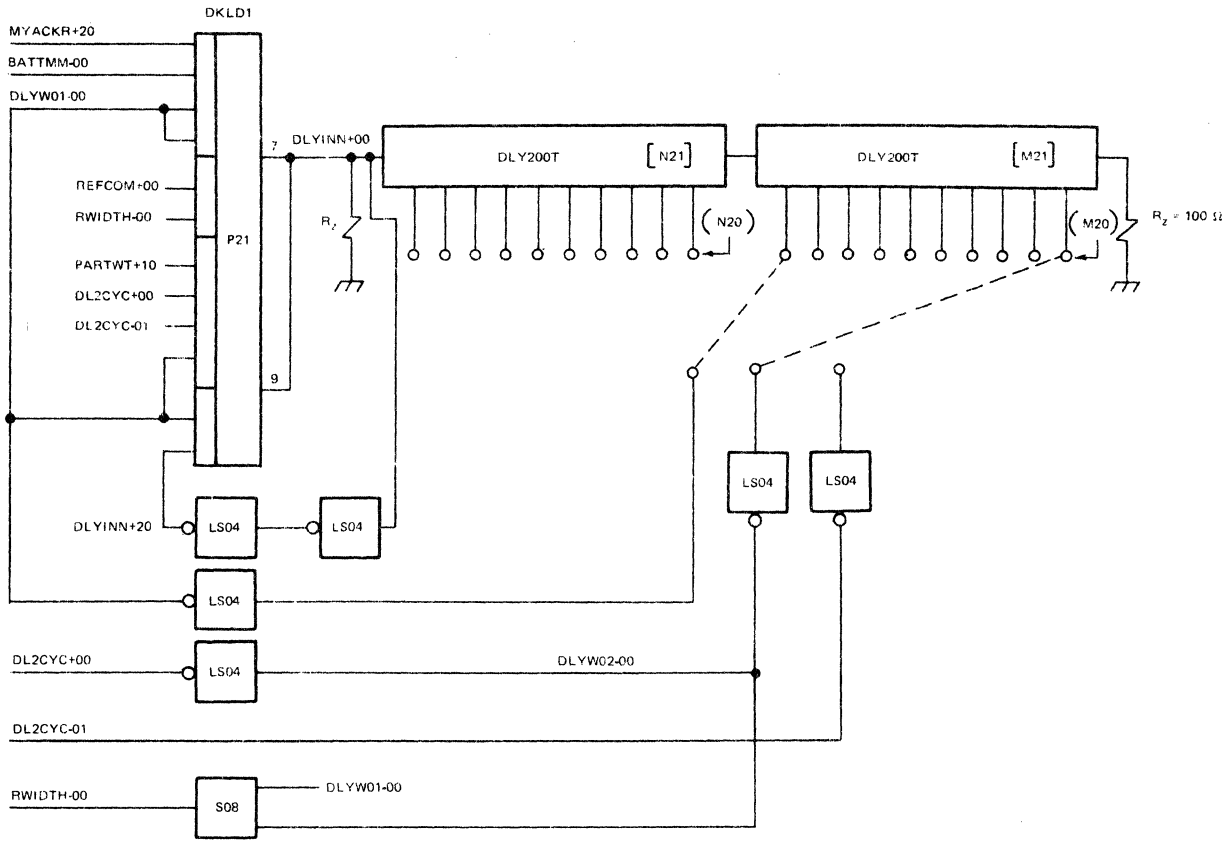


Figure 3-1 Delay Line Timing

Function DLYW01- is fed back to the input gating structure to shut off the Delay Line Input function, thus determining the pulse width. DLYW01- is tapped from the second 200-nanosecond delay line and adjusted so that the Delay Line Input function is approximately 200 nanoseconds in duration. This signal is then tapped through the delay lines to generate the required timing functions.

Figure 3-2 is a timing diagram, showing the relationships between the timing signals when an active cycle is started from a My Acknowledge signal. Figure 3-3 shows the timing signal relationships when a Refresh Command signal initiates a timing cycle.

For each cycle, Normal and Late timing signals of approximately 300 nanoseconds in duration are developed to provide each memory stack with an appropriate clock signal. Each signal is developed with control over the leading and trailing edges. Adjustment of the edges is achieved by a method of Fixed and Variable timing taps located in the N20 and M20 areas of the memory controller board (see Figure 3-4). Two groups of taps, which allow 12 functions to be adjusted, are provided. One end designated as Fixed is etched to a function which requires adjustment. Usually, a positive AND gate or a positive NAND gate is used as a buffer between the function and the delay line circuit. Twenty Variable taps are provided

which can be wired to the Fixed end via a wire-tap pin. The Variable ends are taps off the two 200-nanosecond delay lines and provide 20-nanosecond increments for the functions that require adjustment. Thus, a function can be set up by wire wrapping a connection between the Fixed tap and the appropriate Variable end terminals. The buffer circuits are used to provide small current drain for each function that is attached to the delay line.

Memory devices used with the 128K double-word fetch main memory controller require two clock signals to initiate a memory cycle. Since the controller is capable of supplying the requester with a double data word for each read request, two independent stacks are used, each capable of an active memory cycle slightly skewed or phased from the other. The phasing results from the time necessary to derive the address of the second request location after an Acknowledge signal is issued. Accordingly, once the timing generator is started, two sets of clocking signals are generated; one set is considered as Normal and the other set is considered as Late. These signals are designated MRASSN+00 and MCASSN+00 as Normal, and MRASSL+00 and MCASSL+00 as Late. Each of these signals (see Figure 3-5) has leading edge and trailing edge adjustment controls. Once the Normal and Late clock signals are generated, they are fed to a steering network (see Figure 3-6) where it is decided which stack will be fed with the Normal timing pair and which stack will be fed with the Late timing pair. The steering control logic takes the raw timing pulses and directs them as a function of MLADX2+, which is the off/even address bit latched in an independent register. If MLADX2+00 is a logical Zero, indicating an even location, then the two Normal timing pair signals will be directed into the even stack and the two Late timing pair signals will be directed into the odd memory stack. Conversely, if MLADX2+00 is a logical One, the Normal timing pair will be directed into the odd memory stack and the Late timing pair will be directed into the even memory stack. Also gated into the steering logic are the DFETCH+20 and DFETCH+00 functions, which enable the Late timing pair during Double-Word Fetch cycles. DFETCH+00 and DFETCH+20 enable MRASSL+ and MCASSL+ during Double-Word Fetch cycles, while DFETCH+20 is overridden during Hidden Refresh cycles and only allows MRASSL+00 to get through.

Two passes through the delay lines are needed in two types of memory operations; i.e., Byte Write and Memory Initialize. To perform a Byte Write operation, memory must execute an internal Read cycle followed by the Write cycle. Each cycle requires a separate pass through the delay lines. When memory is being initialized, the Refresh circuitry is used to time the Initialize cycles. After each Refresh cycle, an Initialize cycle is performed, writing Zeros in memory. Therefore, during Initialize modes a Second Pass cycle is needed every 15 microseconds.

During Byte Write (BYWRIT+ true) modes and Initialize (INITMM+ true) modes, function PARTSS- sets the Partial Write Timing (PARTWT+) function (see Figure 3-3). PARTWT+ remains high until the end of the First Pass cycle at which time functions DL2CYC+ and DL2CYC- generate another Delay Line Input signal on one AND gate of the four-way AND/OR input driver. Once the second Delay Line Input function is latched in the driver, the Partial Write Timing function

is reset by function DLYW02-, and DLYINN+ starts down the delay lines. Once the Second Pass cycle is started, another set of identical timing functions is generated. Second Pass cycles are only generated when Byte Write operations are required or when the memory subsystem is being initialized.

The Memory Busy signal is derived whenever the timing generator starts a cycle. The leading edge of Memory Busy is started when MYACKR+ is issued, and the trailing edge of MEMBUZ+ is contolled with a separate timing signal MEMBUZ-TE (see Figure 3-2). Three additional signals influence Memory Busy:

1. MRASSN+00, which preserves the duration of the pulse between the end of MYACKR+ and the start of MEMBUZ-TE.
2. DBCYCE-, which preserves the duration of Memory Busy between the end of MEMBUZ-TE of a first pass and MRASSN+00 of a second pass through the delay lines during a Byte Write or memory Initialize operation.
3. DWDEND+, which preserves the Memory Busy signal during Double-Word Fetch cycles between the first and second MYDCNN+ signals.

Data readout of memory on a memory read request must be latched in the data registers. Each stack (odd and even) is fed to the respective data registers, which are separately controlled by latching signals MDOECT+ and MDOOCT+ (see Figure 3-7). These signals are derived by gating MRASTE+ and MRASTO+ with inhibit signals which delete latching action during Refresh cycles (Normal or Hidden) or the second pass of a Byte Write cycle. MDOECT+ and MDOOCT+ are fed to two banks of local data registers.

DCNNGO-TE is used to clock a stored request in the bus request logic. Figure 3-7 shows the generation of function DCNNGO-TE. This signal also depends on the state of function INREDY- from the read/write control logic.

3.2 ADDRESS CONTROL

Level 6/40 128K memory controllers interface with the Megabus from which the address signals are obtained. There are 24 addresses accompanied by one parity signal on the Megabus. Each signal is buffered on the controller by a transceiver, the receiver output being the source of addresses for the controller. The bus addresses are separated through logic to select the appropriate memory module, Memory-Pac, RAM location, and byte control. Figures 3-8 and 3-9 show how addresses are used on the controller.

BSAD00+ through BSAD04+ and BSADP0, buffered with receiver circuits, are fed to the address compare logic where the value is compared with the value of the static switches. Proper module identification is determined with the setting of the static switch and an Address Go (MYADGO+) signal is derived. MYADGO+ is then used as one of five conditions needed to accept a request from the Megabus (see Figure 3-9). BSAD05+ is routed through the switch network from which BSADX0+ is derived and sent to the address compare logic. Thus, BSADX0+ becomes a variable function which can take on different values for desired memory controller applications.

