1993 5238

Burroughs Corporation
MICRO COMPONENTS GROUP
MEMORY ENGINEERING

RANCHO BERNARDO PLANT

V500 MEMORY DATA CARD

COMPANY CONFIDENTIAL

SYSTEMS DESIGN SPECIFICATION

REVISIONS

REV. LTR	REVISION ISSUE DATE	PAGES REVISED, ADDED, DELETED OR CHANGE OF CLASSIFICATION	PREPARED BY	APPROVED BY
A		Review copy (not released)	L. Simpson	D. Castle
B	10/27/86 ECN 52053	Incorporate new CTRL-ARRAY modifications including busy timeout counter Initial Issue	L. Hanson M. Fitzsimmons	D. Lau H. Li R. Young D. Welbourn J. Chen 11-10-86 Rancho Bernardo: M. Fitzsimmons J. Rudy R. Rhodes C. Williams

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

TABLE OF ILLUSTRATIONS

1 PURPOSE

This specification defines the Memory Data Card (MDC) from which the memory storage structure of the V500 System is constructed.

2 APPLICABLE DOCUMENTS

1257 6005	Corporate Environmental Standards
A1, A14, A16	Corporate Technical Standards
xxxx xxxx	Design Guidelines
1993 5279	V500 Architecture
1993 5162	V500 System
1993 5170	V500 System Index
1997 5390	V Series Instruction Set
1993 5204	V500 Execute Module (XM)
1993 5212	V500 Fetch Module
1993 5220	V500 Memory Control and Cache Module
xxxx xxxx	Inter-Cabinet Buffer Module
1993 5253	I/O Memory Concentrator
1993 5246	I/O Translator
1993 5329	Data Transfer Module
1993 5337	Fault Detection
1993 5295	System Maintenance Controller
1993 5303	V500 Maintenance Subsystem

3 OVERVIEW

The V500 Processor memory storage structure consists of 1 to 16 Memory Data Cards (MDC). Using 256K DRAM technology, each MDC will contain either 524288 20 byte words (half populated), or 1048576 20 byte words (fully populated), plus error correct bits. The words are partitioned such that in certain cases, up to 4 bits in error can be corrected. Data from the output of the card may be corrected in a similar fashion by the requestors. In addition, 1 bit in error on the address and control may be corrected.

The total capacity of the memory subsystem, using 256K DRAMs, is 335,544,320 usable bytes. It is expandable to this quantity in increments of 10,485,760 or 20,971,520 bytes. If 1M DRAMS are used, the capacity may be increased to 1,342,177,280 bytes, with a corresponding increase in expansion increments.

Each memory data card will have a physical location and a logical address. After power-up-clear, each memory data card must be configured to the proper logical address by the maintenance processor. It is important to make the distinction between the physical slot location and the logical address. Each memory data card responds to its logical address. The physical slot location is only used for reporting errors so that the field engineer will be able to replace a failing board.

4 FUNCTIONS

The MDC is organized as four 40 data bit wide sub-modules, each with its own error correction (SEC-DED-S4EC), giving a 48 bit physical sub-module. Each sub-module spans 48 memory SIPs, so that any total failure of one SIP can be corrected.

4.1 COMMANDS

The MDC supports 4 different operations: read-public, read-private, read-modify-write, and write. These commands are described below.

READ-PUBLIC

Read a 40 digit block of memory from the location specified on the address bus. The "public" distinction is only recognized by the MCACM. This command allows MCACM's to keep cached copies of unmodified data. The read-public command is used by cache, in response to a read miss, and by the IOMC.

READ-PRIVATE

Read a 40 digit block of memory from the location specified on the address bus. The "private" distinction is only recognized by the MCACM. This command forces all other MCACM's to release cached copies of this data. The read-private command is only used by cache, and only in response to a write miss.

READ-MODIFY-WRITE

The read-modify-write command is similar to the read-private command. It reads a 40 digit block of memory at the address specified on the address bus. The difference is, this command will force the MDC to turn on MODULE BUSY, and hold it on until the next command. The read-modify-write command is only used by the IOMC, which can read a 40 digit block of data from memory (if cached, from MCACM), modify it, and then write it back to memory.