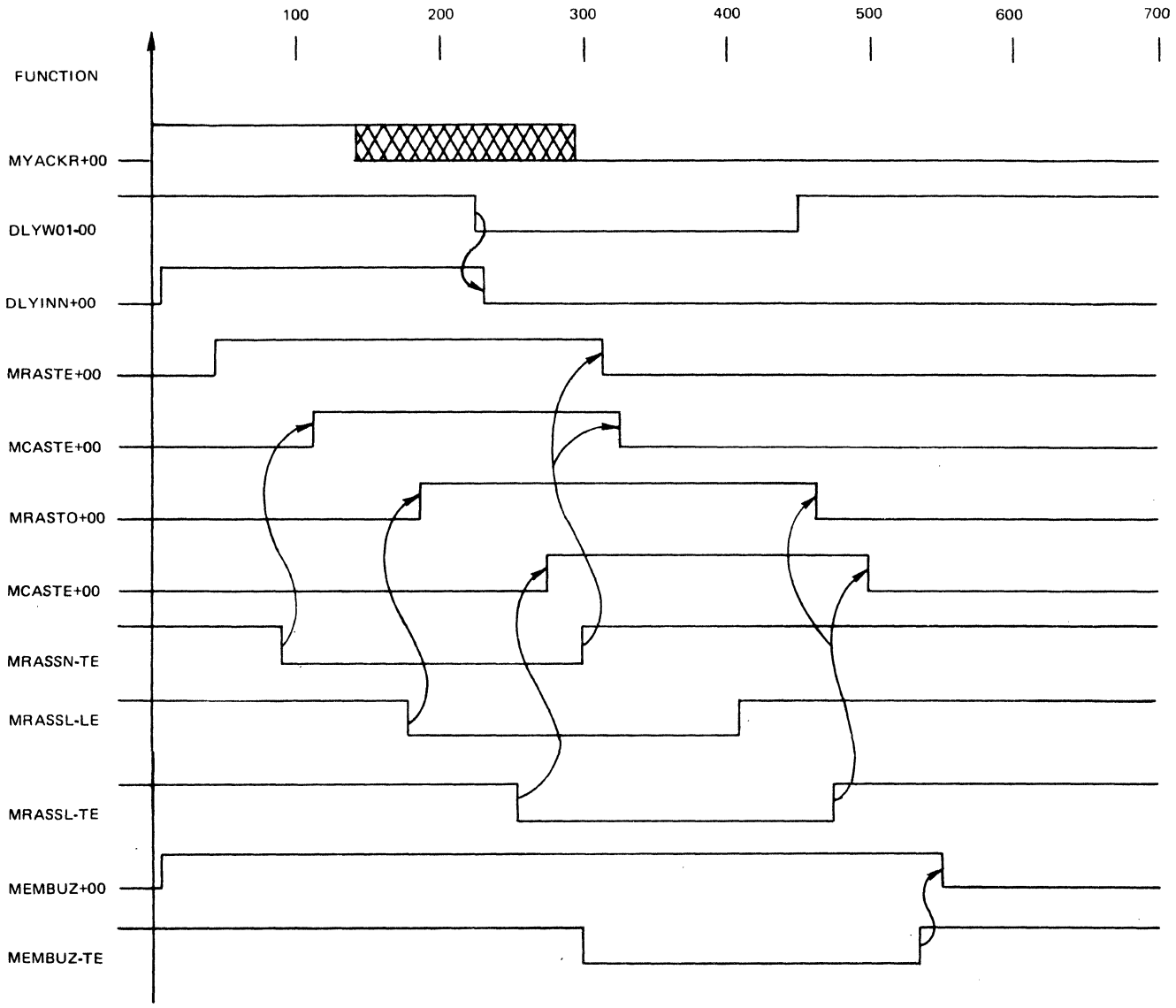


Figure 3-2 Timing Start from MYACKR+

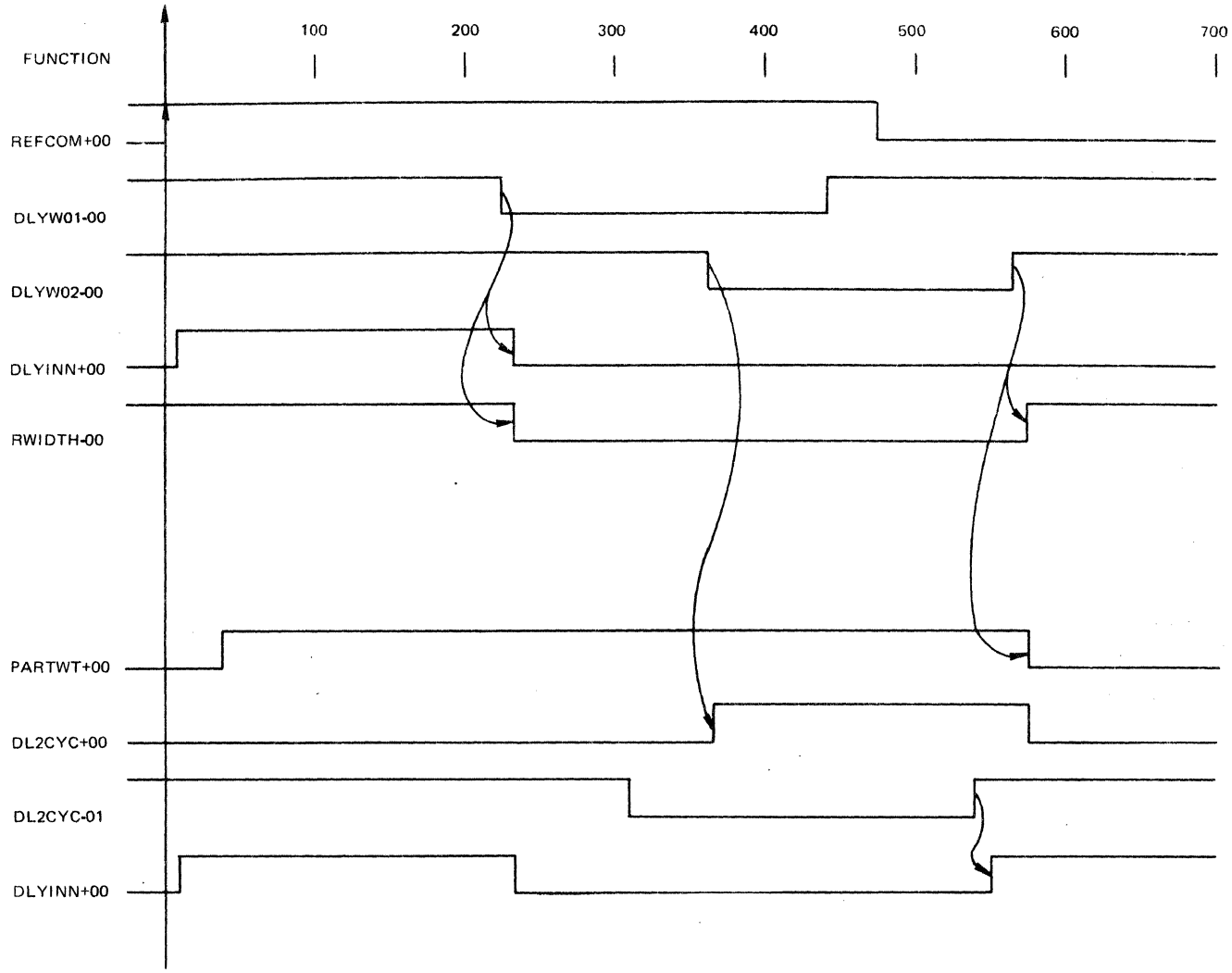


Figure 3-3 Timing Start Refresh and Initialize

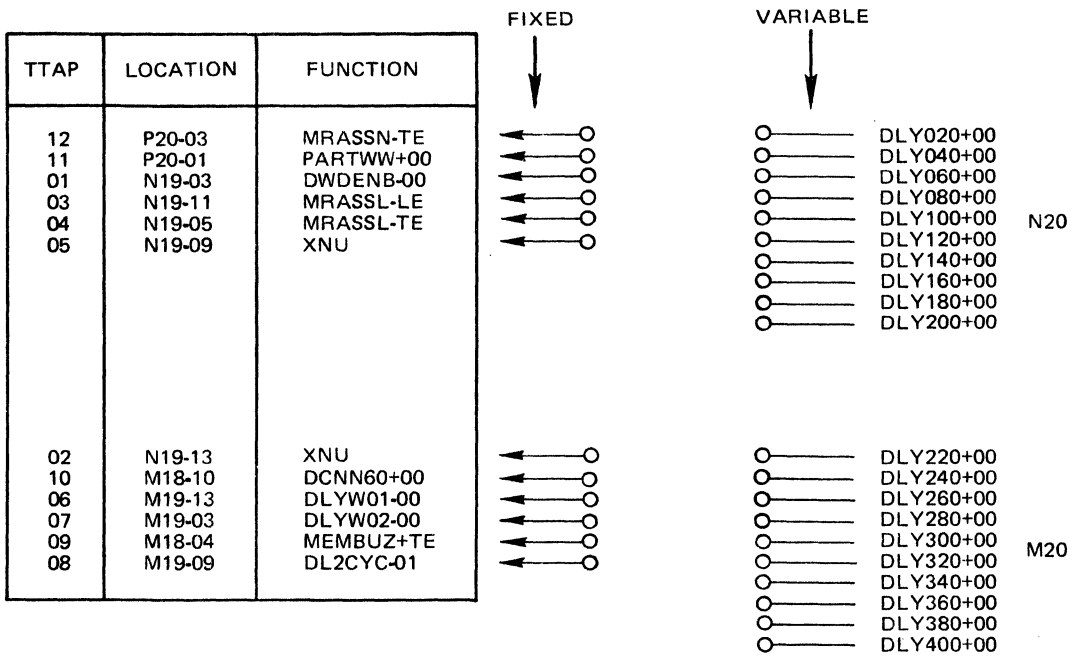


Figure 3-4 N20 and M20 Timing Setup Network

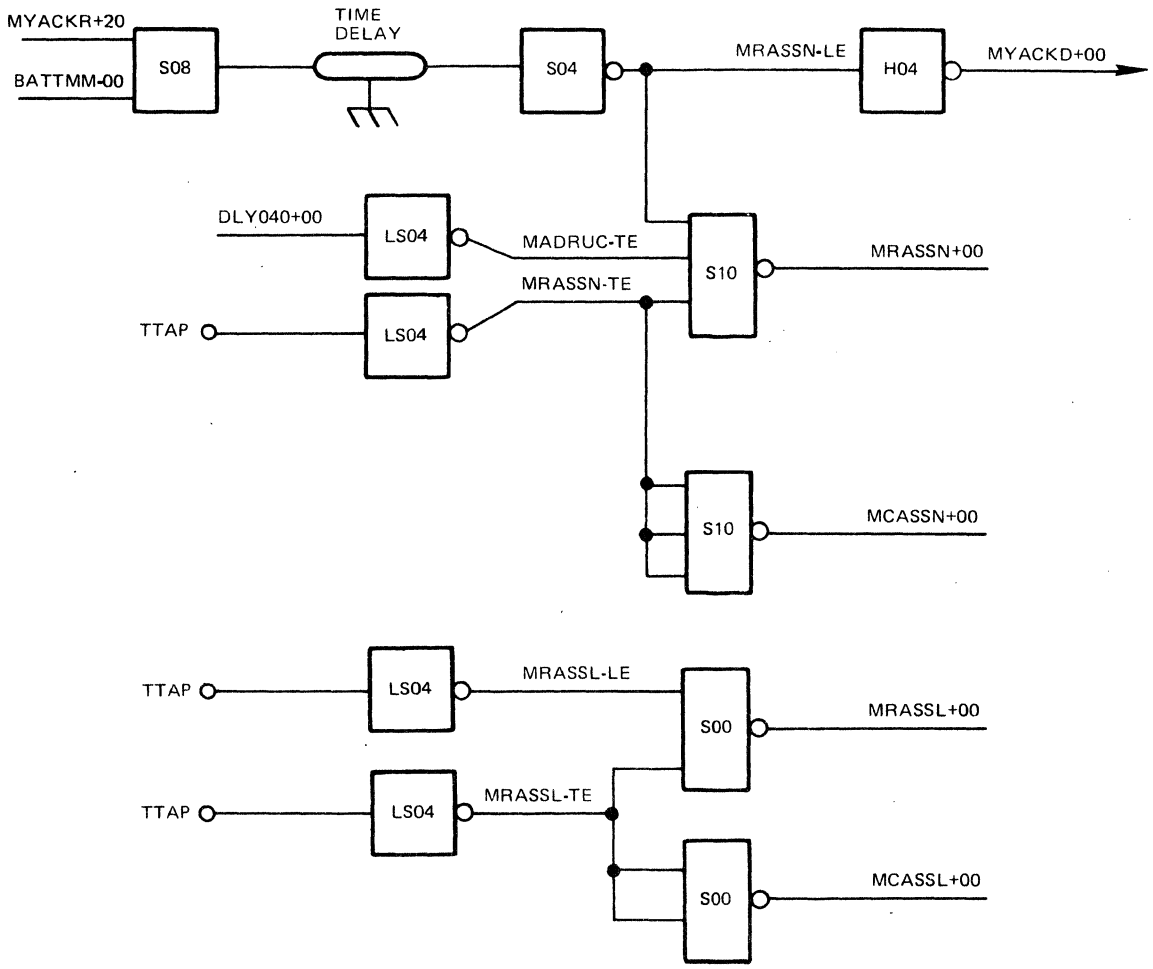


Figure 3-5 RAS/CAS Timing Pair

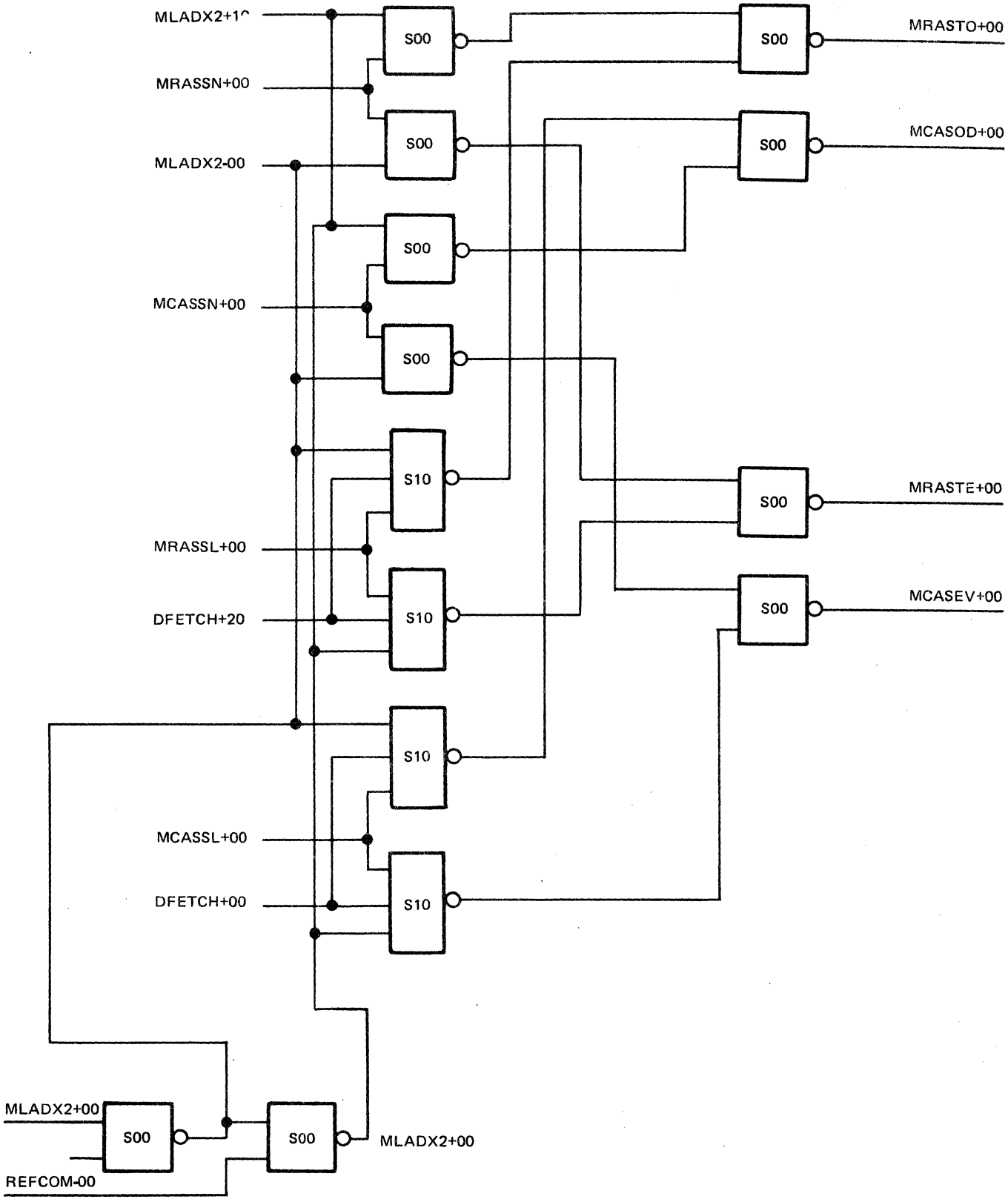


Figure 3-6 Steering Logic

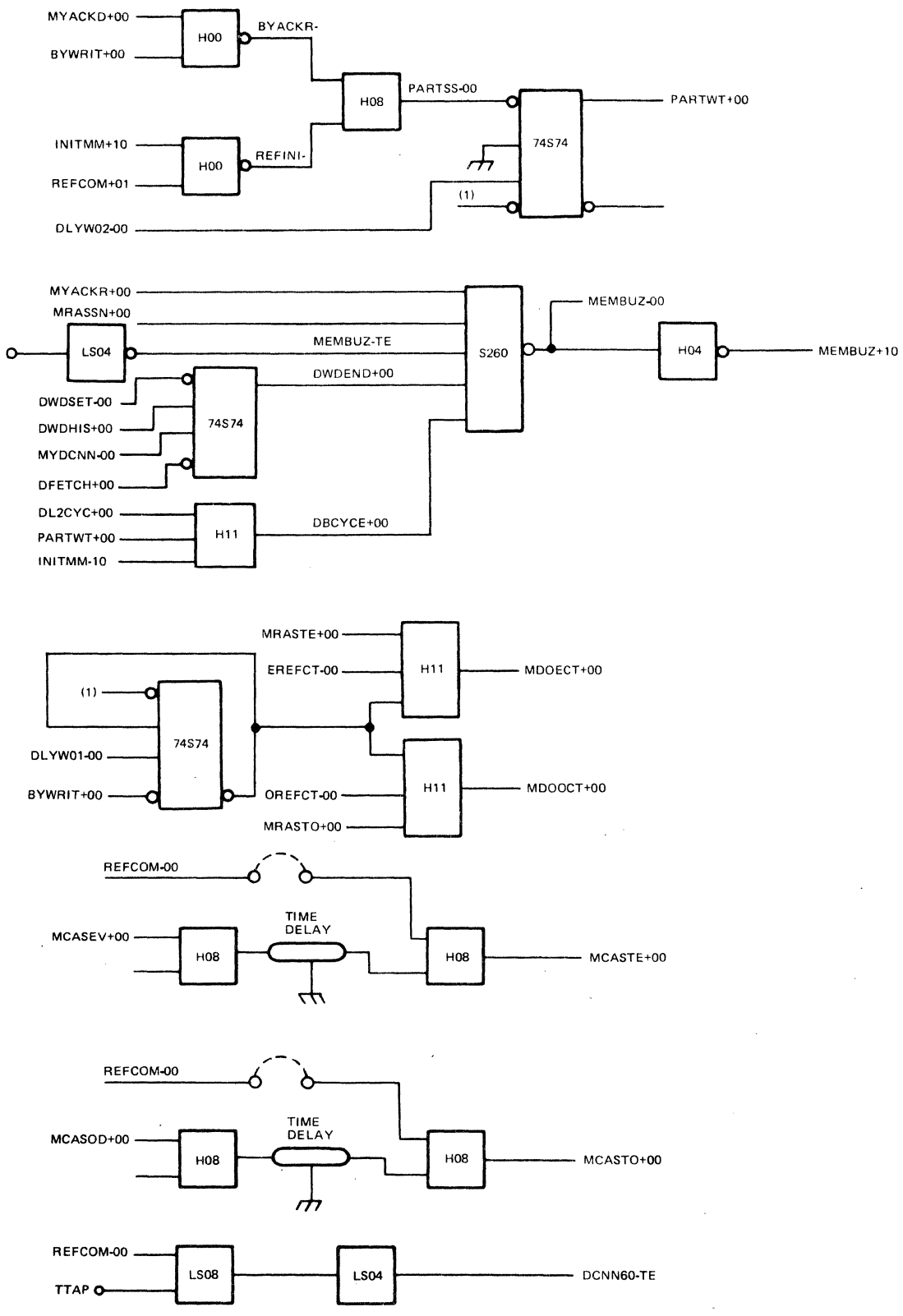
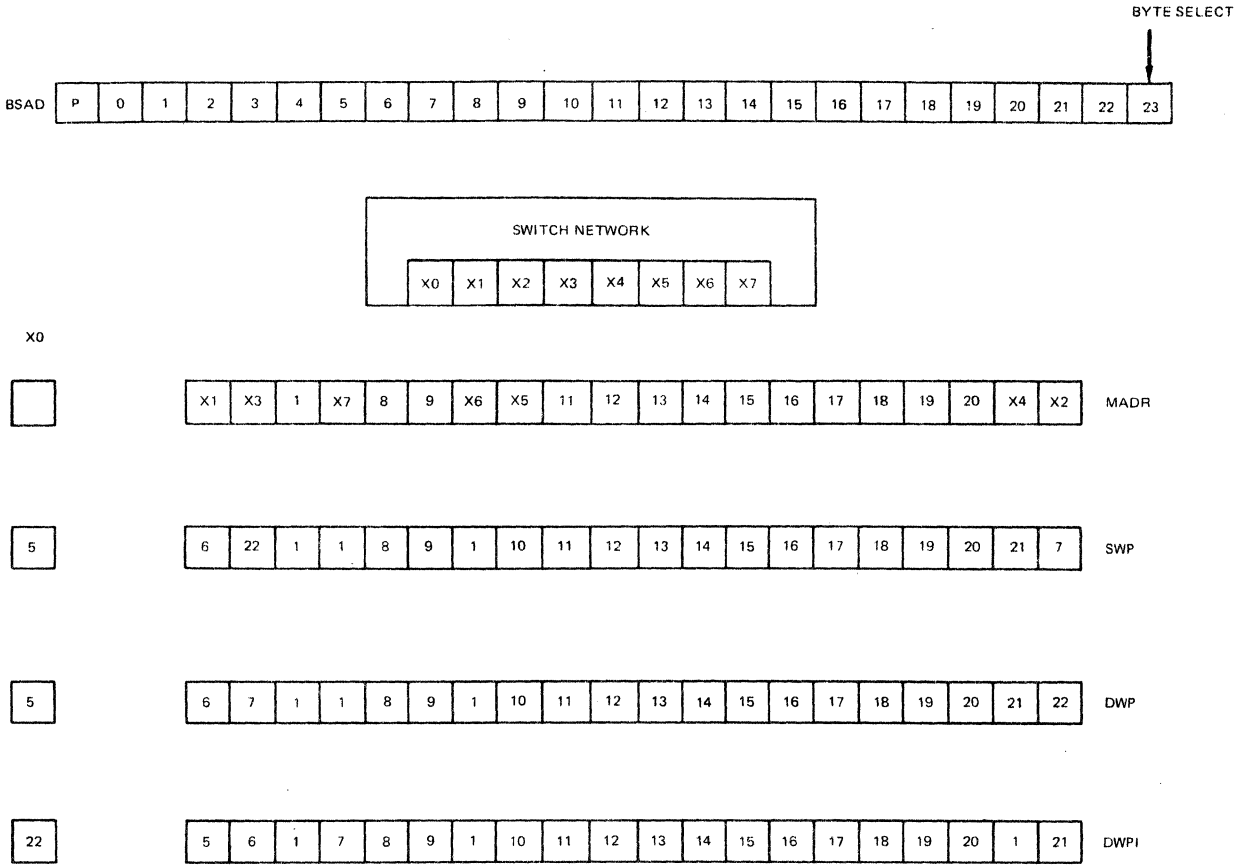


Figure 3-7 Miscellaneous Timing



X0 MODULE SELECT
X1 MEMORY--PAC SELECT
X2 ODD/EVEN STACK SELECT
X3 CLOCK DECODE
X4 MOS ADDRESS
X5 MOS ADDRESS
X6 ADDRESS REGISTER MODIFIER
X7 ADDRESS REGISTER MODIFIER

Figure 3-8 Switch Network

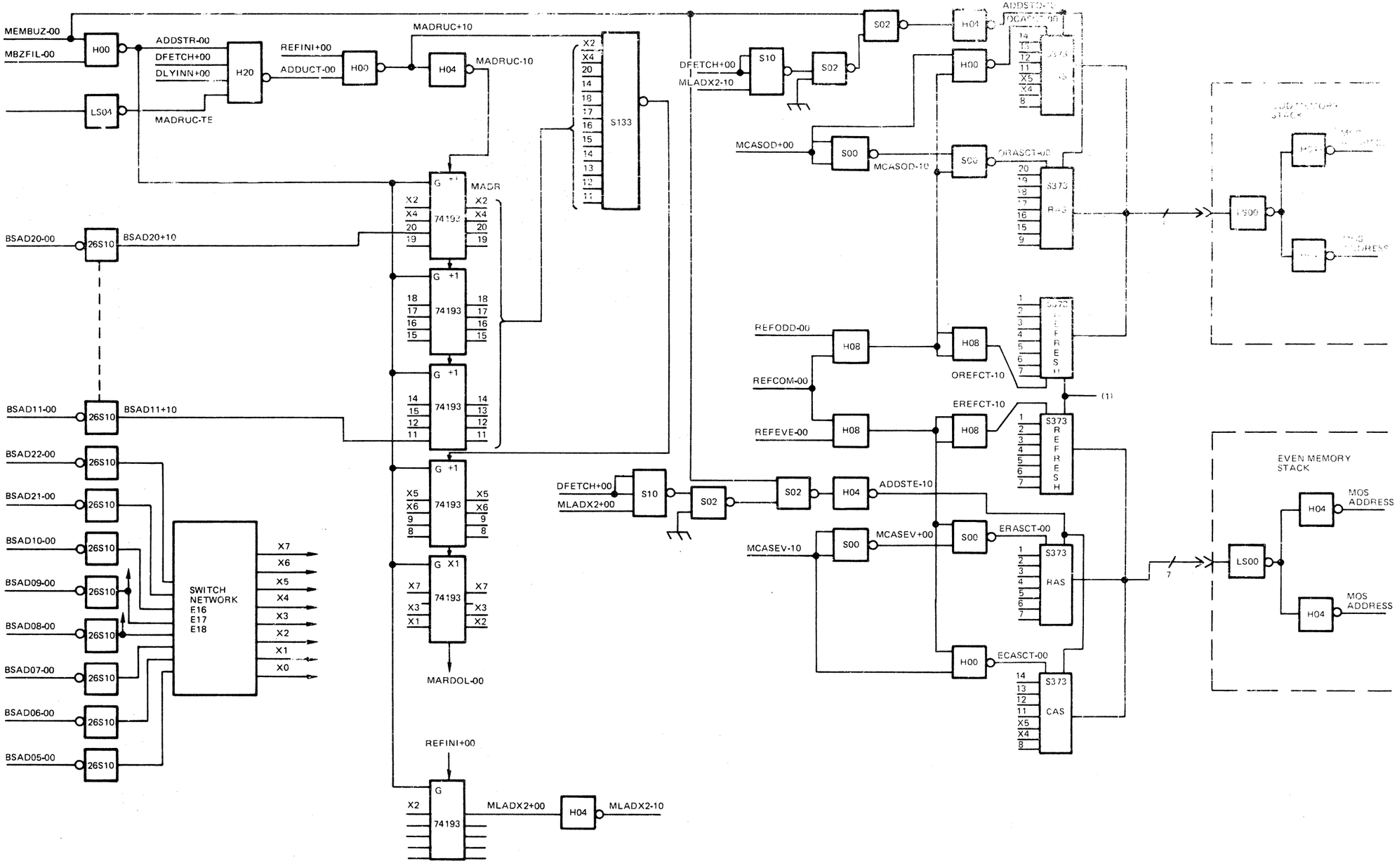


Figure 3-9 Addressing Circuitry

Figure 3-9 shows the majority of the addressing circuitry. The structure of the address path within the Level 6/40 128K memory controller is centered around a dual stack, each stack having an independent address circuit. The independent address path is required because of the double-word fetch nature of the controller where two words can be read out from one memory read. Essentially, the steps taken to provide independent addresses for each stack are as follows:

1. Latch the bus address in a primary address register on acceptance of a Memory cycle.
2. Allow the first address to be latched in an odd or even stack register, depending on the value of the odd/even address bit MLADX2+, while keeping the other register open for new addresses.
3. Increment the primary register so that its location will be the latched value plus one.
4. Allow the incremented value to be passed on to the odd or even secondary register.

From both of the secondary registers, the address bits are multiplexed to form the Row address and the Column address that are required by the MOS RAM. Refresh addresses also are multiplexed at this point. After multiplexing, the addresses are routed to the odd/even memory stacks where they are buffered and applied to the MOS RAM array.

3.2.1 Primary Registers

Five 4-bit registers/counters are connected to form the primary register circuit. These registers are loaded in parallel by ADDSTR-00, the inputs being taken from the 26S10 receiver outputs. Bus addresses 22 through 05 are used as the inputs. However, several bits are routed through a patch or switching network so that the value taken can be altered for different controller applications. Figure 3-9 shows the values that the primary address registers take for three memory controller applications. In addition to the parallel load function, the primary address registers must also be incremented on command. When a memory controller is set up to perform double-word fetch operations, the register is first loaded with the bus address, and then an increment pulse is applied so that the address value plus one is obtained. The increment pulse (MADRUC-) is a very narrow pulse that is developed off the delay line (40 nanoseconds); updating takes place on the positive-going edge. A Look Ahead Carry circuit is used to improve the ripple through time of the five counters. This circuit looks at the outputs of the first 12 bits (MADRX2+ through MADR11+) along with the increment pulse (MADRUC+), and provides an increment pulse for the last two registers whenever the count on the first three is all Ones.

One additional function of the primary address register is that of a counter. This function is used when the memory is in an Initialize mode and it becomes necessary to cycle through the address field and write Zeros into memory. When in the Initialize mode, function REFINI+ is used to increment the counter. REFINI+ is derived off the Refresh counter.

One address bit which requires special attention is the odd/even address bit, MLADX2+. MLADX2+ is used to steer the Normal and Late timing pair signals to the proper stacks (refer to subsection 3.1). The requirements for MLADX2+ are as follows:

1. MLADX2+ must be latched on the positive edge of MEMBUZ+.
2. There must be no change during Refresh.
3. There must be no change during BSDCNN-, Double-Pull cycle.
4. MLADX2+ must not be incremented on a Double-Pull cycle.
5. MLADX2+ must be incremented during Initialize.

3.2.2 Secondary Address Register

Each stack (odd and even) has an independent set of secondary address registers. Each set is comprised of three octal D-type transparent latches with the outputs wired or tied to form the multiplexed addresses. The Row, Column, and Refresh addresses are combined at this point. Inputs to the octal latches are derived from the primary address outputs. Each latch has two control pins. The output tri-state control, shown as (1) on Figure 3-9, is used to switch from the contents of one register to the contents of another register. With this control, the Row Address (RAS) and the Column Address (CAS) are multiplexed by MCASEV+ or MCASOD+ to assure that the proper address setup and hold parameters are obtained. Also, when it becomes time to Refresh (Normal or Hidden), switching in the Refresh addresses occurs. The operation of the tri-state control is independent of the latching action. The RAS and CAS addresses are latched differently, depending on the type of cycle. In Single-Word Pull cycles, the secondary address registers latch on the same positive edge as the primary address register (MEMBUZ+). Thus, during Single-Word Fetch cycles, the addresses flow into two wide open registers and on to the Memory-Pacs. When the memory controller accepts a cycle, the addresses are latched and held until the end of the cycle.

Address latching during Double-Word Fetch cycles is slightly different. First, the primary address register (MADRXX) latches the contents of the bus address and passes it on to the secondary address registers. Depending on the value of MLADX2+, one of the secondary address registers latches and holds its contents. Second, the primary address register is incremented to obtain the address value of the second word to be read. Once the register is incremented, the value is held until the end of the cycle. Third, one secondary register is kept wide open to pass the incremented value on to the Memory-Pac.

Latching of the Refresh address is not used in the secondary register because its input is obtained from binary counters which hold their value throughout the cycle.

3.2.3 Address Decoding

Decode signals for the Memory-Pacs are derived from the circuits shown in Figure 3-10. Registers for the MADEX3+, MADEX1+, MADOX3+, and MADOX1+ signals are similar to the secondary registers described in subsection 3.5.2 because they hold the first address in one register while allowing the incremented value to pass in the other register. The decode bits pass to the 74139 2-to-4 decoders, which pick one out of four decode signals on each stack. These decode signals (DRASX0 through DRASX3) are fed to the Memory-Pacs where they gate MRASTE+ and MRASTO+ to pick one row out of four on the Memory-Pacs. The decode signals are overridden during a Refresh cycle to turn on all of the clock signals.

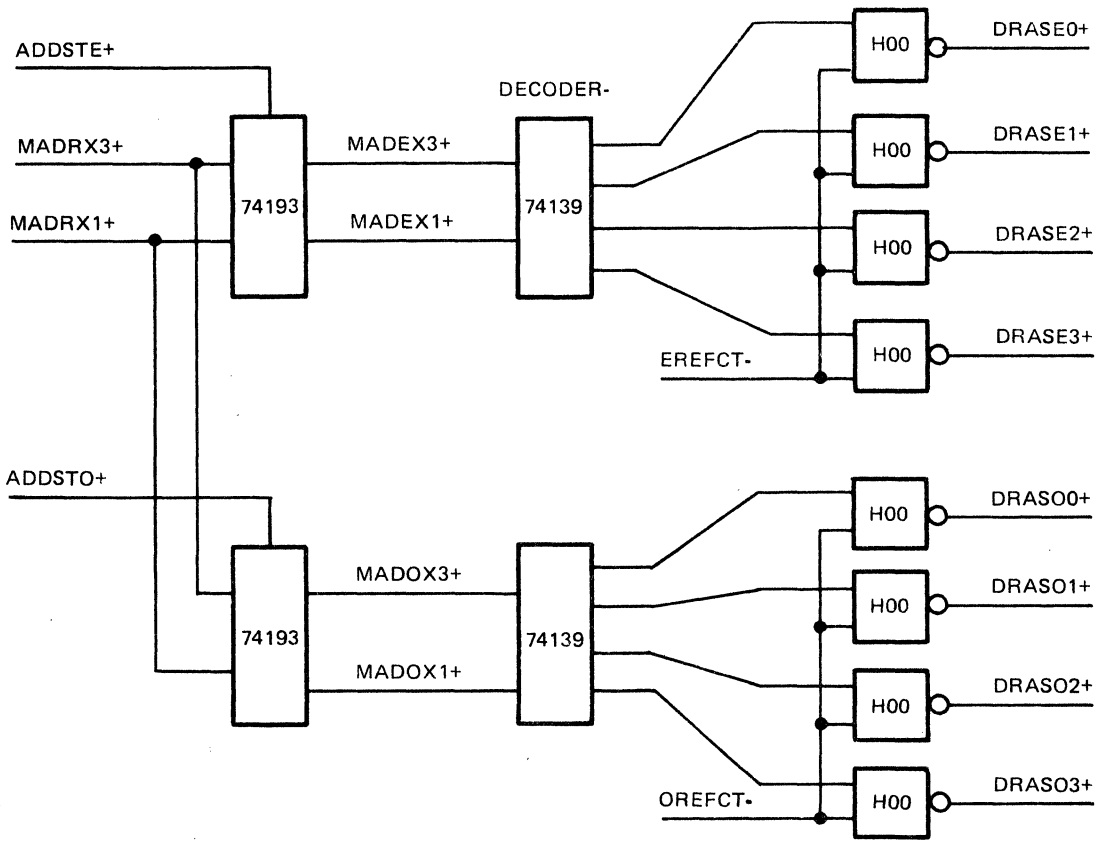


Figure 3-10 Clock Decode Logic

3.3 MEMORY INITIALIZATION

Memory initialization is an internally actuated feature of the double-word fetch memory controller that is used to set or clear the memory contents to a specific state on initial system power up. Subsequent initialization sequences are determined by the presence of the Memory Save Power Supply option (refer to the Power System Manual, Document No. 71010290-400, Order No. FL34, and to subsection 3.11). If the Memory Save Power Supply option is not resident in the system, subsequent memory initialization sequences occur whenever the system is powered up. For systems with the Memory Save Power Supply option, initialization occurs only on the first power-up sequence. On subsequent power ups, memory does not initialize and is available for bus cycles only after it has received the trailing edge of the Master Clear signal. Once memory initialization starts, Zeros are written into the data locations, and proper parity is written into the parity locations. The following description of initialization assumes that this is either the initial power-up sequence or that the Memory Save Power Supply option is not present in the operating system.

Figure 3-11 shows the memory initialization circuitry. The positive edge of Megabus Power On (BSPWON+) sets PWONLL+ via a standard D-type flip-flop. As soon as power is valid, Refresh Command (REFCOM+) cycles start. On the next Refresh cycle, Memory Power On (MYPWON+) is valid. MYPWON+ sets the Clear Memory (MYCLRR+) signal, resetting the address registers to all Zeros. Memory Power On also sets the Initialize (INITMM) flip-flop, which enables the address register to act as a counter to increment through all address locations. When the address counter reaches the highest location, MADROL-00 is issued to reset INITMM, halting initialization.

A Memory Save Power Supply option that contains a battery-operated power supply, providing +12 vdc and +5 vdc to the memory controller, is available for Refresh control to protect the contents of memory during power outages and brownouts. Battery power routed to some of the timing, refresh, and control circuits enables the MOS/RAM integrated circuits to be refreshed when system power is lost. Figure 3-12 shows the circuitry that enables each column of memory to be refreshed every 2 milliseconds.

When a power loss is detected, the Bus Power On (BSPWON+) signal goes low, and the memory initiates a Battery Shutdown (BATTSD-) signal. The BATTSD- signal is gated to the timing generator, which allows a maximum of 1.5 milliseconds of Megabus activity before placing the memory into the Protect mode. The 1.5-millisecond grace period is required to allow the CPU time to write the necessary information into memory for restoring a program. After the grace period, the memory controller refreshes its contents, ignoring all activities on the Megabus. When Bus Power On (BSPWON+) is valid again, the CPU issues a Bus Master Clear (BSMCLR-) signal. The memory controller becomes available for Megabus cycles on the trailing edge of Bus Memory Clear (BSMCLR-).

For more information that pertains to the Memory Save Power Supply option, refer to the Power System Manual, Document No. 71010290-400, Order No. FL34.