If the read-modify-write command cycle exceeds 32 ECL system clocks, the MDC will reset the busy condition. This prevents memory requestors from being locked-out by a broken IOMC.

WRITE

Write the 40 digit block of memory from the data bus into the location specified on the address bus.

4.2 INTERNAL REFRESH

Refresh is generated internally on the MDC. There is a MDC busy line from each MDC to all requestors that must be asserted two bus clocks (four system clocks) before a MDC can start a refresh cycle. In the event that a module request is in transit on the bus at the same cycle as the busy signal, this request must be serviced, the MDC will remain busy and do the refresh later.

The MDC will contain sufficient power supply filtering to allow refresh without causing the voltage to drop below the necessary levels for proper operation.

5 INTERFACES

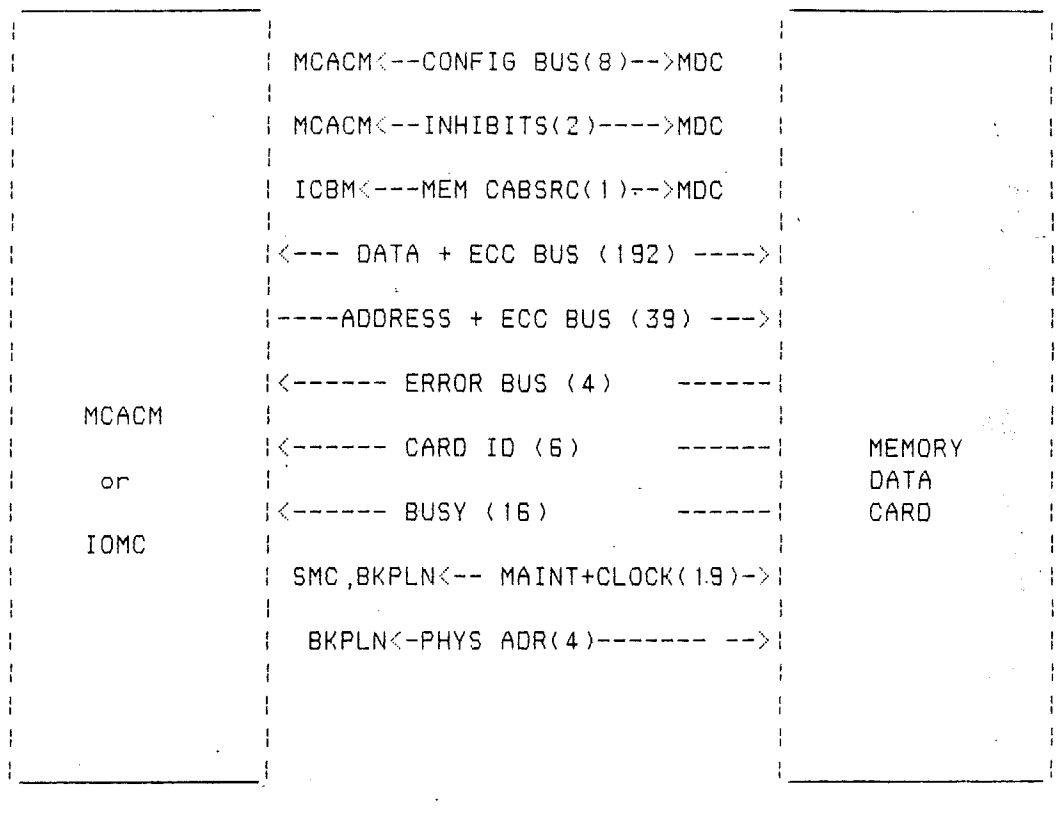


FIGURE 5.1 MEMORY DATA CARD BACKPLANE INTERFACE

5.1 ADDRESS BUS

The address bus contains 4 fields: command, requestor I.D., module select, and block address. This 32 bit bus is protected by 7 bits of ECC.

5.1.1 MEMORY COMMAND FIELD - KMEMCMD$$P(1:0)

The memory data card commands are:

0 - READ MODIFY WRITE
1 - WRITE
2 - READ PUBLIC
3 - READ PRIVATE

The MDC makes no distinction between read private and read public. These different kinds of reads are used by the MCACM and IOMC to maintain cache consistency.