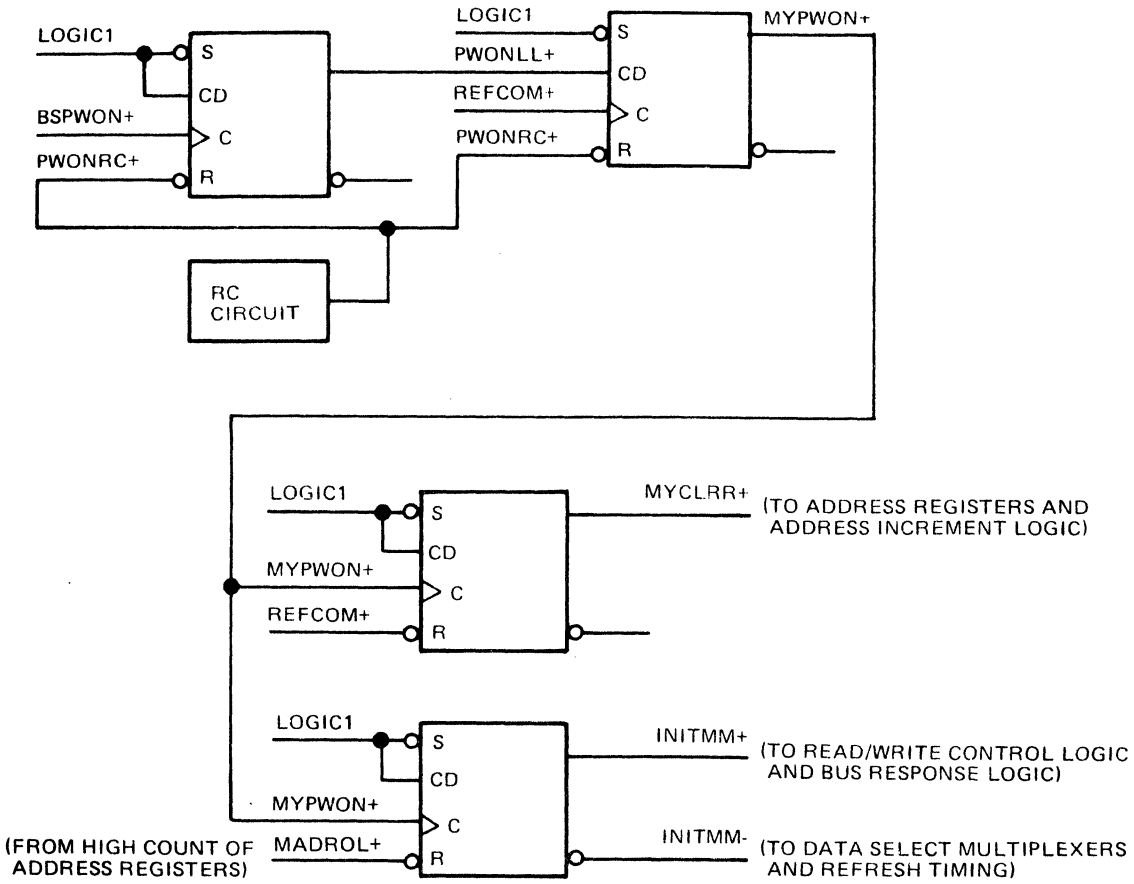


Figure 3-11 Initialize Logic

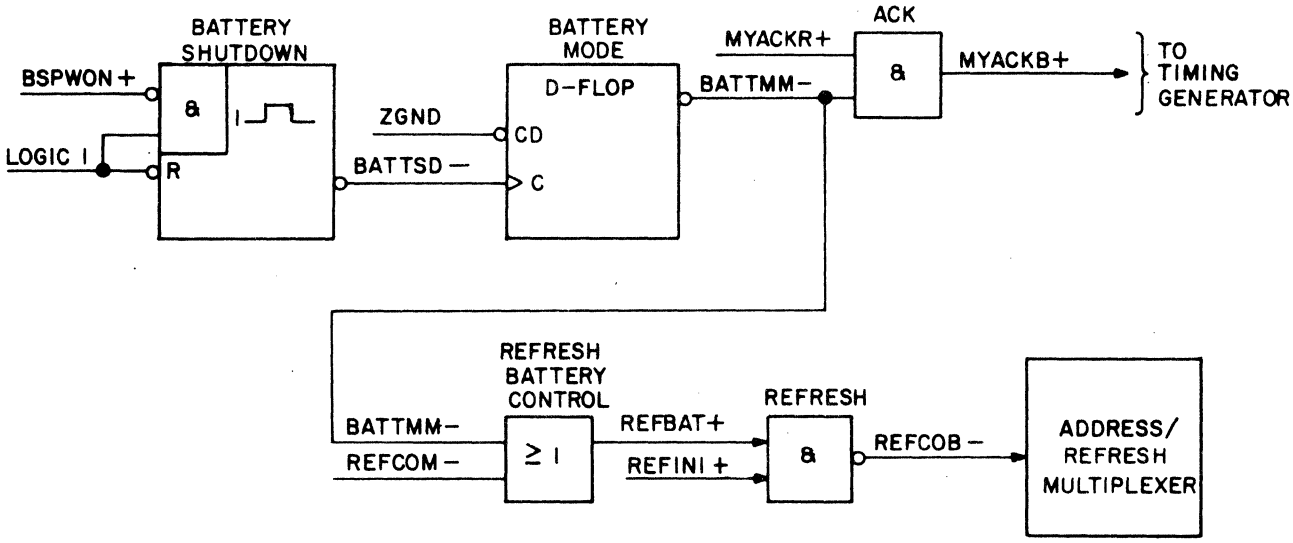


Figure 3-12 Memory Protection Circuit

3.4 READ/WRITE CONTROL

Read, Write, and Byte commands are transferred between the bus and the addressed locations in memory via the read/write control logic (see Figure 3-13).

Four bus signals (BSBYTE, BSWRIT, BSAD23, and BSDBPL) are latched in the read/write control register on the leading edge of the Acknowledge (MYACKR+) response and are used to control data selection logic, Read mode lines, and a portion of the timing generator.

3.4.1 Word Mode

To perform a Write operation in Word mode, Bus Write (BSWRIT+) is latched true and Bus Byte (BSBYTE+) is latched false at the control register. The result is that the read/write memory control lines (MRREAD+ and MLREAD+) are false due to the Read Command (READCM-) signal being false, enabling the memory chips to receive data. The left and right byte select lines (LEFTBY+ and RITEBY+) are both forced high due to Byte Mode (BYTEMM+) being false, enabling new data to be available for a memory location from the left (MDIR00+ through MDIR07+ and MDIRP0+) and right (MDIR08+ through MDIR15+ and MDIRP8+) data multiplexers (refer to subsection 3.6).

A Read operation is performed in Word mode by latching Bus Write (BSWRIT+) and Bus Byte (BSBYTE+) false at the control register. Through the Read/Initialize (READMI+) gate, the Read Command (READCM-) signal is enabled, setting the read/write memory control lines (MRREAD+ and MLREAD+) true and enabling the reading of a memory location. The memory output is delivered to the system bus.

3.4.2 Byte Mode

When writing in Byte mode, the major decision is which byte (left or right) is to receive new data. This decision is determined by the state of the MYLEFT function generated in the read/write control register from BSAD23+. Table 3-1 lists the data selected by the states of BYTEMM+ and MYLEFT-. For example, if BSWRIT+, BSBYTE+, and BSAD23+ are all latched true on the control register, the Left Byte Select (LEFTBY+) signal is false and the Right Byte Select (RITEBY+) signal is true at the input to the data multiplexers. This enables new data (MDIR08+ through MDIR15+ and MDIRP8+) to be available for the right memory byte and recirculated data (MDOC00+ through MDOC07+ and MDOCP0+) to be available for the left memory byte. If Address Bit 23 (BSAD23+) is latched false at the control register, the opposite bytes receive the new and recirculated data.

To have recirculated data available at the data multiplexers, a read of the memory location must first be initiated before writing new data into the left or right byte. For any Byte Write command, memory is forced into a Read cycle prior to initiating the Write cycle. This is initiated by the Byte Write (BYWRIT+) signal, allowing the Partial Write (PARTWT+) function, which enables the Read Command (READCM-) signal. The read/write memory control lines (MRREAD+ and MLREAD+) are forced into a Read state, which allows the memory location to be read and recirculated to the input of the

data multiplexers. The selection of which bytes receive new data and which bytes receive recirculated data is determined as stated above. The read/write memory control lines switch to a Write state when the delayed Partial Write (PARTWW+) signal resets, permitting data to be written into the memory location. Memory enters the Read mode as specified in the Word mode description (refer to subsection 3.6.1) and delivers a full word to the bus when requested. If only one byte is wanted, the requesting controller selects the specific byte.

NOTE

The Read Command signal is also affected by the Refresh Command (REFCOM+), which forces a Read mode during Refresh cycles, and the Memory Busy (MEMBUZ-) signal, which forces a Read condition when memory is not being used.

When true, BSDBPL+, the fourth command latched in the read/write control register, causes the memory to perform a Double-Word Fetch operation, sending out two successive memory locations. The Double-Word Fetch command is routed to the bus control and timing areas (refer to subsection 3.8.4.1.4). Double-Word Fetch cycles are only initiated during Memory Read cycles, and are not initiated during Refresh or Initialize operations.

3.5 REFRESH CONTROL

One characteristic common to all dynamic MOS memory system controllers is an area of logic dedicated to performing Refresh operations on the MOS RAMs. Generally, this logic determines the frequency of Refresh interrupt and synchronizes the Refresh cycle with a memory request so as not to cause interference. Refresh control also determines what location to Refresh, and sometimes provides the substrate bias voltage for the MOS RAMs. In the Level 6/40 128K memory controller, the Refresh logic is divided into two parts: (1) the logic associated with the Normal or Cycle-Steal type Refresh operations, and (2) the logic associated with Hidden Refresh operations.

3.5.1 Refresh Control (Normal or Cycle-Stealing Method, See Figure 3-14)

In dynamic MOS cells, data is stored in the form of a charge on the node capacitors. If the stored charge on the storage node capacitor is not periodically restored, the data is lost. To retain data, each cell must be refreshed at a minimum rate of 2 milliseconds. A normal Read or Write cycle refreshes a column of cells within the selected chip. The 16K MOS chip requires 128 cycles to refresh the entire chip. Because the unit is a RAM, special control logic must be included to ensure that all cells are refreshed within the required 2 milliseconds. The Refresh control logic consists of an oscillator, an address counter, and decision logic which interrupts memory activity and refreshes locations without destroying memory data or operations.

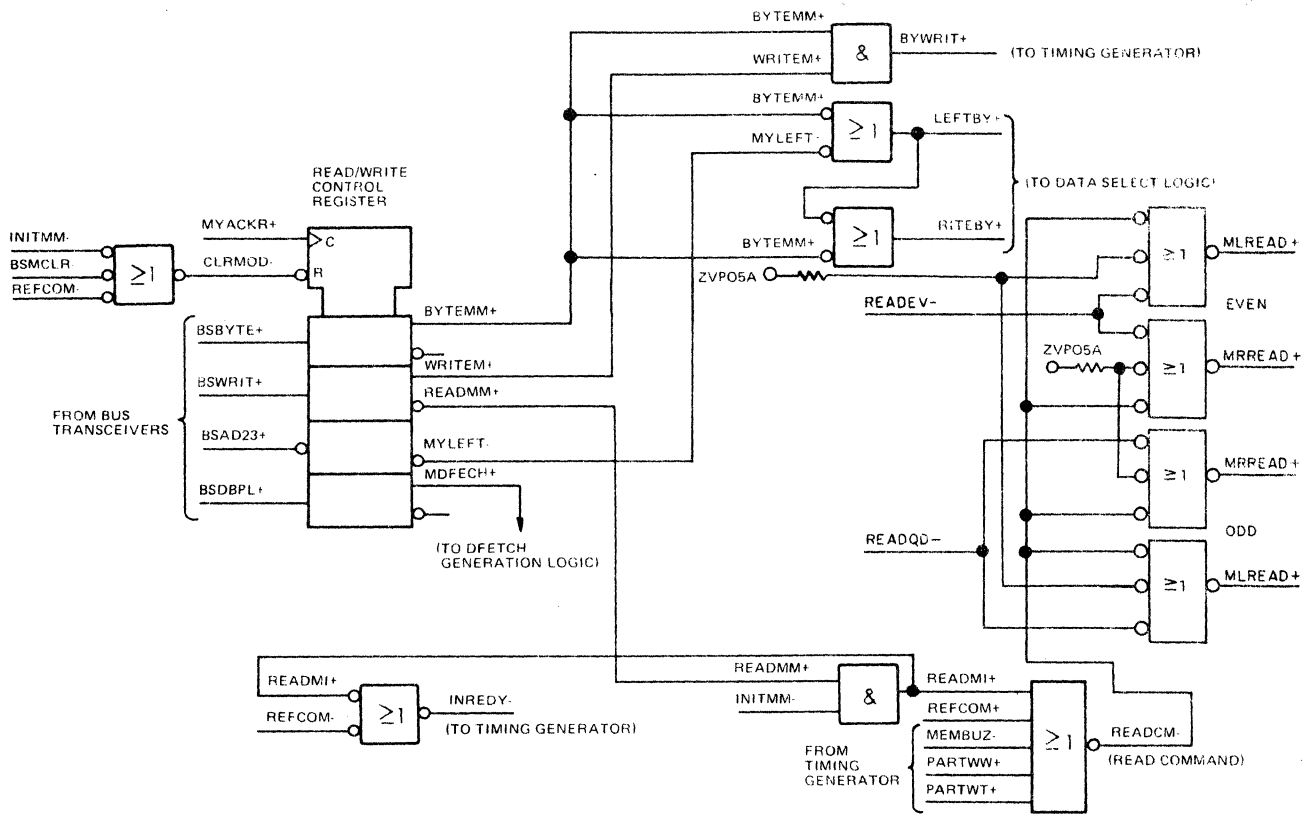


Figure 3-13 Read/Write Control Logic

Table 3-1 Left/Right Byte – New/Recirculated Data Selection

BYTEMM+	MYLEFT-	LEFTBY+	RITEBY+	LEFT BYTE	RIGHT BYTE
0	X	1	1	MDIR00+ through MDIR07+	MDIR08+ through MDIR15+
1	1	0	1	MDOC00+ through MDOC07+	MDIR08+ through MDIR15+
1	0	1	0	MDIR00+ through MDIR07+	MDOC08+ through MDOC15+

X = don't care.

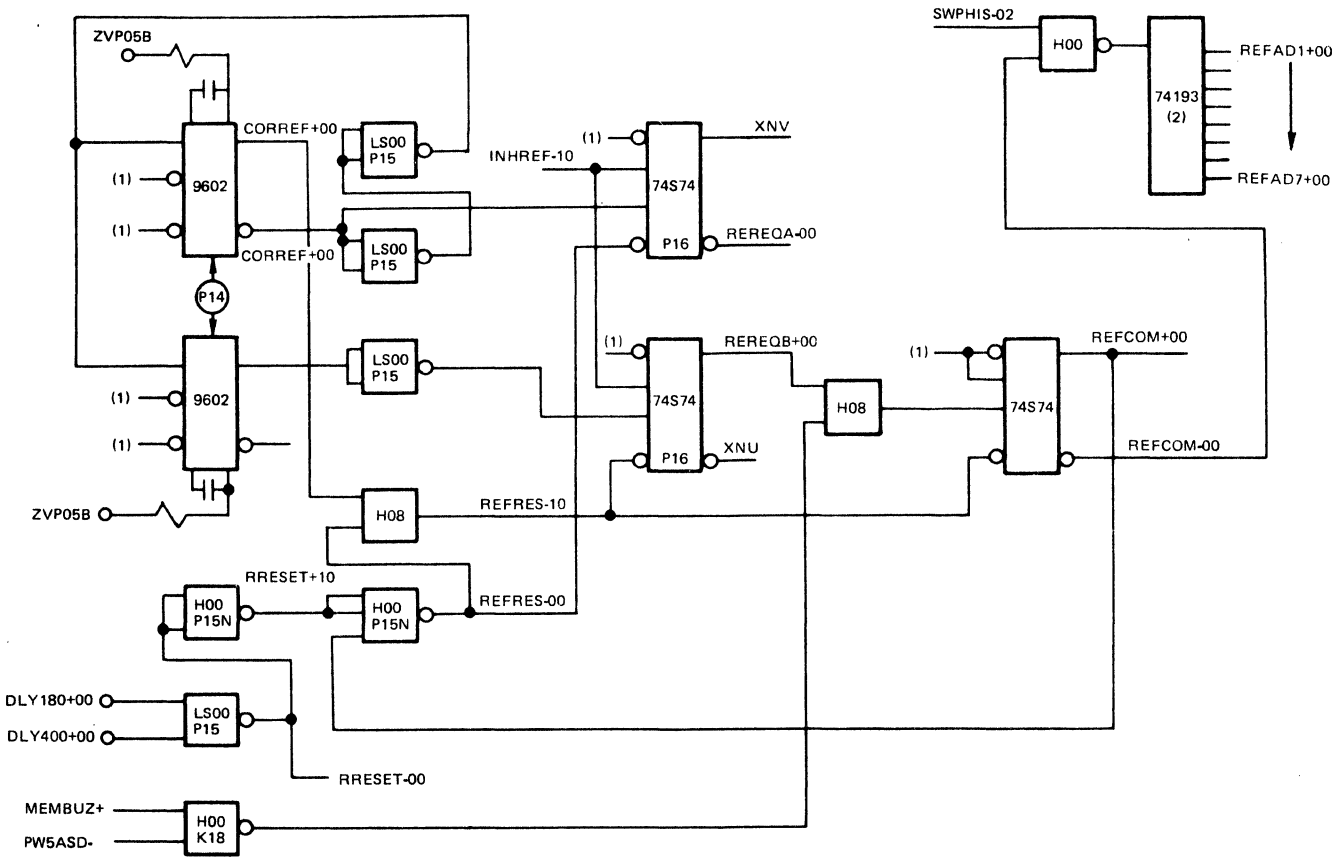


Figure 3-14 Refresh (Normal) Logic

The Refresh logic includes a means of substituting a Refresh cycle for a bus originated cycle and a Refresh address counter to ensure that all 128 columns are selected within 2 milliseconds. During a Refresh cycle, the memory is put into an internal Read mode. All clock decoding is overridden, and the same column of every chip of the module is refreshed.

Refresh cycles are distributed through the 2-millisecond interval and, therefore, occur approximately every 15 microseconds. The Refresh oscillator located on the memory runs asynchronously to activity on the bus. Every 15 microseconds at Refresh Time, an Early Refresh Indication is set, which indicates a Must Refresh condition and causes a Wait response to all future requesting master units. However, a master unit might have received an Acknowledge response just prior to this signal. An Early Refresh Request signal is generated to determine if memory is busy. If memory is not busy (i.e., not in the process of a cycle, not anticipating a cycle, or not acquiring a cycle when the Early Refresh Indication is set), then a Refresh cycle is started. If memory is busy, a Refresh cycle is started as soon as the current cycle is completed. After the Refresh cycle is completed, the next requesting master unit is acknowledged. The process repeats 15 microseconds later when the next Refresh cycle is required.

Figure 3-14 shows the logic used to implement Refresh control. At the heart of the Refresh circuitry is a self-starting Refresh control oscillator, a one-shot multivibrator that runs at a repetition rate of every 15 ±1 microseconds. When the system is powered up, the negation side of the oscillator output comes high and is fed back to the input of the oscillator, causing the one-shot to trigger. At the end of 15 microseconds, the negation side comes high, causing a retriggering, thus sustaining the oscillations. The positive Refresh Control (CORREF+) oscillator signal is latched in an Early Refresh Indication flip-flop, which in turn feeds the WAITCM+ logic.

A second one-shot multivibrator, the Fine Refresh oscillator, is triggered by the delayed Refresh Control (CORREF-) signal and generates a Fine Refresh (FINREF+) signal, a pulse of approximately 150 microseconds in duration. The trailing edge of the Fine Refresh function sets the Refresh Request flip-flop. REREQB+ is delayed approximately 150 microseconds from REREQA- to account for any asynchronous bus cycle requests, which can occur simultaneously with Early Refresh Indications. Once the Refresh Request flip-flop is set, the memory proceeds with a Refresh Command if the Memory Busy (MEMBUZ) function is not true, indicating that memory is not executing a cycle. If memory is busy, the Refresh Command flip-flop is not set until the completion of the cycle that corresponds to MEMBUZ- coming high.

A Refresh cycle functions like a normal Memory cycle except for the following:

1. Refresh addresses (REFAD1+ through REFAD7+) are forced in the address logic.
2. The read/write control logic is forced to Read mode.
3. MCASSN+00 is inhibited.
4. All MRASSN+00 clock timing pulses are turned on.

Once a Refresh cycle has started, the normal memory timing process starts and Memory Busy comes true, indicating that memory is busy. The Early Refresh Indication is sent to the Wait generation logic so that any bus activity directed to memory receives a Wait response.

A Refresh cycle is terminated from a Reset (REFRES-) pulse that is generated in the timing generator. The Refresh Reset signal clears the Refresh flip-flops, preparing them for subsequent Refresh cycles.

Figure 3-15 shows the timing relationships for a Refresh cycle. Figure 3-16 shows potential Refresh conflicts. The Bus Data Cycle Now signal being true indicates that a cycle request is present, and it requires between 64 and 90 nanoseconds to decide whether conditions are met for the generation of an Acknowledge (MYACKR+) response. If memory is to generate an Acknowledge response, it is mandatory not to perform a Refresh cycle so that an accepted bus cycle, once started, is not interfered with by the Refresh operation. If the Early Refresh Indication signal goes true in time to catch a positive edge of BSDCNN+, then when it comes time to generate a response, a Wait response is generated instead of an Acknowledge response.

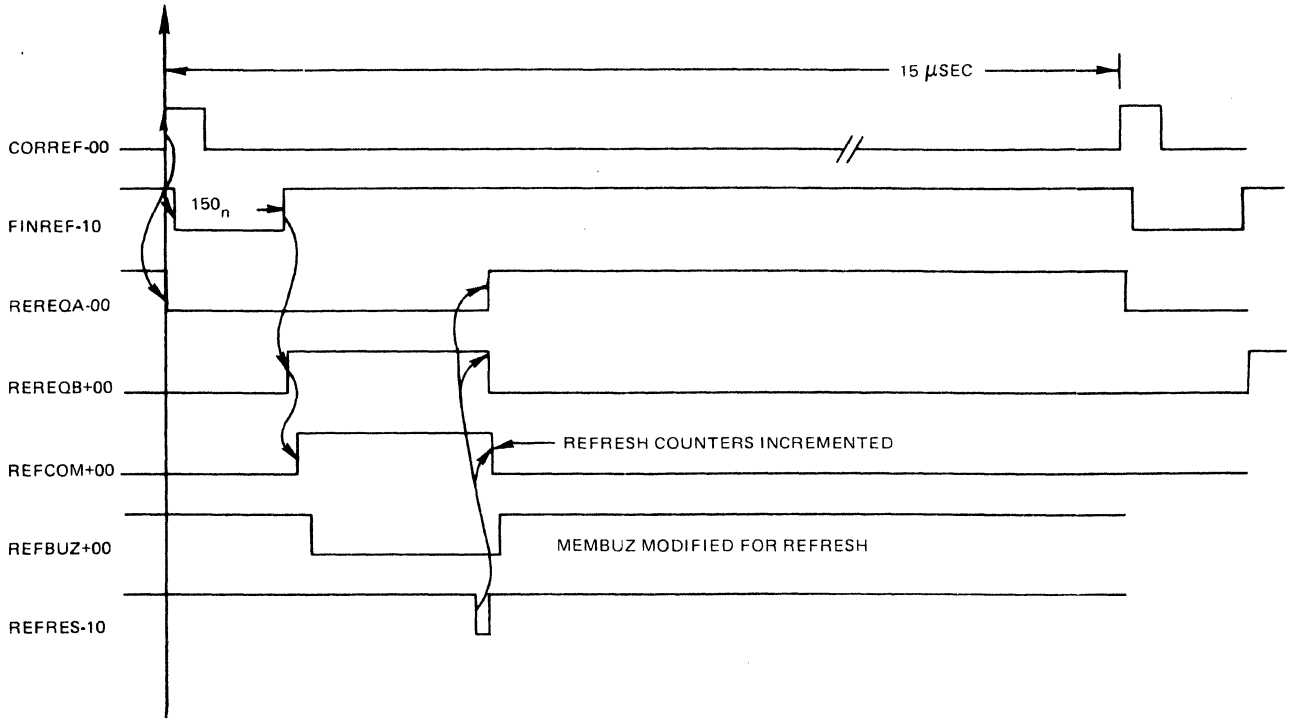


Figure 3-15 Refresh (Normal) Timing

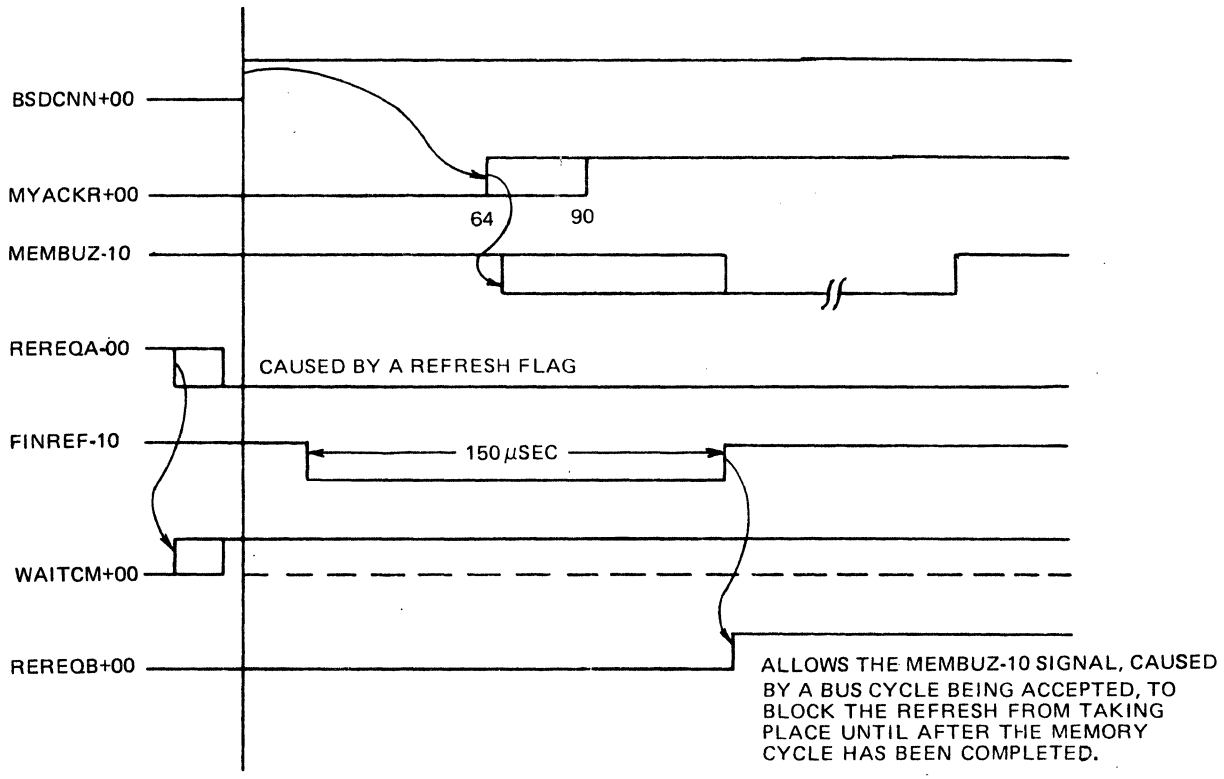


Figure 3-16 Refresh Conflicts

3.5.2 Hidden Refresh

The concept of Hidden Refresh comes about as a result of the memory controller's capability of performing Double-Fetch Read cycles. During a regular Double-Fetch Memory cycle, each stack is fed an independent set of clock signals, address signals, read/write signals, and then the stacks are read out into separate data registers. Because of this duplicate circuitry, the capability exists whereby a Refresh cycle can be substituted for the late timing stack while data is being read out or written into the early timing stack. Additional circuitry is also required to keep track of the Refresh cycles, to synchronize them with regular Single-Word Fetch cycles, and to decide whether or not to do a regular Refresh cycle when it becomes time.

Hidden Refresh cycles begin on the first single-word request on an odd memory location after FINREF-10 comes high (see Figures 3-17 and 3-18). Once the first MYACKR+ comes high, the reset argument on the LK2SWP+ flip-flop comes high, allowing the flip-flop to be clocked. The LK2SWP+ function will be set by the first Single-Word Pull cycle with MLADX2+ high. Once the function is set, further clocking is prevented by forcing the set argument low with LK2SWP-. The positive edge of LK2SWP+ sets the first Single-Word Pull History (SWPHIS+01) register. SWPHIS+01 going true signifies a Refresh operation on the even stack and once started, it causes the Late timing signal MRASSL+00 to execute the stack. SWPHIS+01 is reset by a timing pulse derived off the end of the delay line, which signifies the end of a cycle. Other functions of SWPHIS+01 are the switching of the Refresh addresses into the address path, which forces the read/write line to a Read, inhibits the Data strobe, and overrides the decoded signal for the MRASSE+ clock signal. After SWPHIS+01 is finished, the second Single-Word Pull History (SWPHIS+02) flip-flop looks for a Single-Word Pull cycle on the even stack. When MLADX2-10 and LK2SWP+ are true, SWPCLK+ causes the SWPCLK+02 flip-flop to set. The setting of SWPHIS+02 signifies a Refresh operation on the odd stack, and the same functions occur as with the SWPHIS+01 signal. That is, the odd stack is fed with the Late clock pulse, the Refresh addresses are switched, the read/write line is forced to a Read, the Data strobe is inhibited, and the decoded signals are overridden. SWPHIS+02 is also reset at the end of the cycle by a pulse that is derived off the delay line (RRESET-).

However, once SWPHIS+02 is set, other events must take place which finish the bookkeeping functions of the Hidden Refresh. First, the Inhibit Refresh flip-flop is set on the trailing edge of MYACKD+ (INHREF+ comes high), signifying that both stacks are refreshed and a normal Refresh cycle is not needed. INHREF- is fed to the D-inputs of the Early Refresh Request flip-flop and the Refresh REQUEST-B flip-flop, preventing them from setting and thus eliminating a REFCOM+ from occurring (see Figure 3-14). Second, INHREF- is fed back and inhibits MYACKD+ from clocking the LK2SWP+ and the SWPHIS+02 flip-flop, thus signifying a Stop Looking condition for Single-Word cycles. A third bookkeeping function which occurs at the end of SWPHIS+02 is to increment the Refresh address counters in preparation for the next series of Refresh operations.

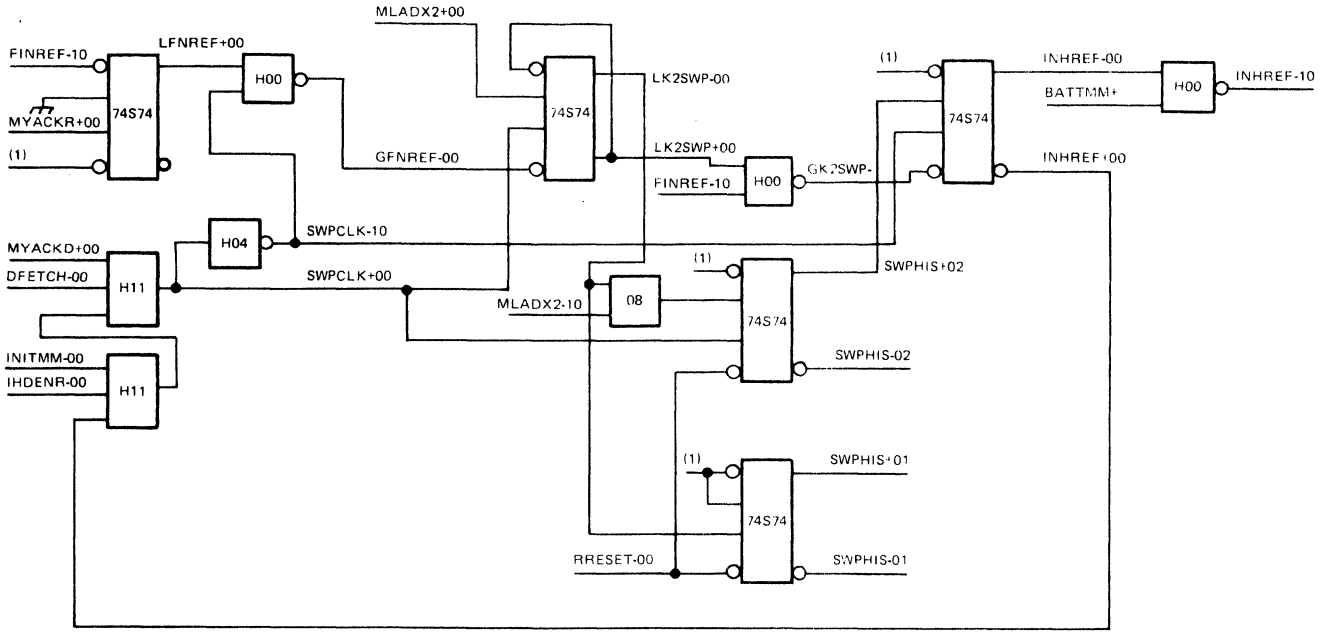


Figure 3-17 Refresh (Hidden) Logic

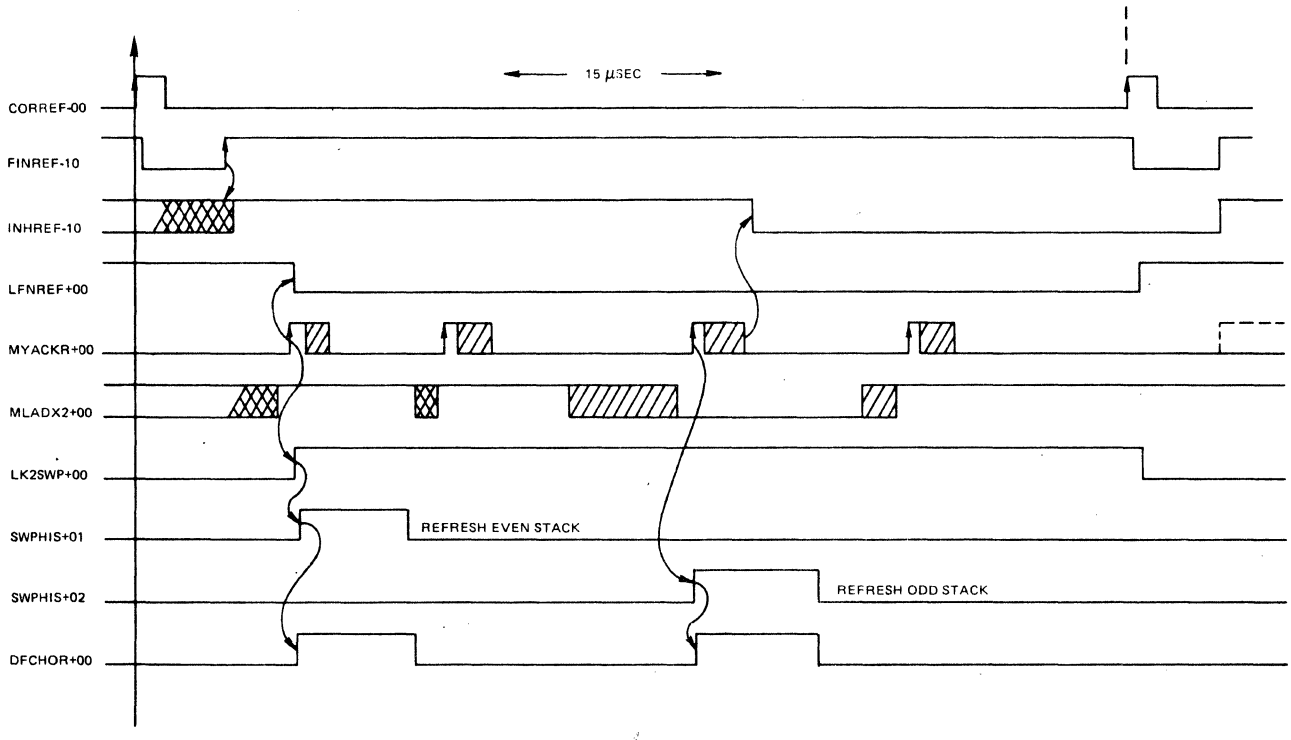


Figure 3-18 Refresh (Hidden) Timing

Two conditions exist which inhibit the functioning of the Hidden Refresh cycles. They are initialization and a manual switch, both of which stop the MYACKD+ signal from performing the necessary clocking operations.

Figure 3-17 shows the logic involved with Hidden Refresh. Figure 3-18 contains an example of the timing sequence within two normal Refresh periods.

3.5.3 Substrate Bias Generator

Each memory controller board supplies a -5 volt ±5 percent bias voltage to its associated Memory-Pacs. This -5 volt (ZVN05B) supply is generated by using a chopper rectifier circuit.

3.6 DATA CONTROL

The data control logic is circuitry that enables data to be written into memory from data multiplexers and/or read from memory and transferred to the Megabus transceivers and the data multiplexers.

Two data formats are available: one for memories that have the EDAC option installed, and one for memories without EDAC. The EDAC version contains 16 data bits and six check bits that are used to detect and correct single-bit errors in the data word and to detect and flag (without correction) double-bit errors in the data word. The non-EDAC version contains 16 data bits and two parity bits. The parity bits, one for each byte of data, are sent to and from memory the same as data bits. They are not checked or flagged in the non-EDAC memory controller.

Data control is explained in three parts:

- Read data flow
- Write data flow
- Data error reporting.

3.6.1 Read Data Flow

During a Read cycle, 16 data bits and two parity bits for non-EDAC memories or 16 data bits and six check bits for EDAC memories are transferred from either odd or even addressed MOS memory stack locations, and latched in independent odd and even local data registers. The information transferred can be either from the odd or even stack as shown in Figure 3-19. Each stack can be read independently of the other. A Double-Word Fetch cycle can be initiated, where one stack is read out early and the other can be read out staggered in time by 250 nanoseconds.

Function MUXDAT- (see Figure 3-19) controls the tri-state operation of the local data registers and is controlled by the state of MLADX2+20, MLADX2-20, and MUXCON+. During normal Single-Word Fetch cycles, MUXCON+ is true. MUXDAT- is controlled by the state of the latched address bit X2. If MLADX2+20 is true, the odd stack data is selected; if MLADX2-20 is true, the even stack data is selected. If a Double-Word Fetch cycle is being executed, MUXDAT- switches (via MUXCON+) at the end of the first MYDCNN-, allowing data in the other register to be selected for the second word. Clocking for the data

registers is generated from the odd/even timing signals MDOOCT+00 and MDOECT+00. These clocking signals are inhibited during Refresh cycles, the second half of Double-Word Fetch cycles, and the Byte Write cycles.

For non-EDAC memories, information lines from the odd and even data registers are hard-wire ORed and inverted to form the Data Out (MDOC00+ through MDOC15+, MDOCP0+, and MDOCP8+) signals. These signals are applied to the inputs of the data bus drivers, and function MYDCNN- comes true to gate the data on the bus. The information that came into the memory during the First Half Read Cycle is turned around and delivered to the address bus to define the address of the unit performing the read from memory.

For memories with the EDAC option, the latched data bits and the check bits from the local register are transferred to the EDAC decoder network, where six syndrome bits (MSYND0 through MSYND5) are generated as indicated in Table 3-2. The decoder network consists of six 9-bit parity generators. For a more detailed explanation of how the decoders work, refer to the explanation of the EDAC encoders in subsection 3.6.2.

Once the syndrome bits are generated, they are applied to five error locator cirucits. The error locator circuits decode one of eight lines, depending on the conditions at the three binary select and three enable inputs. Altogether there are five distinct codes that enable the inputs of the decoders, each code enabling one of the five groups of data bits. Syndrome bits 3 through 5 are then interrogated on the select inputs to further decode to one data bit. At the output of the circuits, a normally high level (indicating no need for correction) is applied to the correction circuits. The data correctors are exclusive-OR circuits that compare the EDAC error locator outputs (MEDMxx) with the incoming data bits from the memory local register (MDOMxx). If one error exists, the particular output that reflects the defective bit goes low, flipping the state of the latched data or check bit. This corrects the state of the data or parity information.

As in the non-EDAC version, once the correct information is obtained at the inputs of the bus drivers, function MYDCNN- comes true to gate the information out of the data bus.

The information that came into the memory controller board on the data lines during the First Half Read Cycle is turned around and sent out on the address bus. This information is used to define the address of the unit that is performing the read from memory.

3.6.2 Write Data Flow

Figure 3-20 shows the write data flow through memory. There are two types of write operations performed by memory: Word Write and Byte Write (see subsections 3.6.2.1 and 3.6.2.2).

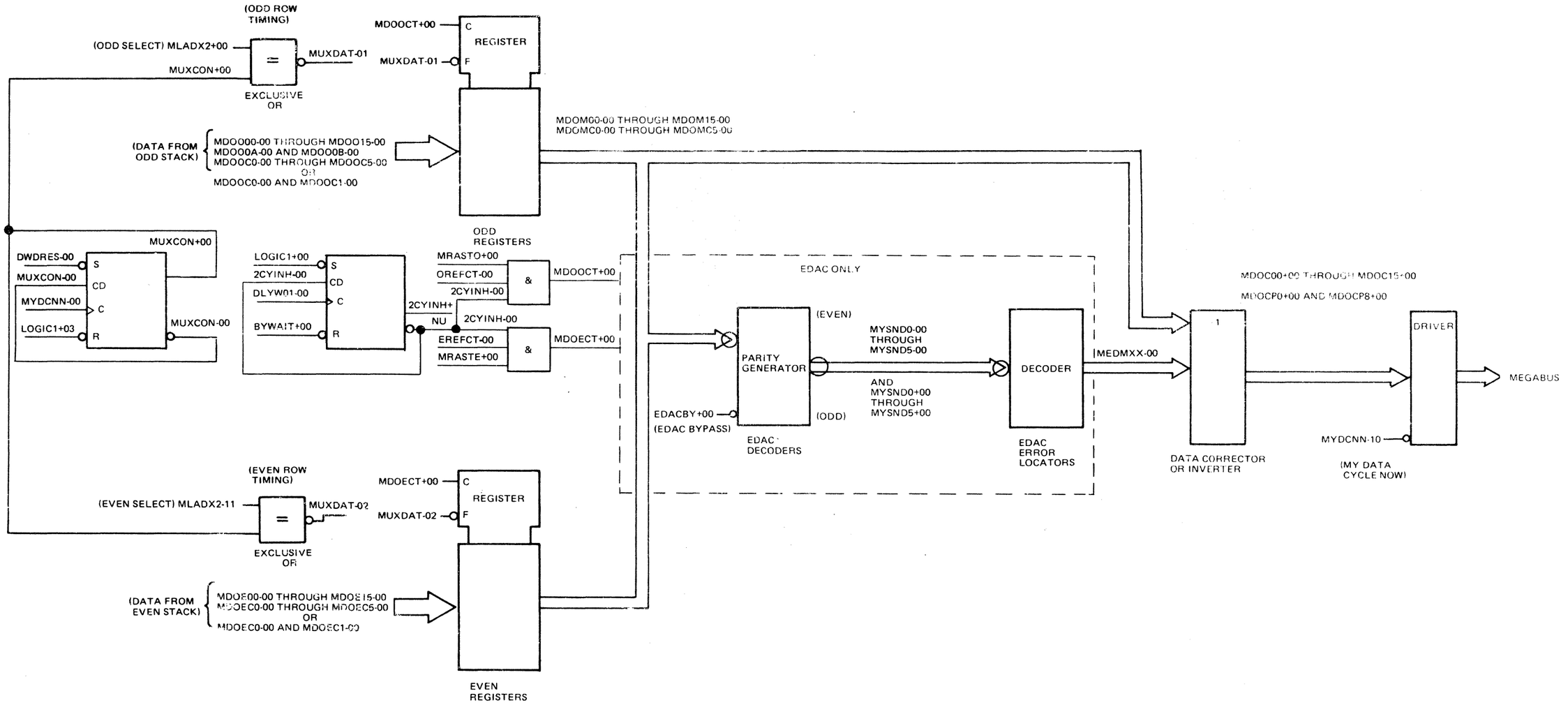


Figure 3-19 Read Data Logic Flow

Table 3-2 Encoder/Decoder Matrix

Encoder Matrix

	0	1	2	3	4	5	6	7	A*	8	9	10	11	12	13	14	15	B*	BPS	PDE
MMDIC0												1	1	1	1	1	1	1	1	
MMDIC1	1	1	1	1	1	1	1												1	
MMDIC2				1	1	1	1								1	1	1	1		
MMDIC3	1	1		1			1	1	1			1	1		1			1		
MMDIC4	1		1		1		1		1	1		1		1		1		1		1
MMDIC5		1	1			1	1		1		1		1	1			1	1		1

BPS = Bus Parity Sum
= MMDIP0 ⊕ MMDIP8
PDE = Partial Write Double Error

Decoder Matrix

	Data Bits																	* *		Check Bits					
	0	1	2	3	4	5	6	7	A*	8	9	10	11	12	13	14	15	B*	BPS	PDE	C0	C1	C2	C3	C4	C5
MSYND0	1	1	1	1	1	1	1	1	1	1	1								1		1					
MSYND1								1	1	1	1	1	1	1	1	1	1	1	1			1				
MSYND2				1	1	1	1								1	1	1	1					1			
MSYND3	1	1		1			1	1	1			1	1		1			1						1		
MSYND4	1		1		1		1		1	1		1		1		1		1		1					1	
MSYND5		1	1			1	1		1		1		1	1			1	1		1						1

*Although bits A and B are not data bits, they are introduced at the data transceivers to assure proper EDAC functionality.

**Not physically applied to decoder inputs; however, if the specified condition exists, the code will be generated.

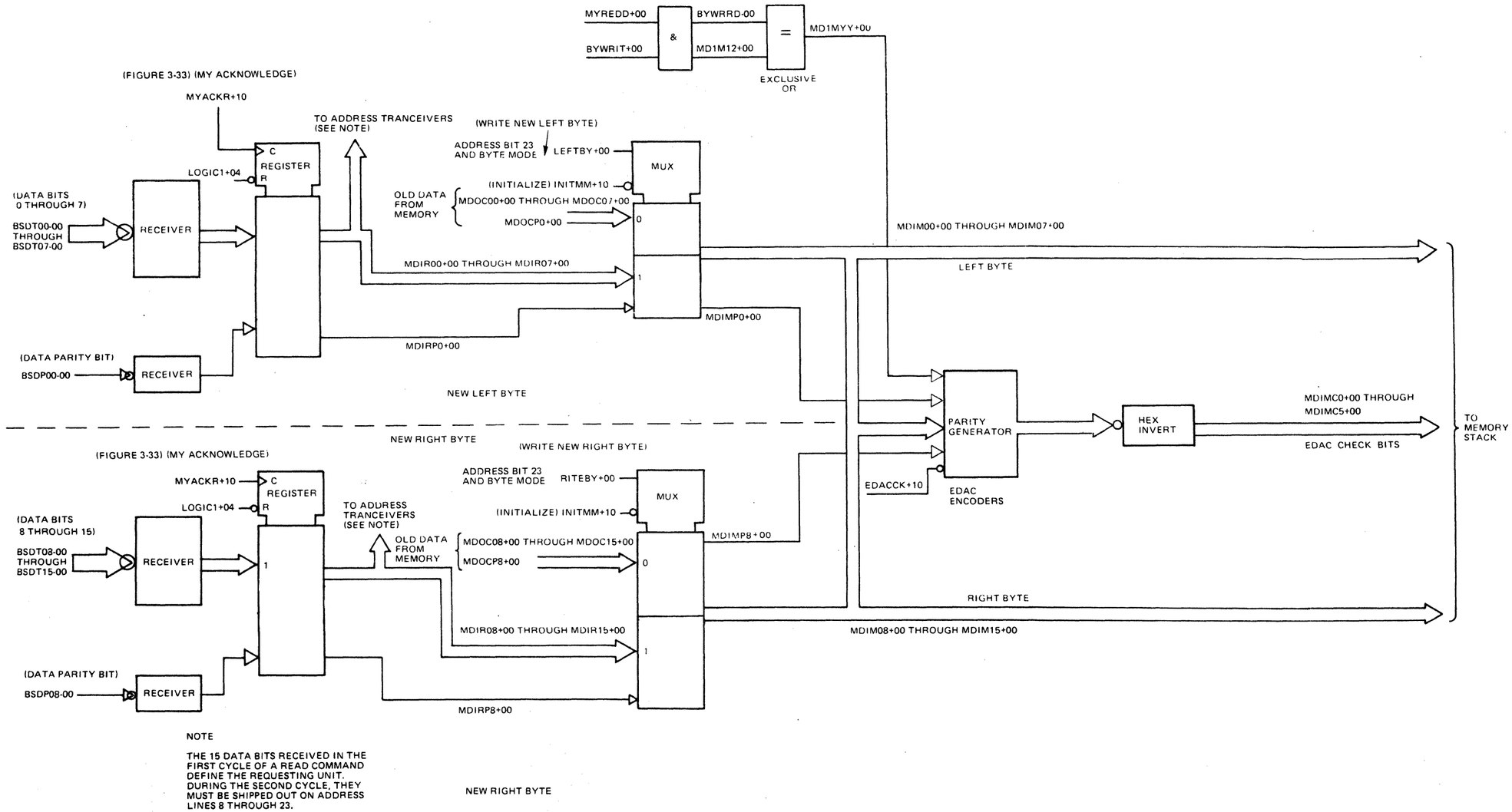


Figure 3-20 Write Data Logic Flow

3.6.2.1 Word Write

During a Word Write operation, 16 data bits and two parity bits are gated from the data bus receivers to the data registers. On the leading edge of MYACKR+, data and parity bits are sent to the data multiplexers while the data bits are also returned as the address code to the address transceivers.

In non-EDAC memories, the Word Write command delivers the data and parity bits to the MOS memory inputs.

In memories that have the EDAC option installed, the Word Write command delivers the data and parity bits to the EDAC encoders and transfers the data (not parity) bits to the inputs of the MOS memory. The data bits are also returned as the address code to the address transceivers.

The EDAC encoder generates six check bits, which are combined with the original 16 data bits at the MOS memory inputs. When the write lines become valid, the 22 bits are written into the specified memory locations.

The EDAC encoder is comprised of six 9-bit parity generators, which produce six check bits (MDIMC0 through MDIMC5) to be written into memory with the data. Each EDAC encoder circuit that supplies a check bit to memory is wired to produce even or odd parity on the groups of bits indicated in Table 3-2. Check bits are inverted prior to being written into memory.

Four special signals are derived to ensure proper functioning of the EDAC encoder. Pseudo data bits BSDT0A+ and BSDT0B+ are introduced false at the data receivers to be processed through the data registers and the EDAC encoder. These bits are not sent to memory, but are used only for the EDAC encoder. Special signal Bus Parity Sum (BPS) is parity bit 0 exclusive-ORed with parity bit 8. Partial Write Double Error (PDE) is a special code forced in the check bit field whenever a double-bit error occurs during a read portion of a Byte Write cycle. The latter two special signals (BPS and PDE) are not written into memory, but they do affect the logical state of the check bits.

If a bus parity error exists, check bits 0 and 1 modify their codes written into memory due to data parity bits 0 and 8 being gated on their inputs. When a request is made to read this specific location for memory, a special syndrome error (110000) is generated, and a Memory Red (MYREDD+) error is delivered to the bus. In this case, the MYREDD+ signal indicates a bus parity error and not a memory error.

If a double-bit error (MMDIYY+) is detected in memory during the read portion of a Byte Write cycle, the appropriate check bits will be affected during the write portion of the cycle. When a read request is later made to this location, a syndrome code of 000011 is generated, and memory delivers a Memory Red (MYREDD+) error to the bus.

3.6.2.2 Byte Write

A Byte Write operation consists of a read of the memory location into which the new byte is to go and a write of the new byte into memory. The Read cycle of the Byte Write operation is executed as described in subsection 3.6.1. The information that is taken from memory is sent to the data multiplexers instead of the bus transceivers with the proper data bytes going to the right and left multiplexer.

The byte that is to receive the new data is determined by the state of Address Bit 23 (BSAD23). If bit 23 is false, the left byte is selected for new data; if bit 23 is true, the right byte is selected.

While the read portion of the Byte Write cycle is in progress, memory is already processing the new data byte through the EDAC encoders if EDAC is installed, or placing the new data at the MOS memory inputs if EDAC is not installed. At the other data multiplexer, the recirculated data is passed through and delivered to the EDAC encoders and/or the MOS memory inputs. For EDAC memories, when encoding of the new data byte is completed, the check bits calculated on one new byte and one old byte are delivered to the MOS memory inputs where they are united with the new and recirculated data bytes. In both the EDAC and non-EDAC systems, when the read/write lines switch to a Write mode, the data word is written into memory.

3.6.3 EDAC Encoder/Decoder

The EDAC encoder is comprised of six 9-bit parity generators, which produce six check bits (MDIMC0 through MDIMC5) to be written into memory with the data. For a more detailed explanation of how the encoders work, refer to the explanation in subsection 3.6.2.1 (see Table 3-2, EDAC Encoder Matrix).

The EDAC decoder network consists of six 9-bit parity generators. For a more detailed explanation on how the decoders work, refer to the description of the EDAC encoders in subsection 3.6.2.1 (see Table 3-2, EDAC Decoder Matrix).

3.6.4 Data Error Reporting

NOTE

Data error reporting is applicable only to controllers with the EDAC option installed. There is no data error reporting in non-EDAC controllers, and the Red and Yellow signals are held false. In this case, data errors are detected via bus parity errors.

Memory error reporting to the CPU or other components on the Megabus is via two lines: Bus Red (BSREDD-) and Bus Yellow (BSYELO-).

When true, BSREDD- indicates to a requesting unit that memory detected a double-bit (noncorrectable) error or a Megabus error from a previous Write cycle and that the information contained on the Megabus is invalid.

When true, BSYELO- indicates that a single error was detected and corrected by memory and that data on the Megabus is valid.

The circuitry involved in generating the memory error signals is shown in Figure 3-21. If the data read from memory contains no error, the syndrome bits (MSYND0- through MSYND5-) from the EDAC decoder are all false. If an error is detected, the Read Error (MREADE+) gate is enabled, and an odd input from the syndrome bits sets the Memory Yellow (MYYELO+) error signal. Even though the MYYELO+ signal is set, the single-bit error is corrected.

If a double-bit error is present, the Read Error (MREADE+) gate sets, yet the parity generator for the single-bit error (MYYELO+) fails to detect the error due to an even input from the syndrome bits. Therefore, the double-bit error gate (MYREDD+) sets and delivers the signal to the Megabus transceivers.

The EDAC Switch (EDACSW+) signal inhibits the MYREDD+ signal whenever diagnostic checks are being made on the EDAC circuitry. When operating in the Diagnostic mode, the generation of check bits from the six parity checkers is also inhibited so that the check field is all Zeros. This allows specific coded data patterns to be written into memory to simulate the picking and dropping of bits. When read out of memory, the decoded portion of the EDAC logic should point to and correct the proper bit position. This scheme verifies proper EDAC operation.

The EDAC error display is available on 128K memory controller boards with the EDAC option installed. There is no EDAC error display in the parity memory controller boards.

The memory EDAC error display's function is to locate failing MOS chips on the Memory-Pacs. The EDAC error display consists of nine Light Emitting Diodes (LEDs) that are located along the right-hand front edge of the BF2MYE memory controller board (see Figure 3-22). In groups of three from left to right, the LEDs are designated D0, D1, and D2. Within the three groups of LEDs, the leftmost LED is the most significant bit and the rightmost LED is the least significant bit. Of the three groups of LEDs, D0 is the most significant bit and D2 is the least significant bit.

Figure 3-23 shows the EDAC error display logic flow. Nine LEDs are driven by two register DIPs. The registers are cleared by Bus Cycle Master Clear (BSMCLR-) and are loaded whenever a MYREDD+00 or MYYELO-00 signal occurs along with MYDCNN+00. Six syndrome bits and three address bits are loaded into the registers when an error is detected. The syndrome bits (MSYND0+00 through MSYND5+00) originate from the EDAC decoder circuits. Address bit MLADX2+00 is the stack select bit. Address bits FADRX1+00 and FADRX3+00 are first multiplexed prior to being applied to the register. This is necessary since the stacks might have different values of FADRX1+00 and FADRX3+00. The LED display always contains the most recent failure code. To determine the identification of the failing Memory-Pac, use Table 3-3.

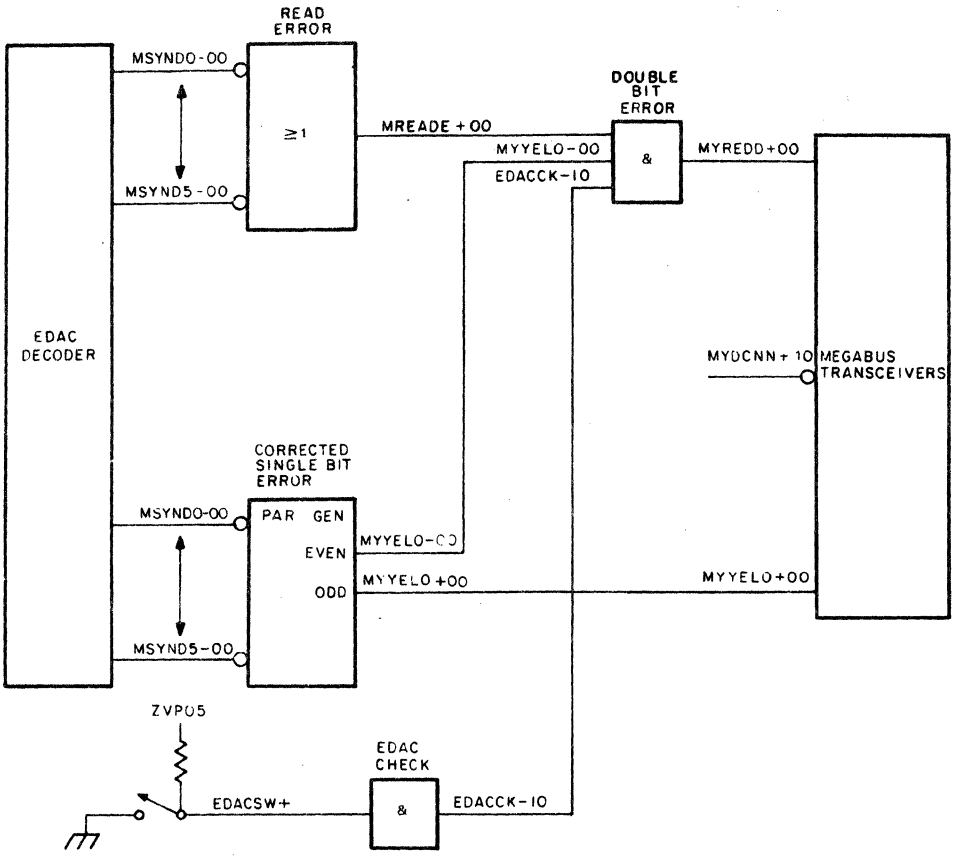


Figure 3-21 EDAC Error Reporting

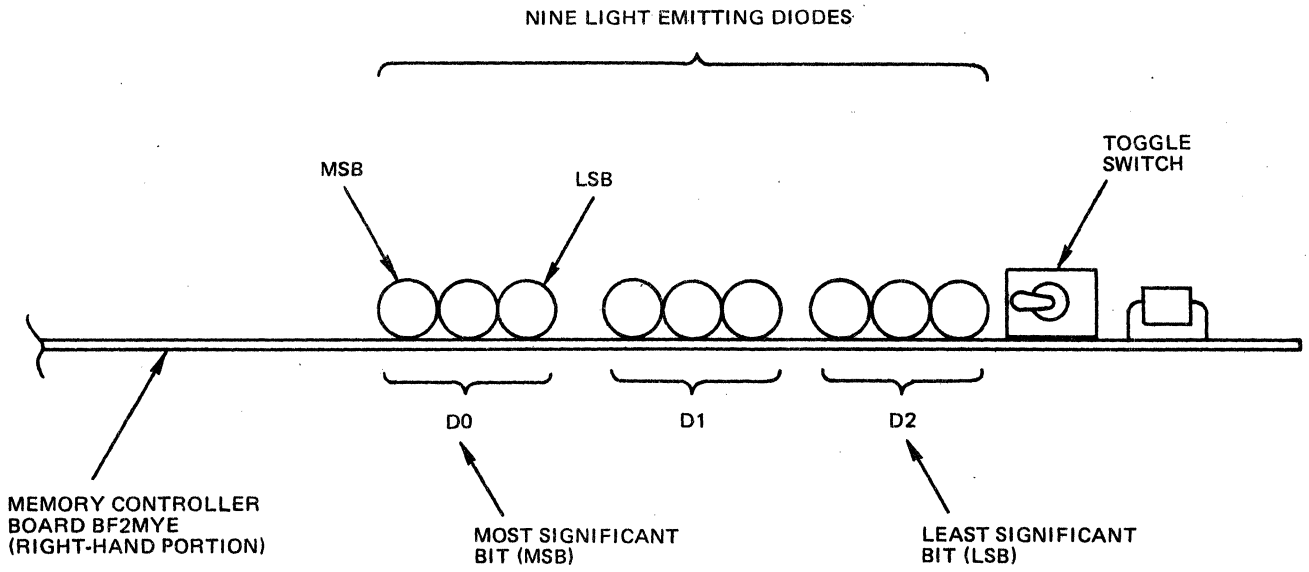


Figure 3-22 Physical Location of Nine Light Emitting Diodes

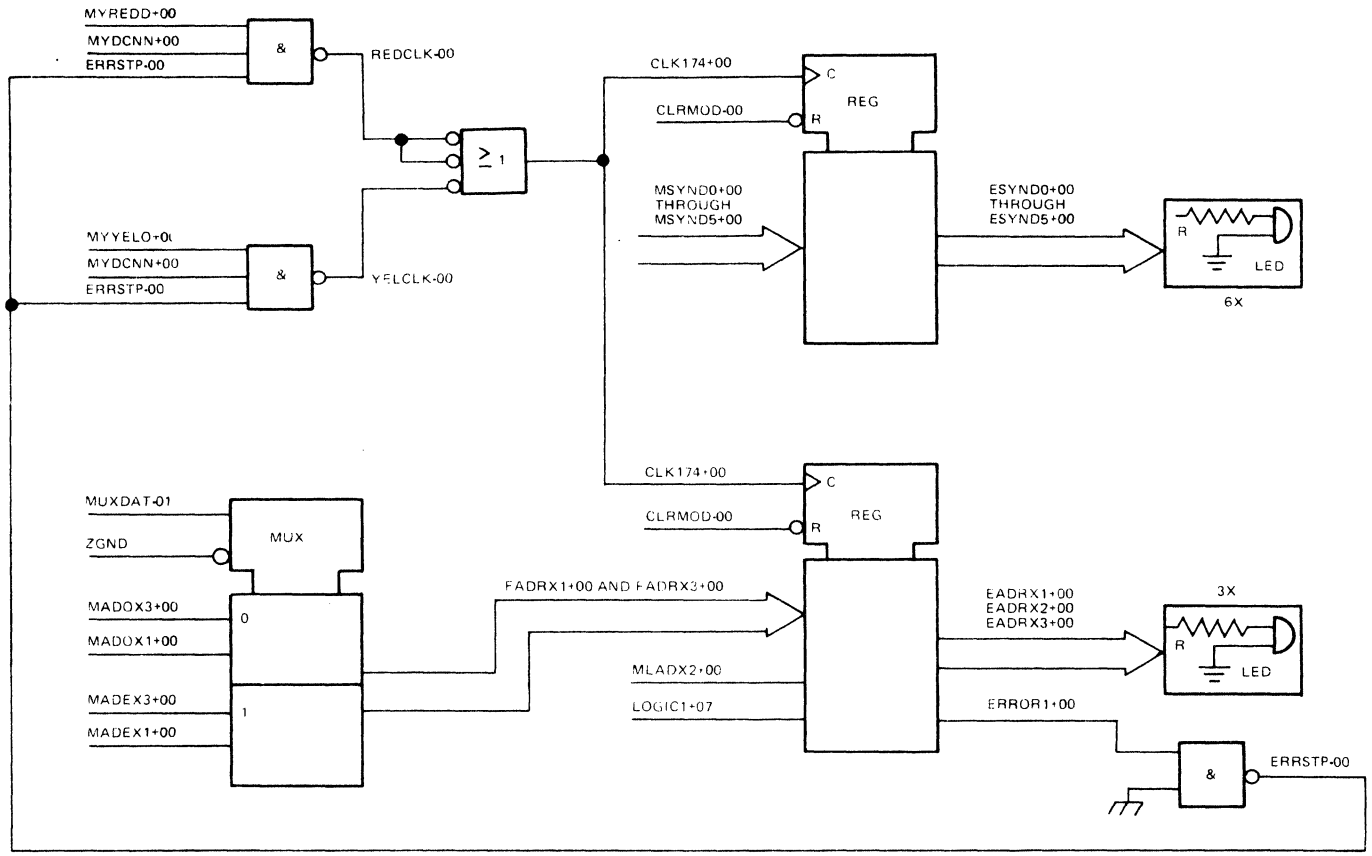


Figure 3-23 EDAC Error Display Logic Flow

Table 3-3 Failing Memory-Pac Identification

DO STATE	FAILING MEMORY-PAC
0	1
1	
2	3
3	
4	2
5	
6	4
7	

3.7 MOS RAM MEMORY

Interfacing with the 128K double-word fetch memory controller are four 32K by 22-bit or 32K by 18-bit Memory-Pacs designated BS2SH4 or BS2SH6, respectively. The Memory-Pacs provide 32K words of memory locations and are attached to the controller via 80-pin connectors. Interfacing signals are TTL logic levels in and out. Once the Memory-Pacs are installed, mechanical integrity is provided via two pan-head screws, which affix the rear of the Memory-Pacs to the controller via threaded stand-offs.

Seven address bits that are multiplexed on the controller are buffered on the Memory-Pacs and routed to address lines A0 through A6 of the MOS array (see Figure 3-24). Each address bit is buffered twice and is fed to 24 MOS RAMs. Address loading is approximately 230 picofarads with total delay being 15 through 25 nanoseconds.

DRASTE+ and DRASTO+ are buffered and routed to 12 MOS RAMs within a byte. Each clock line is series terminated through a 39-ohm damping resistor and is pulled up to a zener diode derived 5 volts through a 5.6-Kohm resistor. Two of the four circuits use MRASTE+ and the other two are used for MCASTE+ tracking purposes. Distribution of MCASTE+ is similar to that of the MRASTE+ (see Figure 3-25) except that MCASTO+ and MCASTE+ are not decoded.

The read/write line is distributed up byte oriented; i.e., the left byte is buffered differently from the right byte (see Figure 3-26). Data in and Data out lines are wired in parallel. Each data bit pair is as shown in Figure 3-27.

3.8 MEGABUS CONTROL

The Megabus provides a common communication path among available system units. The design of the Megabus is asynchronous to make communications possible between units of varying speeds.

3.8.1 Interface Logic

The Megabus interfaces with the memory via a group of transceivers that provide the equivalent electrical characteristics required of all bus connections to allow data, address, and control signals to be routed to and from the memory. Table 3-4 provides a complete list of the interface signals, while subsections 3.8.1.1 through 3.8.1.6 provide a brief description of each Megabus/Memory interface signal.

3.8.1.1 Timing Signal Lines

The following signals provide the Megabus handshake functions that are required by the available units to either initiate, accept, or deny a request for a Megabus cycle from a particular unit (i.e., slave or master).

Bus Request (BSREQT-)

When true, this signal indicates to memory that one or more of the units connected to the Megabus requested a bus cycle. Memory drives BSREQT- true only when it is requesting a Read Second Half Bus Cycle.

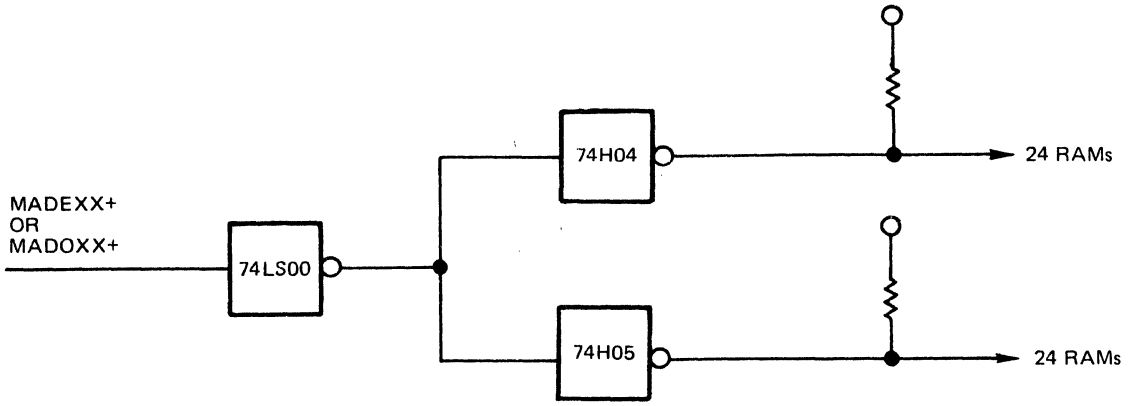


Figure 3-24 Address Lines

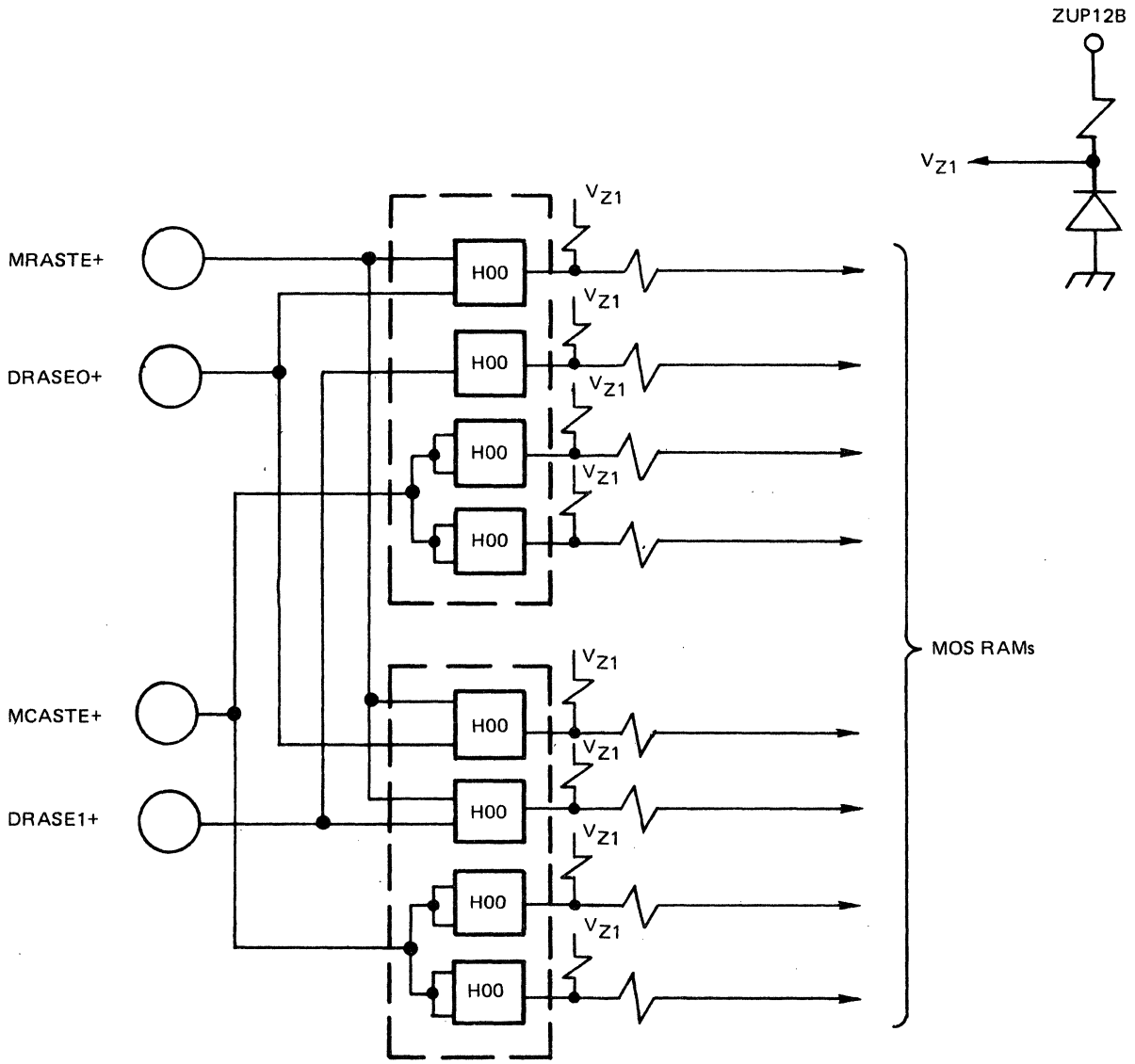


Figure 3-25 Memory-Pac Clock Distribution

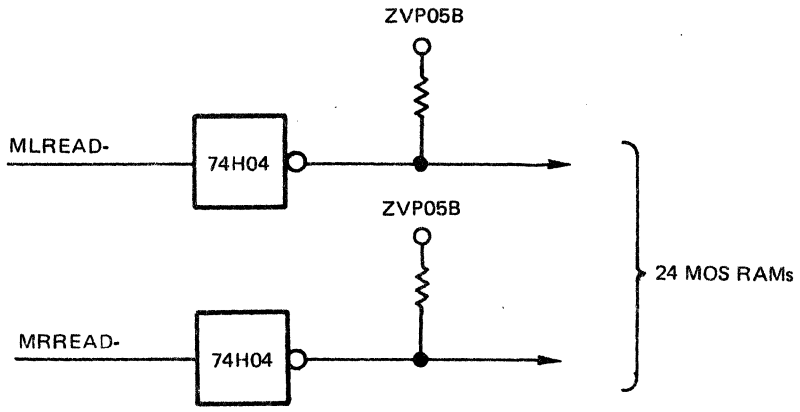


Figure 3-26 Read/Write Distribution

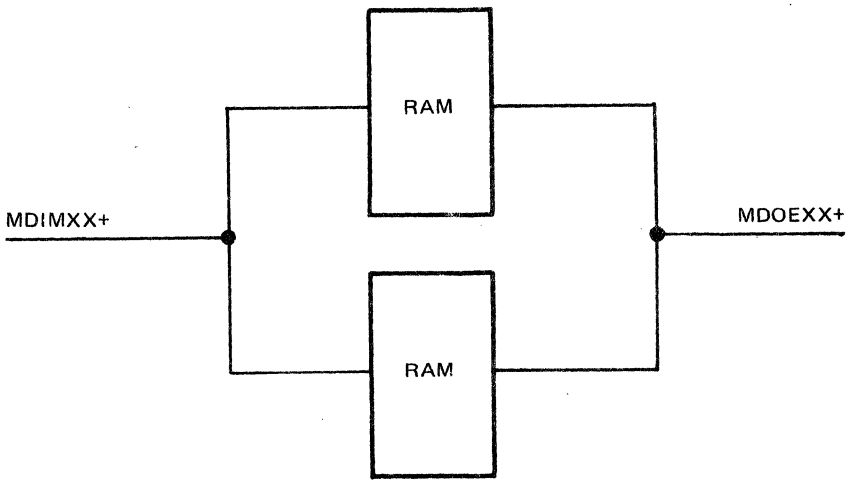


Figure 3-27 Data In/Data Out Logic

Table 3-4 Megabus Interface Signals (Sheet 1 of 3)

TYPE	FUNCTION	MEMORY SIGNAL			
		DRIVER ENABLE	DRIVER INPUT	RECEIVER OUTPUT	BUS SIGNAL
Bus Handshake	Bus Request	GROUND	MYREQT+	BSREQT+	BSREQT-
	Data Cycle Now	GROUND	MYDCNN+	BSDCNN+	BSDCNN-
	Acknowledge	GROUND	MYACKR+	BSACKR+	BSACKR-
	Negative Acknowledge	GROUND	MYNAKR+	BSNAKR+	BSNAKR-
	Wait	GROUND	MYWRIT+	BSWAIT+	BSWAIT-
Data	Data Bit A (unused)	-	-	-	BSDT0A-
	0	MYDCNN-	MDOC00+	BSDT00+	BSDT00-
	1	MYDCNN-	MDOC01+	BSDT01+	BSDT01-
	2	MYDCNN-	MDOC02+	BSDT02+	BSDT02-
	3	MYDCNN-	MDOC03+	BSDT03+	BSDT03-
	4	MYDCNN-	MDOC04+	BSDT04+	BSDT04-
	5	MYDCNN-	MDOC05+	BSDT05+	BSDT05-
	6	MYDCNN-	MDOC06+	BSDT06+	BSDT06-
	7	MYDCNN-	MDOC07+	BSDT07+	BSDT07-
	B (unused)	-	-	-	BSDT0B-
	8	MYDCNN-	MDOC08+	BSDT08+	BSDT08-
	9	MYDCNN-	MDOC09+	BSDT09+	BSDT09-
	10	MYDCNN-	MDOC10+	BSDT10+	BSDT10-
	11	MYDCNN-	MDOC11+	BSDT11+	BSDT11-
	12	MYDCNN-	MDOC12+	BSDT12+	BSDT12-
	13	MYDCNN-	MDOC13+	BSDT13+	BSDT13-
	14	MYDCNN-	MDOC14+	BSDT14+	BSDT14-
	Data Bit 15	MYDCNN-	MDOC15+	BSDT15+	BSDT15-
Main Memory Address	Address Bit 0	GROUND	GROUND	BSAD00+	BSAD00-
	1	GROUND	GROUND	BSAD01+	BSAD01-
	2	GROUND	GROUND	BSAD02+	BSAD02-
	3	GROUND	GROUND	BSAD03+	BSAD03-
	4	GROUND	GROUND	BSAD04+	BSAD04-
	5	GROUND	GROUND	BSAD05+	BSAD05-
	6	GROUND	GROUND	BSAD06+	BSAD06-
	7	GROUND	GROUND	BSAD07+	BSAD07-
	8	MYDCNN-	MDIR00+	BSAD08+	BSAD08-
	Address Bit 9	MYDCNN-	MDIR01+	BSAD09+	BSAD09-

HONEYWELL PROPRIETARY AND CONFIDENTIAL

Table 3-4 Megabus Interface Signals (Sheet 2 of 3)

TYPE	FUNCTION	MEMORY SIGNAL			
		DRIVER ENABLE	DRIVER INPUT	RECEIVER OUTPUT	BUS SIGNAL
Main Memory Address	Address Bit 10	MYDCNN-	MDIR02+	BSAD10+	BSAD10-
	11	MYDCNN-	MDIR03+	BSAD11+	BSAD11-
	12	MYDCNN-	MDIR04+	BSAD12+	BSAD12-
	13	MYDCNN-	MDIR05+	BSAD13+	BSAD13-
	14	MYDCNN-	MDIR06+	BSAD14+	BSAD14-
	15	MYDCNN-	MDIR07+	BSAD15+	BSAD15-
	16	MYDCNN-	MDIR08+	BSAD16+	BSAD16-
	17	MYDCNN-	MDIR09+	BSAD17+	BSAD17-
	18	MYDCNN-	MDIR10+	BSAD18+	BSAD18-
	19	MYDCNN-	MDIR11+	BSAD19+	BSAD19-
	20	MYDCNN-	MDIR12+	BSAD20+	BSAD20-
	21	MYDCNN-	MDIR13+	BSAD21+	BSAD21-
	22	MYDCNN-	MDIR14+	BSAD22+	BSAD22-
	Address Bit 23	MYDCNN-	MDIR15+	BSAD23+	BSAD23-
Transfer	Memory Reference	MYDCNN-	GROUND	BSMREF+	BSMREF-
	Byte (unused)	MYDCNN-	GROUND	BSBYTE+	BSBYTE-
	Bus Write	GROUND	GROUND	BSWRIT+	BSWRIT-
	Second Half Read	MYDCNN-	LOGIC1+	BSSHBC+	BSSHBC-
	Lock (unused)	MYDCNN-	GROUND	BSLOCK+	BSLOCK-
	Memory Error (Red)	MYDCNN-	MYREDD+	BSREDD+	BSREDD-
	Memory Error (Yellow)	MYDCNN-	MYYELO+	BSYELO+	BSYELO-
	Data Parity Left	MYDCNN-	MDOCP0+	BSDP00+	BSDP00-
	Data Parity Right	MYDCNN-	MDOCP8+	BSDP08+	BSDP08-
	Address Parity (bits 0-7)	MYDCNN-	LOGIC1+	BSAP00+	BSAP00-
	Double Pull	GROUND	DWRESP+	BSDBPL+	BSDBPL-
Verify Bus Continuity	Logic Test Out	-	-	-	BSQLTO-
	Logic Test In	-	-	-	BSQLTI-
	Logic Test Active	-	-	-	BSQLTA+

HONEYWELL PROPRIETARY AND CONFIDENTIAL

Table 3-4 Megabus Interface Signals (Sheet 3 of 3)

TYPE	FUNCTION	MEMORY SIGNAL			
		DRIVER ENABLE	DRIVER INPUT	RECEIVER OUTPUT	BUS SIGNAL
Establish Positional Priority	Tie-Breaking Network	-	-	-	BSAUOK+
	↕	-	-	-	BSBUOK+
	↕	-	-	-	BSCUOK+
	↕	-	-	-	BSDUOK+
	↕	-	-	-	BSEUOK+
	↕	-	-	-	BSFUOK+
	↕	-	-	-	BSGUOK+
	↕	-	-	-	BSHUOK+
	↕	-	-	-	BSIUOK+
	Tie-Breaking Network	-	-	-	BSMYOK+
Miscella-neous	Master Clear	GROUND	GROUND	BSMCLR+	BSMCLR-
	Power On	-	-	-	BSPWON+
	Resume Interrupt	-	-	-	BSRINT-
	Spare Line (unused)	-	-	-	BSSPR1-
	Spare Line (unused)	-	-	-	BSSPR3-
	Spare Line (unused)	-	-	-	BSSPR4-
	External Connection (unused)	-	-	-	BSEXTC+

HONEYWELL PROPRIETARY AND CONFIDENTIAL

Bus Data Cycle Now (BSDCNN-)

When true, this signal indicates that a unit on the Megabus was granted a requested bus cycle, and placed information on the bus for use by another unit. The memory must now determine if it is the unit being requested, and it must also determine the proper response. The four following responses are possible:

- Bus Acknowledge (BSACKR-)
- Bus Wait (BSWAIT-)
- Bus Negative Acknowledge (BSNAKR-)
- No response.

Memory drives the Bus Data Cycle Now signal when it is transmitting requested data back to a unit on the Megabus. Prior to driving this signal true, memory must have requested and received a bus cycle.

Bus Acknowledge (BSACKR-)

When true, this signal indicates that memory is informing a master unit that it is accepting the present Megabus transfer (Read First Half Bus Cycle or Write Cycle). During a Read Second Half Bus Cycle, the unit which originally requested information (First Half Bus Cycle) always sends a true Bus Acknowledge signal to memory.

Bus Negative Acknowledge (BSNAKR-)

Memory signals to the master unit that it is refusing a specified transfer by setting BSNAKR- true. This indicates that the master unit should not retry, but should take specific action, depending on the type of master unit. Memory generates a true BSNAKR- signal only if the following conditions occur:

1. The Memory Lock flip-flop must be set.
2. The present master unit is requesting a Megabus cycle with a Test and Set Lock condition (BSLOCK- true; BSSHBC- false).

In all other cases with the Lock flip-flop set, memory responds with a Bus Acknowledge (BSACKR-), a Bus Wait (BSWAIT-), or no response.

Memory never expects to receive a true BSNAKR- in response to its Read Second Half Megabus cycle. If a true BSNAKR- is received, memory terminates its present Megabus cycle and the data is not accepted by the requesting master unit. Memory does not request another Read Second Half Megabus cycle, and the requesting master unit never receives the required data.

Bus Wait (BSWAIT-)

When this signal is true, memory is informing the master unit that it cannot accept a specified Megabus transfer at this time. The master unit must initiate successive retries until the transfer is acknowledged. Memory drives BSWAIT- true for one of the following reasons:

1. Memory is busy performing an internal Read or Write cycle.

2. Memory is requesting the Megabus for a Read Second Half Megabus cycle. (Memory may have completed an internal Read cycle and may be in a Standby state.)

3. Memory is anticipating a Refresh mode.

4. Memory is busy performing a Refresh cycle.

5. Memory is busy in the Initialize mode.

Memory never expects to receive BSWAIT- true in response to its Read Second Half Megabus cycle. If a true BSWAIT- is received, memory terminates its present Megabus cycle with the data not being accepted by the requesting master. Memory does not request another Read Second Half Megabus cycle, and the requesting master unit never receives the required data.

No Response

When BSDCNN is true and a situation arises where no unit on the system bus recognizes a transfer (i.e., due to improper addressing or a malfunction), no response is generated. Under these conditions, the CPU, which monitors all bus transfers, issues a Bus Negative Acknowledge Response (BSNAKR) signal on behalf of the entire system if a response (ACK, NAK, or WAIT) is not generated within 5 microseconds. This operation is defined as a Dead Man Time-Out.

Memory does not respond to a true BSDCNN signal under any of the following conditions:

1. When Bus Address Bits 00 through 05 (BSAD00 through BSAD05) do not compare with the memory identification.

2. When memory receives the proper identification, but detects bad address parity on bits 00 through 07.

3. When the requested word address is located on a nonexistent Memory-Pac.

3.8.1.2 Information Signal Lines

The information signals are defined as address and data signals, and are transferred as the information content of each Megabus cycle. The information signals are valid for use by the memory controller on the leading edge of the strobe (MYDCNN-), which is generated within the memory controller (delayed 60 nanoseconds from BSDCNN-).

Bus Data Lines (BSDT00- through BSDT15-)

The data field is two bytes (16 bits) wide and contains either data or the channel number of the master unit, the configuration being dependent on the Megabus cycle being performed.

If the operation is a Write cycle, memory receives new information from the master unit and writes this information into the location specified by the address field.

For a Read First Half Bus cycle, memory receives identification information from the master unit (channel number) across the data lines.

During a Read Second Half Bus cycle, memory transmits to the requesting master unit, across the data lines, the contents of the location specified by the address field during the Read First Half Bus cycle.

Bus Address Lines (BSAD00- through BSAD23-)

The Bus Address Lines transfer a 24-bit address to memory or a 16-bit identifier from memory to the slave unit.

When used for memory addressing, signals BSAD00- through BSAD05- select a particular 128K-word module. Signals BSAD06- through BSAD22- select one of the 128K words in a given module. Signal BSAD23- selects one of the bytes in a given word (Zero selects the right byte; One selects the left byte).

When the address field is used by memory to identify a receiving unit, signals BSAD00- through BSAD07- are not used. Signals BSAD08- through BSAD23- carry the identification of the receiving unit as transmitted to memory during the previous memory read request.

3.8.1.3 Information Control Signal Lines

The signals which effect the transfers of information or data during a Megabus cycle are defined as control accompanying transfer signals.

Bus Memory Reference (BSMREF-)

When true, BSMREF- indicates that the Bus Address Lines (BSAD00- through BSAD23-) contain a complete memory controller address. BSMREF- true indicates to memory that a master unit has a Write or Read operation to perform on a specified memory location.

When false, BSMREF- indicates to memory that the address lines contain information which is directed at a controller other than memory.

Byte Mode (BSBYTE-)

When this signal is true, memory is informed that the master unit wishes to perform a Byte (8-bit) operation rather than a Word (two-byte) operation. The operation performed by memory is dependent on whether a Write or Read operation is performed.

If BSBYTE- and BSWRIT- are both true, memory writes new data into either the left or right byte of the memory location (two bytes) addressed; the byte written depends on the state of Address Bit 23 (BSAD23-).

If BSBYTE- is true and BSWRIT- is false, the master unit is requesting a read from a memory location, and memory transmits the entire contents of the location (two bytes) to the Megabus. The requesting unit then selects the requested byte within the confines of its controller.

Bus Write (BSWRIT-)

When true, this signal in conjunction with a true Memory Reference (BSMREF-) signal indicates to memory that a master unit wants to perform a Write cycle. During this period, the address lines contain a memory controller address and the data lines contain information to be stored in the memory location.

When false, this signal in conjunction with a true Memory Reference signal indicates that a master unit wants to perform a Read operation on memory. The address and data lines contain the information as dictated in subsection 3.8.1.2.

Bus Second Half Bus Cycle (BSSHBC-)

When true, this signal indicates to a unit on the bus that the current bus information (sent from memory) is the information previously requested during a read request. Under these conditions, both memory and the unit receiving the information are busy to all other units from the start of the initiation cycle until memory completes the transfer. This signal is also used in conjunction with the Bus Lock (BSLOCK) signal to set or reset the memory Lock signal.

If a unit on the bus is requesting to write data into memory or is requesting information to be sent from memory and BSLOCK is true, memory assumes the following definition of BSSHBC: true equals Reset Lock and false equals Test and Set Lock. If BSLOCK is false, memory ignores the BSSHBC signal.

Lock (BSLOCK-)

When true, this signal indicates to memory that a CPU or Intersystem Link (ISL) is requesting a test or change in status of the memory Lock flip-flop. When a CPU and memory are in a Lock state, any other unit requesting a Lock condition with memory receives a Bus Negative Acknowledge Response (BSNAKR-). Units which try to perform normal Read or Write operations with memory receive BSACKR-, BSWAIT-, or no response, as applicable.

The Lock flip-flop resides only in the memory controller and is usable only by the CPU or the ISL on behalf of a CPU.

3.8.1.4 Status/Error Signal Lines

The following bus signals provide memory controller error reporting signals for the available units, and bidirectional parity lines for the generation of odd parity for the address/data bits placed on the system bus. The memory error lines signify the integrity of the memory subsystem, but memory does not monitor either error signal during any type of system bus transfer.

Memory Error Bus Red (BSREDD-)

This signal can only be used by memory if the EDAC option is installed. When true, this signal indicates that memory detected an uncorrectable error in either the currently read data bits or bad bus parity from the last write into the current location. If memory contains parity logic instead of EDAC, the BSREDD- signal always remains false. Bus parity errors determine memory errors.

Memory Error Bus Yellow (BSYELO-)

This signal can only be used by memory if the EDAC option is installed. When true, this signal indicates that memory detected and corrected a single data bit error. If memory contains parity logic rather than EDAC logic, the BSYELO- signal always remains false. Bus parity errors determine memory errors.

Bus Data Parity – Left Byte (BSDP00-)

The level of the Bus Data Parity – Left Byte signal provides odd parity for the left data byte (bits 0 through 7).

Bus Data Parity – Right Byte (BSDP08-)

The level of the Bus Data Parity – Right Byte signal provides odd parity for the right data byte (bits 8 through 15).

Bus Address Parity (BSAP00-)

The level of the Bus Address Parity signal provides odd parity for address bits 0 through 7 (module address bits).

Double-Word Pull (BSDBPL-)

When true, this signal indicates that a double-word is requested from memory. Memory acts on a true state, and if the required locations are valid, drives the line true during Data Cycle Now. This indicates that the second word is to follow. If the second location is nonexistent, the line is not driven true and only a single word is transferred.

Bus Quality Logic Test In (BSQLTI-)

When true, this signal indicates that all units of lower priority than memory completed their respective QLTs. When BSQLTI- is false, it indicates that a unit of lower priority than memory is missing and/or that one or more units did not complete their QLTs.

Bus Quality Logic Test Out (BSQLTO-)

When true, this signal indicates that memory and all units with lower priority are present and did complete their respective QLTs. When BSQLTO- is false, it indicates that memory or a unit with lower priority is missing and/or that memory or one or more units did not complete their QLTs.

Bus Quality Logic Test Active (BSQLTA+)

This signal is not used by memory.

Power On (BSPWON+)

When true, this signal indicates that all system power is valid. The first transition from false to true causes memory to enter the Initialize mode. During initialization, memory is not available for use by other units. When the Power On signal makes a transition from true to false, it indicates to memory that power is going down. If the Memory Save Power Supply option is installed, memory allows

Read/Write cycles for a maximum of 1.5 milliseconds. Afterwards, memory does not permit bus cycles to alter its stored data. Additional transitions from false to true, with no Memory Save Power Supply option installed, cause memory to become unavailable while in initialization. After initialization, memory is again available for bus cycles. Additional transitions from false to true, with the Memory Save Power Supply option installed, do not cause memory to initialize. Memory remains unavailable until the trailing edge of the Master Clear signal.

3.8.1.5 Tie-Breaking Control Signals

The memory uses all 10 tie-breaking signals to form its tie-breaking network. This network, along with similar networks on other units, resolves simultaneous bus requests and grants bus cycles on the basis of a positional priority system.

If memory makes a bus request, it drives the tie-breaking network signal BSMYOK+ false to notify units of lower priority that it wants the bus. Memory also looks at the tie-breaking network signals BSAUOK+ through BSIUOK+ to determine if units of higher priority have requested the bus. If these signals are all high, then memory receives the bus and can send the Data Cycle Now signal. If any of these signals are low, memory did not receive the bus and therefore must not drive the Data Cycle Now signal on the bus.

3.8.1.6 Operational Control Signal Bus Master Clear (BSMCLR-)

The BSMCLR- control signal is asynchronous in relation to the functions it performs and the normal initiation and control of bus cycles. This signal is generated true when the Master Clear (CLR) pushbutton on the CPU control panel is depressed or a power fail sequence occurs.

When true, this signal causes memory to clear its Megabus handshaking logic. On the trailing edge of Master Clear (i.e., the low to high transition), memory becomes available for Megabus cycles. For the exception to normal procedures, Initialize mode, refer to subsection 3.8.1.4, Power On (BSPWON+).

3.8.2 Megabus Formats

Figure 3-28 depicts the information format for the data and address lines during various Megabus cycle operations. Each format shown reflects the occurrence of a single Megabus cycle. Available units can identify the cycle type by the state of five control signals that are transmitted onto the bus with each format. These control signals are defined in subsection 3.8.1.3 and three of these signals (BSWRIT, BSSHBC, and BSMREF) are shown in Figure 3-28.

KEY CONTROL SIGNALS				OPERATION	NO. OF CYCLES	MASTER	SLAVE	ADDRESS LINES (BSAD)	DATA LINES (BSDT)
BSDBPL–	BSWRIT–	BSSHBC–	BSMREF–						
F	T	F	T	MEMORY WRITE	1	CPU+CU +CIP	MEM	0 – 23: ADDRESS	0 – 15: DATA
F	F	F	T	MEMORY READ AND RESPONSE	1	CPU+CU +CIP	MEM	0 – 23: ADDRESS	0 – 9: MASTER CHANNEL NUMBER; 10 – 15: VARIABLE USAGE
F	F	T	F	SINGLE-WORD PULL	1	MEM	CPU+CU +CIP	0 – 7: (shaded); 8 – 17: SLAVE CHANNEL NUMBER; 18 – 23: VARIABLE USAGE	0 – 15: DATA
T	F	F	T	MEMORY READ AND RESPONSE	1	CPU+CU +CIP	MEM	0 – 23: ADDRESS	0 – 9: MASTER CHANNEL NUMBER; 10 – 15: VARIABLE USAGE
T	F	T	F*	DOUBLE-WORD PULL	1	MEM	CPU+CU +CIP	0 – 7: (shaded); 8 – 17: SLAVE CHANNEL NUMBER; 18 – 23: VARIABLE USAGE	0 – 15: DATA (FIRST WORD)
F	F	T	F*		1	MEM	CPU+CU +CIP	0 – 7: (shaded); 8 – 17: SLAVE CHANNEL NUMBER; 18 – 23: VARIABLE USAGE	0 – 15: DATA (SECOND WORD)
F	F	T	F**		1	MEM	CPU+CU +CIP	0 – 7: (shaded); 8 – 17: SLAVE CHANNEL NUMBER; 18 – 23: VARIABLE USAGE	0 – 15: DATA (FIRST WORD)
F	T	F	T	MEMORY WRITE	1	CPU+CU +CIP	MEM	0 – 23: BYTE ADDRESS	0 – 7: DATA; 8 – 15: DATA

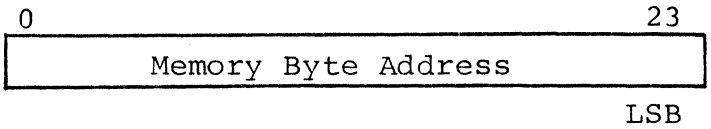
*BOTH WORDS IN MEMORY.
**SECOND WORD NOT IN MEMORY.

Figure 3-28 Megabus Formats

3.8.3 Unit Addressing

3.8.3.1 Slave Unit Addressing

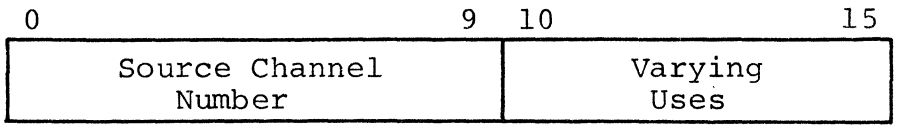
A master unit may address any other unit on the bus as the slave unit. It does this by placing the slave address on the address line. There are 24 address lines, which can have either of two interpretations, depending on the state of the Memory Reference (BSMREF) signal. If Memory Reference is true, the following format applies to the address lines:



If Memory Reference is false, the master unit is communicating with a unit other than memory.

3.8.3.2 Identification of Master Unit

When a master unit requires a response from the slave unit, it indicates this to the slave unit by forcing the Bus Write (BSWRIT-) signal false. In addition, the master unit provides its own identity to the slave unit by means of a channel number. This is coded on the data lines of the bus as follows:



The response cycle is directed to the register (master unit) by a nonmemory reference transfer. The Second Half Bus Cycle (BSSHBC-) accompanies the transfer to indicate that this is the awaited cycle.

3.8.4 Megabus Network Operation

This subsection describes the bus cycle timing and controls for the handshaking technique that establishes communications between memory and other units on the Megabus. Bus dialog is completely asynchronous, and each bus cycle can be considered as an independent handshaking sequence between a master unit and a slave unit.

A Write operation requires one bus cycle during which the unit wanting to write into memory is the master unit and memory is the slave unit.

Reading a single word from memory requires two bus cycles. During the first cycle, the unit requesting information is the master unit and memory is the slave unit. During the second cycle when memory returns the requested information, memory is the master unit and the unit that requested the information is the slave unit.

Reading a double-word from memory requires three bus cycles. During the first cycle, the unit requesting the double-word from memory is the master unit and memory is the slave unit. Memory returns the double-word during two subsequent Megabus cycles, returning one word per cycle. During this time, memory is the master unit and the unit that requested the double-word is the slave unit.

To implement handshaking, all major logic boards that comprise the units of the sytem have similar system bus cycle control logic. Figure 3-29 shows the general timing relationships of the handshaking signals between a master unit and a slave unit.

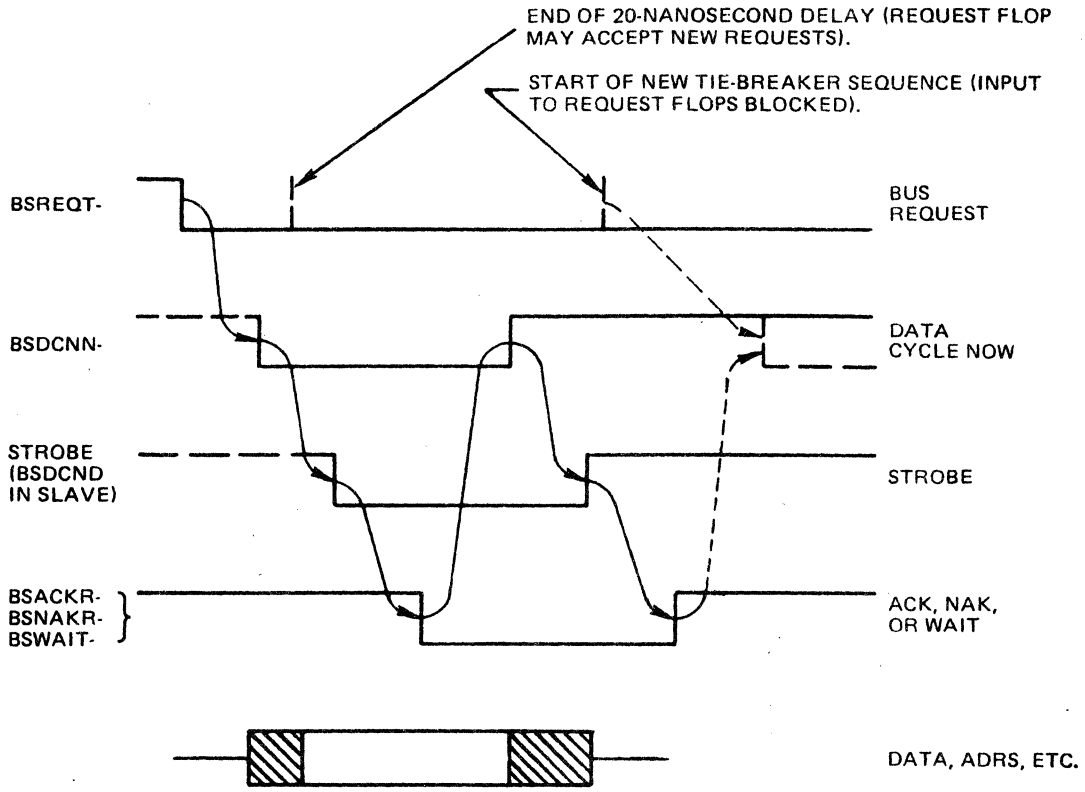


Figure 3-29 Megabus Timing Diagram

The Bus Cycle Request (BSREQT-) signal is common to all units on the Megabus. When true, BSREQT- indicates that at least one unit is requesting a bus cycle. If BSREQT- is the result of simultaneous requests, the tie-breaking logic in each unit establishes priority by granting the bus cycle to a specific unit on the basis of its physical position. In any given system, memory has the highest priority and the CPU has the lowest priority. The remaining units have intermediate priorities based on their relative physical position between the memory and the CPU.

When a bus cycle is granted, the Bus Data Cycle Now (BSDCNN-) signal is driven true, which indicates that the tie-breaking function is completed and a specific unit was granted master status (gained bus control). At this time, data and address information for the slave unit is placed on the bus. BSDCNN+ is delayed approximately 60 nanoseconds at the slave unit, providing for bus skew correction time and allowing the slave unit time to compare its internally stored address or channel number to the address on the bus. The slave unit then develops an internal Strobe (MYSTBB-) from BSDCNN+ and returns one of three response (BSACKR-, BSWAIT-, or BSNAKR-) signals to the master unit, which completes the handshaking sequence (i.e., the slave unit acknowledges the communication as being accepted, postponed, or denied, respectively). If no acknowledgement is received by a master unit (i.e., no response signal is generated), the Dead Man Time-Out operation (a function of the CPU) allows a delay of 5 microseconds to occur before the CPU clears the bus.

3.8.4.1 Bus Request and Tie-Breaking Logic

Figure 3-30 shows the logic each memory requires to request a bus cycle, perform tie-breaking, and generate a Data Cycle Now signal. All other units on the Megabus have similar logic for the handshaking function. Also shown in Figure 3-30 is the logic which alters memory's handshaking function during a Double-Word Fetch operation.

3.8.4.1.1 Bus Request Generation

Bus requests are made by memory only when memory is ready to transmit previously requested data (i.e., during the initiation of a Second Half Bus Cycle).

When memory accepts a read request and is not performing a Refresh cycle, the timing generator logic generates the clock function DCNNGO-TE. DCNNGO- sets function Stored Request true (see Figure 3-30). If there are no other bus cycle requests pending (BSREQT- false), no data cycles are in progress (BSDCNN- false), and the system is not clearing all logic from initialization (BSMCLR- false), then the Bus Busy function is false. The Stored Request function sets the Request flip-flop via the Request Set gate. The Request flip-flop sends the function My Request to the bus tie-breaking network to compare the priority of this request to other possible simultaneous requests (if any). At the same time, My Request is sent to the bus transceivers where it is inverted to become the Bus Request (BSREQT-) function on the Megabus.

When BSREQT- comes true, it prevents (via the Bus Busy function) any other stored requests from setting other Request flip-flops in the system. BSREQT- also brings the Bus Busy Delayed (BBUZ20+) function true, which provides a delay of approximately 20 nanoseconds to allow the Request flip-flops time to stabilize before tie-breaking signals are generated via the Request Flip-Flop Set gates.

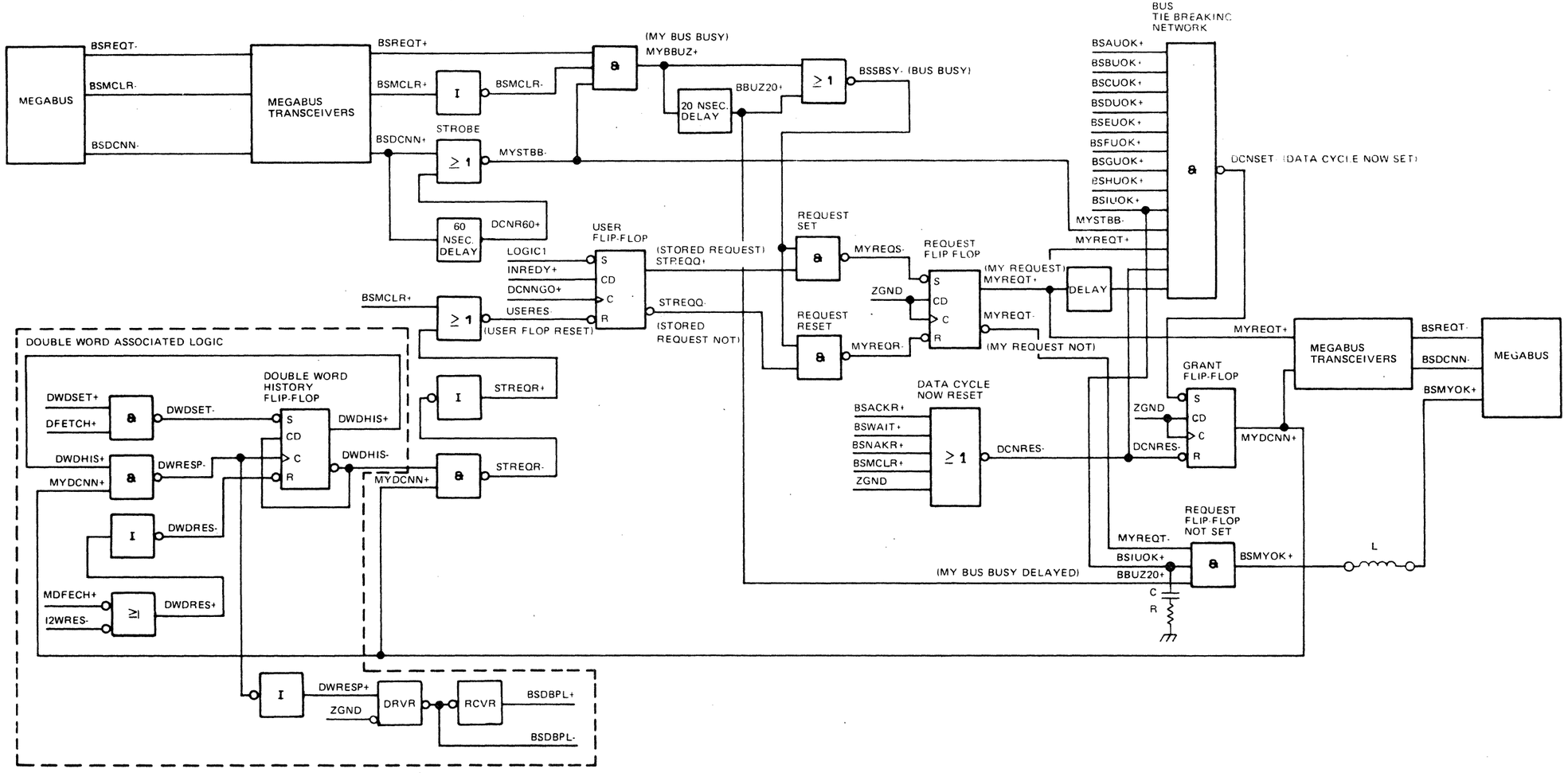


Figure 3-30 Memory Bus Cycle Request and Tie-Breaking Logic

3.8.4.1.2 Request Tie-Breaking Logic

Since any unit can request a bus cycle by setting its User flip-flop and Request flip-flop, more than one Request flip-flop can be set at any given time, each representing a possible future bus cycle. When there are simultaneous requests, the bus tie-breaking network grants a data cycle to the highest priority requesting unit by setting the appropriate Grant flip-flop.

To grant a data cycle to any unit, the bus tie-breaking network must have all of its input tie-breaking signals (BSAUOK+ through BSIUOK+) true (see Figure 3-30). Each unit drives the output of its Request Flip-Flop Not Set gate false when making a bus request. BSMYOK+ is sent false to the bus backplane connectors where it becomes a disabling signal for the tie-breaking gates on lower priority units.

Figure 3-31 shows how BSMYOK+ becomes the BSAUOK+ input for the bus tie-breaking network in the next lower priority unit, BSBUOK+ in the second lower priority unit, BSCUOK+ in the third lower unit, etc., and BSIUOK+ in the tenth lower priority unit.

Since there could be more than 10 units on the Megabus, BSIUOK+ is used to generate BSMYOK+ in the tenth lower priority unit, avoiding the loss of a bus request indication from a tenth higher priority unit in the eleventh lower unit. Thus, BSAUOK+ on the eleventh lower priority unit's bus tie-breaking network could result from a bus request from either the tenth lower priority unit, the highest priority unit, or both.

Memory always occupies the Unit 1 (see Figure 3-31) position on the backplane connectors. In this position, the tie-breaking signals are tied to logic One signals from pull-up resistors. Unless there is a higher order memory controller in the system when memory generates MYREQT+, there is no false tie-breaking signal, which would prevent the bus tie-breaking network from generating the Data Cycle Now Set function true to set the Grant flip-flop.

3.8.4.1.3 Data Cycle Now Generation

When memory requests a bus cycle (MYREQT+) and is the highest priority requesting unit, the Data Cycle Now Set function is sent true from the bus tie-breaking network. The Data Cycle Now Set function sets the Grant flip-flop, which generates the My Data Cycle Now (MYDCNN+) function (see Figure 3-30). MYDCNN+ is inverted through the bus transceivers and sent out on the system bus as BSDCNN-. MYDCNN+ resets the User flip-flop (unless there is a double-word transfer; refer to subsection 3.8.4.1.4). MYDCNN+ also gates data, module identification codes, and some controls on to the Megabus.

3.8.4.1.4 128K Double-Word Fetch Operation

During a 128K Double-Word Fetch cycle, a requesting unit informs memory that a double-word is requested. The timing generator (refer to subsection 3.3) and a portion of the bus control circuitry enables memory's response with two words rather than one, described as follows.

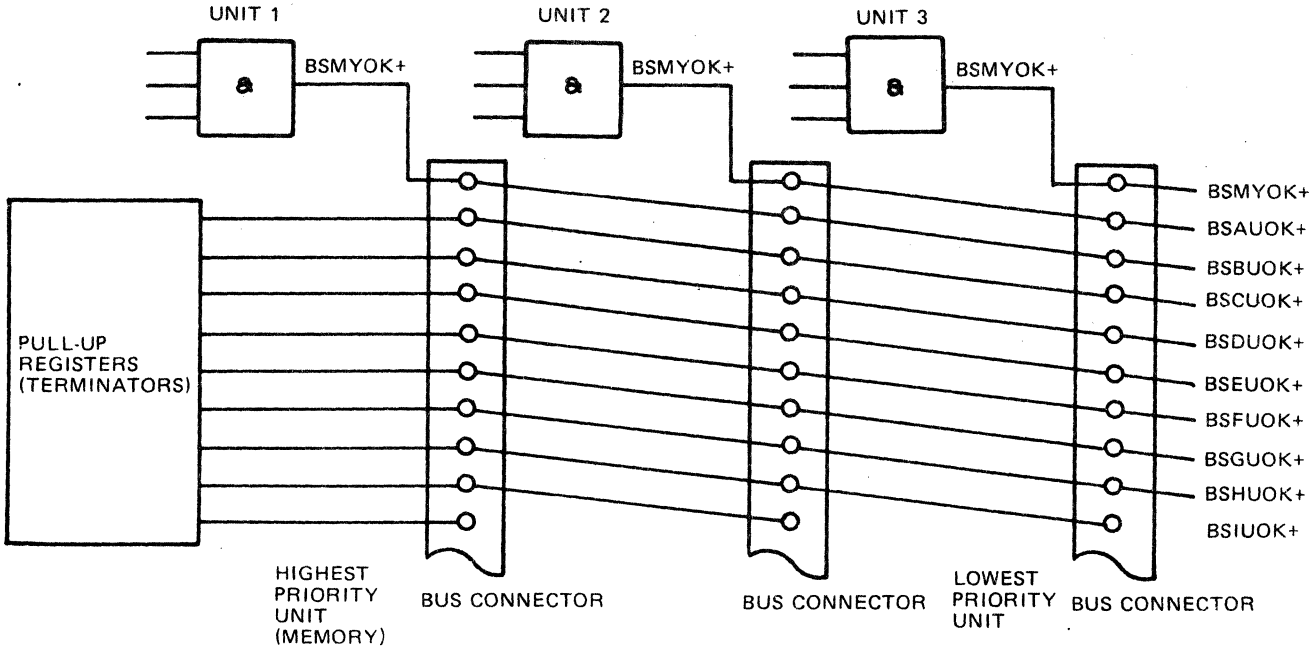


Figure 3-31 Tie-Breaking Network Signals

Figure 3-30 shows the bus control and response logic used for double-word transfers. During single-word transfers, function MYDCNN+ is generated when memory is granted a bus cycle, sending back the requested data word. The memory User flip-flop is reset on the leading edge of MYDCNN+, and the Stored Request Not function resets the memory's Request flip-flop. Function My Request Not frees the bus for the next operation.

When memory accepts a double-word read request, the Double-Word Fetch (DFETCH+) function is generated (see Figure 3-32), provided memory is not in a Refresh cycle (REFCOM- is false).

The Double-Word Fetch function enables memory to generate two successive MYDCNN+ signals, which send out two data words requested by the master unit, described as follows.

The coincidence of the Double-Word Fetch and the My Acknowledge functions sets the Double-Word History flip-flop (see Figure 3-30). When memory responds with MYDCNN+ and the first requested data word, function DWDHIS- true prevents MYDCNN+ from resetting the User flip-flop via function User Flop Reset. Consequently, the Request flip-flop is not reset and MYREQT+ remains true.

The leading edge of MYDCNN+ generates the clock function Double-Word Response (DWRESP-). Function Double-Word Response is inverted and sent out on the bus as BSDBPL-, which informs the requesting unit that a second data word is coming when a second MYDCNN+ is generated. Function Double-Word Response also resets the Double-Word History flip-flop, generating DWDHIS- false so that the next MYDCNN+ function can reset the User flip-flop.

The requesting unit acknowledges the first data word by responding with BSACKR-, which resets the memory Grant flip-flop. (If the requesting unit sends a Negative Acknowledge signal to memory or does not respond, the data is lost.)

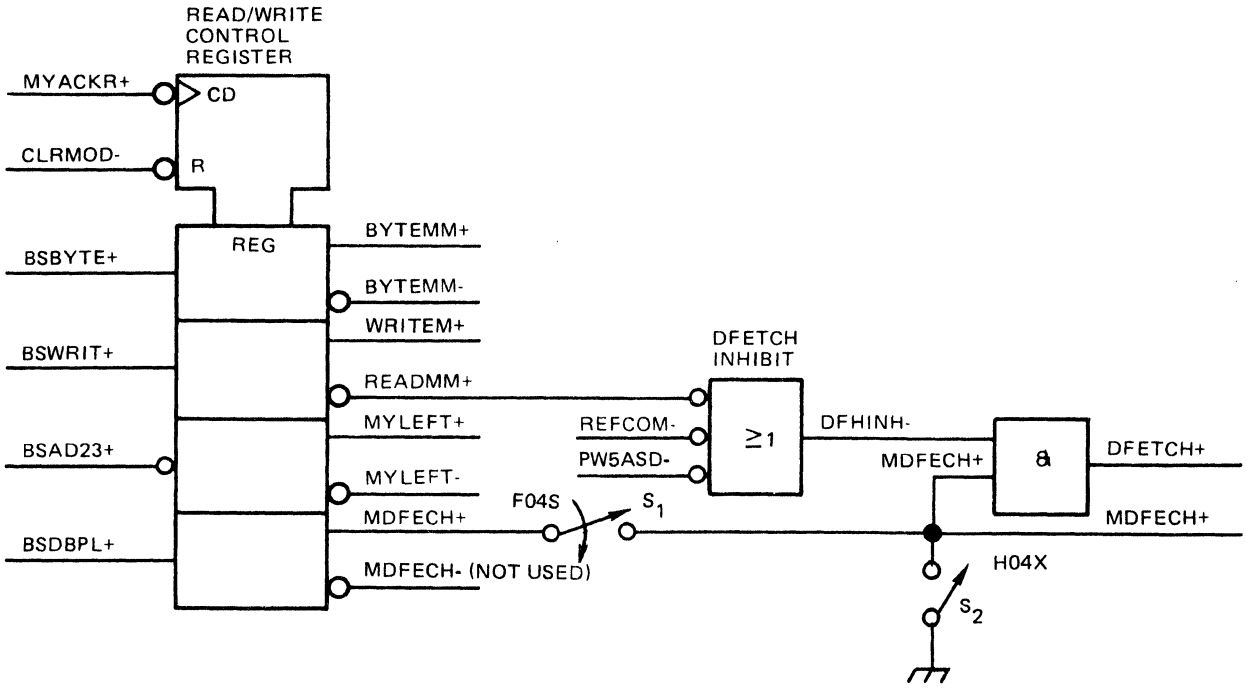


Figure 3-32 DFETCH Signal Generation

Since the Request flip-flop is not reset, memory still retains control of the bus via BSREQT. It therefore generates another MYDCNN+ via the bus tie-breaking network and the Grant flip-flop to send the second data word.

NOTE

In this case, the tie-breaking circuitry is used simply to generate function DCNSET-. It is not used to establish request priorities on the bus.

Since the Double-Word History flip-flop is reset, the second MYDCNN+ resets the User and Request flip-flops. Also, BSDBPL- is not driven true, indicating that no other information should be expected by the requesting unit.

If for some reason a second data word cannot be obtained (e.g., if a request for a double-word addressed the highest location in memory), the memory inhibits (resets) the Double-Word History flip-flop via function Inhibit Double-Word Response (I2WRES). Memory does not drive BSDBPL- true with the first data word delivery, indicating to the requesting unit that a second word is not coming.

Figure 3-33 shows the timing relationships between signals during double-word MYDCNN+ generation.

Certain conditions exist during Double-Word Fetch Memory cycles when the memory cannot send back two data words. These conditions are when a memory read request is made on boundary locations where the second word is nonexistent. Such cases exist at memory locations 0FFFF on a 64K controller, and at 1FFFF on a 128K controller.

If a Double-Word Pull cycle is requested at these locations, function Inhibit Second Word Response (I2WRES-) goes true to reset the double-word history register (see Figure 3-30), and this inhibits the sending out of the second word (see Figure 3-33).

When the 128K memory controller is configured as a single-word pull memory, as would be the case if a Memory-Pac failed and it was advantageous to remove the Memory-Pac but still operate the system, the double-word pull functionality must be disabled. This is done by placing the H04X switch position 02 to ON and switch F04S position 01 to OFF (see Table 1-2).

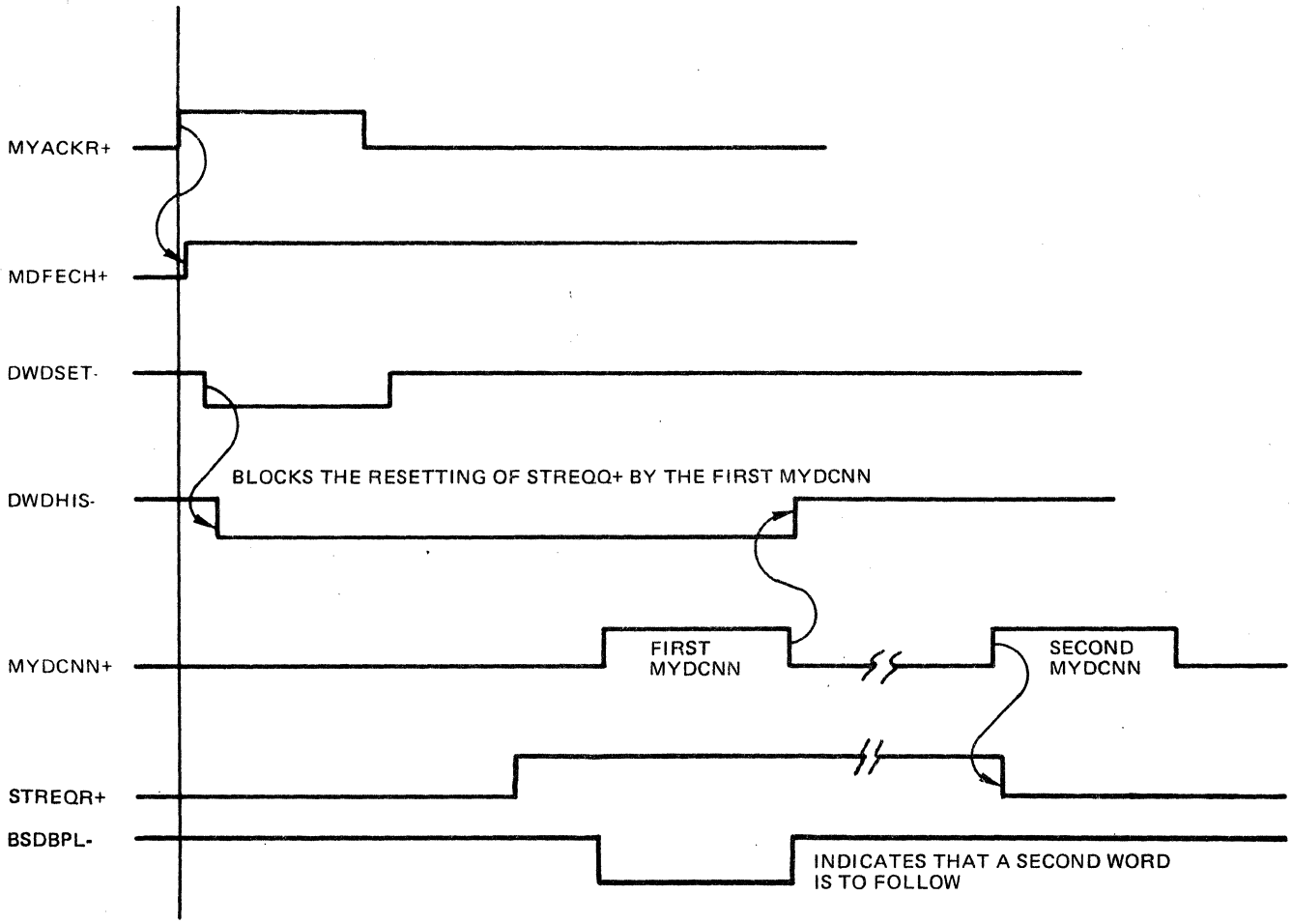


Figure 3-33 Main Memory Double-Word MYDCNN+ Generation Timing Diagram

3.8.4.2 Bus Response Logic

When a master unit requests a bus cycle to transfer information to or from memory, memory normally returns one of three responses: Acknowledge (ACK), Wait (WAIT), or Negative Acknowledge (NAK), completing the handshaking sequence (i.e., memory acknowledges the communication as being accepted, postponed, or denied). The response signals are defined as follows:

1. ACK (BSACKR-): Memory accepts the bus cycle (refer to subsection 3.2.2.1).

2. WAIT (BSWAIT-): Memory receives a request for a bus cycle when it is busy (refer to subsection 3.8.4.2.2).

3. NAK (BSNAKR-): Memory receives a bus cycle that requests a Lock operation while processing a previously requested Lock operation (refer to subsection 3.8.4.2.3).

Figure 3-34 shows the memory logic used to generate the three response (ACK, WAIT, and NAK) signals. Memory's generation of the proper response is controlled by three AND gates: ACK Generate, WAIT Generate, and NAK Generate. Signals are sent to these gates in such a manner that only one gate can be made during any one Response cycle. Three input signals are common to all three gates: MYADGO-, MODPAR-, and RESPIN+.

The output of the Address Go (MYADGO-) gate comes true when the address presented to the memory module compares to the settings of the module select switches on the controller. This indicates that the correct memory module identification code is on the bus address lines (BSAD00- through BSAD07-, BSAD21-, and BSAD22-) and that memory is being referenced (BSMREF- true).

Signal Module Parity (MODPAR-) comes true when the parity on bus address lines 0 through 7 (module address) is correct.

Signal Response Inhibited (RESPIN+) goes false when the Memory-Pac being addressed is present on the memory controller.

If any of the conditions in the three preceding paragraphs is not satisfied, the ACK, WAIT, and NAK Generate gates are not made, and memory does not issue a response. In this case, the Dead Man Time-Out function in the CPU detects no response on the bus. After 5 microseconds, it issues an NAK response to the requesting unit on behalf of the entire system. If all of the conditions in the three preceding paragraphs are satisfied, memory can issue an ACK, WAIT, or NAK response as described in subsections 3.8.4.2.1 through 3.8.4.2.3.

3.8.4.2.1 Acknowledge (ACK) Generation

If memory is capable of accepting a bus request from a master unit, it acknowledges the master unit's request with the Acknowledge (BSACKR-) response. Figure 3-34 shows the logic used to generate the Acknowledge response. The following five conditions must be satisfied before memory can issue an Acknowledge response.

1. The memory module must be correctly addressed (MYADGO- true; refer to subsection 3.8.4.2).

2. Parity for the memory module address must be correct (MODPAR- true; refer to subsection 3.8.4.2).

3. The Memory-Pac addressed must be connected to the memory controller (RESPIN+ false; refer to subsection 3.8.4.2).

4. The memory Wait flip-flop must be reset (WAITXX+ false; refer to subsection 3.8.4.2.2).

5. Memory must not be receiving a Lock request while a Lock operation is already in process (NAKSEC+ false, refer to subsection 3.8.4.2.3).

When each of the preceding five conditions is satisfied, signal ACKGEN+ comes true via the ACK Generate gate and is fed to the memory response register. The memory response register generates MYACKR+ true when the Data Cycle Now Delayed (DCNR60+) signal arrives at the memory response register. MYACKR+ is inverted through the bus transceivers and set out on the bus as BSACKR-.

3.8.4.2.2 Wait (WAIT) Generation

If memory receives a request when busy, it issues the Wait (BSWAIT-) response, indicating that the master unit should try to access memory at a later time.

Figure 3-34 shows the logic used to generate the Wait (BSWAIT-) response. Similar to BASCKR- generation, five conditions must be satisfied before memory can issue the Wait response. Four of these five conditions are the same conditions required for ACK generation: MYADGO- true, MODPAR- true, RESPIN+ false, and NAKSEC+ false (refer to subsection 3.8.4.2.1). The fifth input (WAITXX-) indicates that the memory Wait flip-flop is set. The Wait flip-flop is set on the leading edge of BSDCNN-, which generates function WAITXX- true, when any one of the following conditions is true:

1. Memory is busy doing a cycle (MEMBUZ- true).
2. Memory is obtaining the bus for a Second Half Read Cycle (STREQQ- true).
3. Memory is in an Initialize mode (INITMM- true).
4. Memory is anticipating a Refresh cycle (REREQA- true).

When WAITXX- comes true and each of the five input conditions to the WAIT Generate gate is satisfied, signal WAITGE+ comes true and is fed to the input of the memory response register. The memory response register generates MYWAIT+ true when the Data Cycle Now Delayed (DCNR60+) signal arrives at the clock input to the memory response register. MYWAIT+ is inverted through the bus transceivers and sent out on the bus as BSWAIT-.

3.8.4.2.3 Negative Acknowledge (NAK) Generation and Lock Operation

The only time memory generates the Negative Acknowledge (BSNAKR-) response is when it is processing a transfer that involves a Lock operation. The Lock operation allows a requesting master unit temporary privileged use of certain reserved storage locations. A master unit gains access to the reserved storage locations by sending memory the proper Lock code.

NOTE

Allocation of reserved storage locations and control of the Lock codes are handled external to the memory.

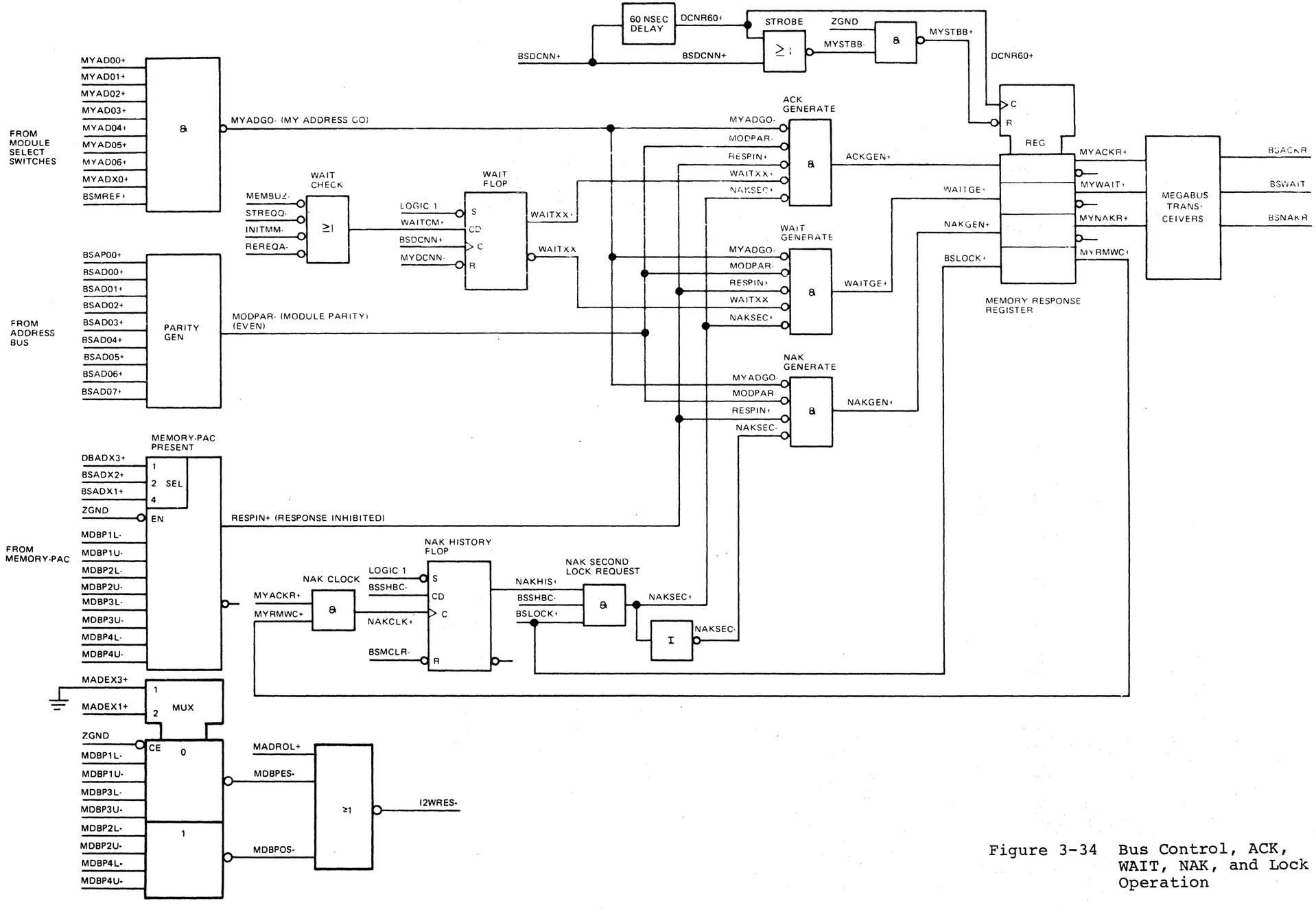


Figure 3-34 Bus Control, ACK, WAIT, NAK, and Lock Operation

Once accepted as the privileged unit, the master unit may perform Read Modify Write, Multiword Write, Multiword Read, or other combinations of cycles at the reserved storage locations. Other units requesting access to the reserved locations are denied access until the privileged unit completes its sequence of cycles. Other units requesting access to nonreserved locations are granted access even though the privileged unit has not completed its sequence of cycles.

If memory receives a bus cycle requesting a Lock operation while processing a previously requested Lock operation, it must issue a Negative Acknowledge (NAK) response to the unit requesting the second Lock operation. NAK indicates that memory cannot accept the bus cycle and that the requesting unit should not attempt a retry.

Figure 3-34 shows the logic used to implement the Lock operation and generate the Negative Acknowledge (BSNAKR-) response. A unit requests a Lock condition by sending signals Bus Lock (BSLOCK-) true and Second Half Bus Cycle (BSSHBC-) false with its request for a bus cycle. These signals specify a Test and Set Lock condition (refer to subsection 3.8.1.3).

NOTE

BSLOCK- is inverted through the bus transceivers so that it appears as BSLOCK+ in Figure 3-34.

When memory receives a bus cycle request with a Test and Set Lock condition, it issues an Acknowledge response via its acknowledge generation logic (refer to subsection 3.8.4.2.1), provided all five input conditions on the ACK Generate gate are satisfied.

NOTE

One of the ACK Generate gate inputs (NAKSEC+) must be false for the Acknowledge response. This signal indicates that memory is not processing a previously requested bus cycle with a Lock operation.

When the bus request is acknowledged, BSLOCK+ drives Read Modify Write Control (MYRMWC+) true through the memory response register. MYRMWC+ is gated with MYACKR+ on the NAK Clock gate, providing a clock signal for the NAK History flip-flop. Since BSSHBC- is false during a Lock request, the NAK History flip-flop is set, generating NAKHIS+ true.

If memory now receives a subsequent bus cycle request with a Test and Set Lock condition, the NAK Second Lock Request gate drives NAKSEC+ true, inhibiting the ACK Generate and WAIT Generate gates. At the same time, NAKSEC+ is inverted, and NAKSEC- true enables the NAK Generate gate, provided the other three conditions on this gate are satisfied (refer to subsection 3.8.4.2). The output of the NAK Generate gate is fed to the input of the Memory Request register, which generates MYNAKR+ true when the Data Cycle Now Delayed (DCNR60+) signal comes true on the clock input of the memory response register. MYNAKR+ is inverted through the bus transceivers and sent out on the bus as BSNAKR-, generating a Negative Acknowledge signal on the second Lock Bus cycle.

The NAK History flip-flop remains set until a cycle is initiated with signals BSLOCK- and BSSHBC- true (refer to the Reset Lock condition described in subsection 3.8.1.3). Since BSSHBC- is true, the NAK Second Lock Request gate is inhibited, driving NAKSEC+ false. NAKSEC+ false now inhibits the NAK Generate gate and allows the ACK or WAIT Generate gates (assuming the other input conditions are satisfied). The coincidence of signals MYACKR+ and MYRMWC+ (from BSLOCK- true) resets the NAK History flip-flop.

3.9 QUALITY LOGIC TEST CIRCUITRY

Each unit connected to the Megabus has the potential to run an internal logic test during a Master Clear (BSMCLR-) command or during system power up. Memory does not perform a physical Quality Logic Test (QLT), but it supports the QLT functionality as shown in Figure 3-35.

Memory receives the Quality Logic Test In (BSQLTI-) signal true, indicating that all units of lower priority completed their resident QLTs, and activates its QLT input to mask its Initialize mode. When the system is powered up (PWONLL+), the memory initialize logic increments the memory address register to its high-order count (128K), ending the Initialize mode. When the Initialize function is completed, memory notifies the QLT circuitry via MADROL-00, which drives the QLT Out (BSQLTO-) signal true, notifying the next higher priority unit (another memory controller) that all units of lower priority completed their resident logic tests.

Subsequent memory QLT activity is a function of the Memory Save Power Supply option. Systems without the option perform the QLT function each time the system powers up. Systems with the option perform the QLT function on the first system power up only.

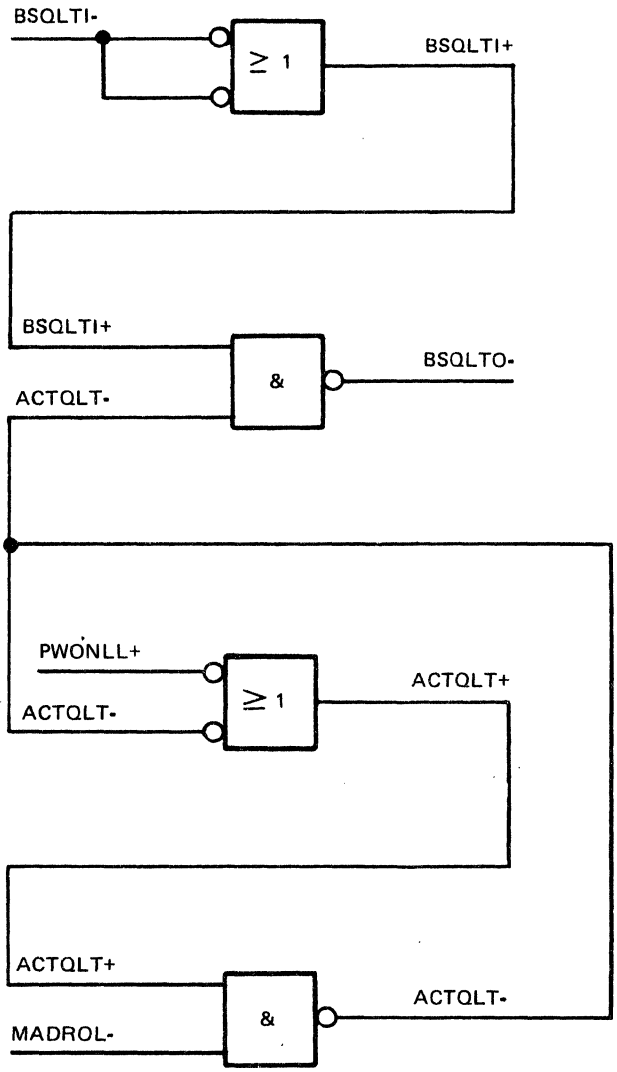


Figure 3-35 Quality Logic Test

PLEASE FOLD AND TAPE –
NOTE: U. S. Postal Service will not deliver stapled forms

M&TO HARDWARE PUBLICATIONS
USER COMMENTS FORM

DOCUMENT TITLE: _____

PART NO.: _____

ORDER NO.: _____

ERRORS:

HOW DO YOU USE THIS DOCUMENT?

THEORY _____ ☐

MAINTENANCE _____ ☐

TROUBLESHOOTING ☐

OTHER: _____

_____

DOES THIS MANUAL SATISFY YOUR REQUIREMENTS?

YES ☐ NO ☐

IF NOT, PLEASE EXPLAIN _____

_____

_____

FROM: NAME _____ DATE _____

TITLE _____

COMPANY _____

ADDRESS _____

FIRST CLASS
Permit No. 39531
Waltham, Ma.

Business Reply Mail NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

HONEYWELL INFORMATION SYSTEMS INC.
200 SMITH STREET
WALTHAM, MA. 02154

MAIL STATION 872A
HARDWARE PUBLICATIONS, BILLERICA

Honeywell

1 Meg. memory boards (BMMU033B-002)

HONEYWELL CONFIDENTIAL AND PROPRIETARY

2

Honeywell

CUSTOMER SERVICE

Technical Service Bulletin

FILE CONTROL		ORIGIN	ISSUE DATE	EXP. DATE
[1] Level 6/DPS6	[2] Memory	NWT	11 NOV 83	Indef
SUBJECT: Memory Configuration for BF2MZE Controller - Level 2.2 only - BMMU033B-002			[4] DOSSIER 709.03	[3] NO. REV. NWT-019A

The level 2.2 board starts shipping in November. This board eliminates previous restricted memory pac configurations. Table 1 indicates the allowable pac configurations for all board levels.

TABLE 1

BOARD LEVEL	*ENGINEERING NUMBER	IPI NUMBER	ADAPTER CONFIGURATIONS 32Kwd.	128Kwd.
2.0	60146280	BMMU033A-001	2 or 4 pacs	1, 2 or 4 pac
2.1	60147897	BMMU033A-002	2 or 4 pacs	1, 2 or 4 pacs
2	60156198	BMMU033B-002	1, 2, 3 or 4	1, 2, 3 or 4 pacs

* Eng. No. is etched on the edge of the board.

	BOARD NAME	IPI NUMBER
32K word pac	BS2STF	BCMM044A
	BS2SH4	BCMM044A
8K word pac	BS2ST4	BCMMST4A

This TSB indicates the switch and jumper configurations for the level 2.2 board only. Configurations for the other board levels is covered in the systems level T&RP manual (K910).

T&V CMMX5 Rev. E has been updated to test all memory pac configurations. It will be released with Tacpac Revision H in November. CMMX5 Rev. D can be utilized when necessary but only for 2 or 4 pac configurations.

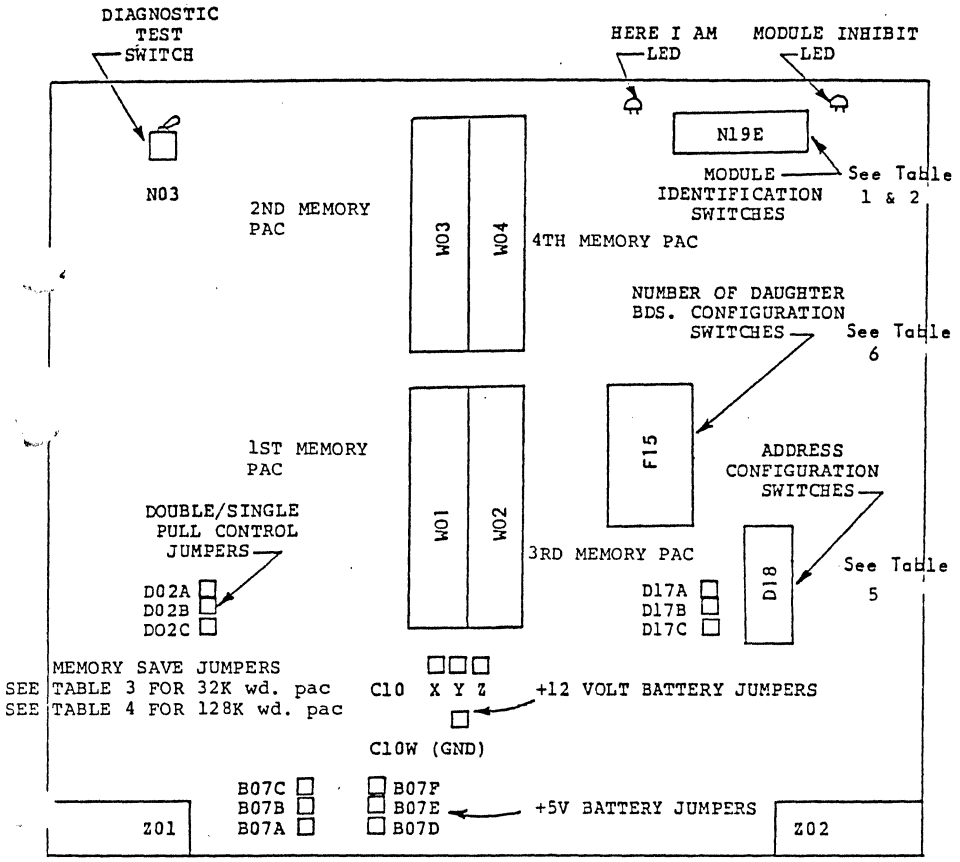
By: Joe Hernon

Approval:

DISTRIBUTION: Standard	PAGE 1/5

Level 6/DPS6	Memory	3 NWT-019A	4 709.03

Figure 1 BF2MZE MEMORY CONTROLLER - Level 2.2



COMPONENT SIDE

Table 1 Module Identification

N19E SWITCH POSITION FOR 32K word pac

1=ON 0=OFF	S1 *	S2	S3	S4	S5	S6	S7	S8	S9	S10	MODULE ADDRESS RANGE WHEN USING BS2SH4 OR BS2STF
1	0	0	0	0	0	0	0	0	0	0	0-128K
2	0	0	0	0	1	0	0	0	0	0	128K-256K
3	0	0	0	0	0	1	0	0	0	0	256K-384K
4	0	0	0	0	1	1	0	0	0	0	384K-512K
5	0	0	0	0	0	0	1	0	0	0	512K-640K
6	0	0	0	0	1	0	1	0	0	0	640K-768K
7	0	0	0	0	0	1	1	0	0	0	768K-896K
8	0	0	0	0	1	1	1	0	0	0	896K-1024K
9	0	0	0	0	0	0	0	1	0	0	1024K-1152K
10	0	0	0	0	1	0	0	1	0	0	1152K-1280K
11	0	0	0	0	0	1	0	1	0	0	1280K-1408K
12	0	0	0	0	1	1	0	1	0	0	1408K-1536K
13	0	0	0	0	0	0	1	1	0	0	1536K-1664K
14	0	0	0	0	1	0	1	1	0	0	1664K-1792K
15	0	0	0	0	0	1	1	1	0	0	1792K-1920K
16	0	0	0	0	1	1	1	1	0	0	1920K-2048K

* To inhibit module response, place S1 in the "ON" position. This function will be indicated by a LED which is shown in Figure 1.

TABLE 3 WITH MEM. SAVE OPTION

JUMPER	
From	To
B07A	B07B
B07D	B07E
C10X	C10Y

W/O MEM SAVE OPTION

JUMPER	
From	To
B07B	B07C
B07E	B07F
C10Y	C10Z

[1] Level 6/DPS6	Memory	[3] NWT-019A	[4] 709.03

Table 2 Module Identification

1 = Down

N19E SWITCH POSITION FOR 128K word pac

1=ON 0=OFF	S1 *	S2	S3	S4	S5	S6	S7	S8	S9	S10	MODULE ADDRESS RANGE WHEN USING BS2ST4
1	0	0	0	0	1	1	0	0	0	0	0-512K
2	0	0	0	0	1	1	1	0	0	0	512K-1024K
3	0	0	0	0	1	1	0	1	0	0	1024K-1536K
4	0	0	0	0	1	1	1	1	0	0	1536K-2048K
5	0	0	0	0	1	1	0	0	1	0	2048K-2560K
6	0	0	0	0	1	1	1	0	1	0	2560K-3072K
7	0	0	0	0	1	1	0	1	1	0	3072K-3584K
8	0	0	0	0	1	1	1	1	1	0	3584K-4096K
9	0	0	0	0	1	1	0	0	0	1	4096K-4608K
10	0	0	0	0	1	1	1	0	0	1	4608K-5120K
11	0	0	0	0	1	1	0	1	0	1	5120K-5632K
12	0	0	0	0	1	1	1	1	0	1	5632K-6144K
13	0	0	0	0	1	1	0	0	1	1	6144K-6656K
14	0	0	0	0	1	1	1	0	1	1	6656K-7168K
15	0	0	0	0	1	1	0	1	1	1	7168K-7680K
16	0	0	0	0	1	1	1	1	1	1	7680K-8192K

* To inhibit module response, place S1 in the "ON" position. This function will be indicated by a LED which is shown in Figure 1.

TABLE 4 WITH MEMORY SAVE OPTION

JUMPER	
From	To
B07A	B07B
B07D	B07E
C10Y	C10W

W/O MEMORY SAVE OPTION

JUMPER	
From	To
B07B	B07C
B07E	B07F
C10Y	C10W

1 Level 6/DPS6	Memory	3 NWT-019A	4 709.03

TABLE 5 D18 SWITCH SETTINGS FOR LEVEL 2.2

128K Word Pac	64K RAM BDS.									
NUMBER OF BDS.	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
1 OR 3	0	0	1	0	0	0	1	0	1	1
2 OR 4	0	0	0	0	1	0	1	0	0	1

32K Word Pac	16K RAM BDS.									
NUMBER OF BDS.	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
1 OR 3	1	1	0	1	0	1	0	0	1	0
2 OR 4	1	1	0	0	1	1	0	0	0	0

Table 6 F15 SWITCH SETTINGS FOR LEVEL 2.2

EITHER 32K OR 128K word pac					
NUMBER OF BDS.	S1	S2	S3	S4	
1 OR 3	0	1	1	0	
2 OR 4	1	0	0	1	

SPECIAL JUMPERS

128K word pacs

Location D17
Jumper B to C 1 OR 3 Bds.
B to A 2 OR 4 Bds.

Jumper D17 dont't care when 32K word pacsused.

Location D02.
Jumper B to A 1 OR 3 Bds. SWP
B to C 2 OR 4 Bds. DWP

Honeywell

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.