5.1.2 REQUESTOR ID FIELD - KMEMREQIDP(2:0)

This field is for distributed error detection of the memory bus arbitration logic. The definitions of the values of this field are:

0 - MCACM 0
1 - MCACM 1
2 - MCACM 2
3 - MCACM 3
4 - IOMC 0
5 - IOMC 1
6 - IOMC 2
7 - IOMC 3

Not used functionally by the MDC. Used in ECC checking and correction of Address Bus.

5.1.3 MODULE SELECT FIELD - KMODSEL$$P(4:0)
(Ranges from 10 to 1F)

This field selects the memory data cards. The processor can accommodate 16 memory cards. The other 16 values are used to select the I/O memory concentrators for memory mapped I/O operations.

When the module secect is "00" hex through "0F" hex, I/O memory concentrators are selected. When the module select is "10" hex through "1F" hex, memory data cards are selected.

5.1.4 ADDRESS FIELD - KRAMADR$P(21:0)

The absolute binary module address. This allows addressing 4M blocks per memory data card. This is the size of a memory card which is fully populated with 1M bit DRAM chips.

Each MDC requires 19 to 22 bits of address depending on the type and number of RAM chips it contains. A board that is half-populated with 256k-bit chips requires 19 bits, while a board that is fully-populated with 1meg-bit chips requires 22 bits.

5.1.4 ADDRESS FIELD - KRAMADR$P(21:0) (Continued)

	6	5	4	3	2	1	0
00:=	x	x			x		
01:=	x	x					x
02:=	x	x				x	
03:=	x	x	x				
04:=	x			x	x		
05:=		x	x			x	
06:=	x			x			x
07:=	x			x		x	
08:=		x	x				x
09:=		x	x			x	
10:=			x	x	x	x	x
11:=	x		x			x	
12:=				x	x	x	
13:=				x		x	x
14:=			x	x			x
15:=		x		x			x
16:=	x			x		x	
17:=	x			x	x		
18:=	x	x		x			
19:=		x	x	x			
20:=			x		x		x
21:=	x	x	x	x	x		
22:=		x			x	x	
23:=	x				x	x	
24:=		x			x		x
25:=	x				x		x
26:=			x		x	x	
27:=				x	x		x
28:=					x	x	x
29:=		x				x	x
30:=	x					x	x
31:=			x			x	x

FIGURE 5.2 ADDRESS BUS HAMMING CODE FIELD

5.1.4 ADDRESS FIELD - KRAMADR$P(21:0) (Continued)

	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
6543	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
000	*	X	X	D	X	D	D	28	X	D	D	13	D	27	12	D
001	X	D	D	31	D	20	26	D	D	14	M	D	M	D	D	10
010	X	D	D	29	D	24	22	D	D	15	M	D	M	D	D	M
011	D	08	09	D	05	D	D	M	19	D	D	M	D	M	M	D
100	X	D	D	30	D	25	23	D	D	M	16	D	17	D	D	M
101	D	06	07	D	11	D	D	M	04	D	D	M	D	M	M	D
110	D	01	02	D	00	D	D	M	18	D	D	M	D	M	M	D
111	03	D	D	M	D	M	M	D	D	M	M	D	21	D	D	M

* = NO BITS IN ERROR X = CHECK BIT IN ERROR
D = DOUBLE BIT ERROR M = MULTIPLE BIT ERROR

FIGURE 5.3 ADDRESS BUS SYNDROME TO BIT IN ERROR DECODE TABLE

5.2 DATA BUS

The 192 data lines are bi-directional, and composed of 160 bits of actual data plus 32 bits of SEC-DED-S4EC code.

5.2.1 DATA FIELD - XRAMDT(3:0)$P(39:0)

The data bus is divided into four 48 bit groups of 40 bits of data and 8 bits of check bits. There are two reasons for this division: it fits into an array-pair, and it allows full coverage of any single (and total) memory SIP failure. The error code bits 0 and 7 are inverted such that a data word of all zeroes or all ones is detected as invalid. For example, 'proper' words of 0's or 1's are:

All zero's = 810000000000810000000000810000000000810000000000
All one's = 7EFFFFFFFFFF7EFFFFFFFFFF7EFFFFFFFFFF7EFFFFFFFFFF

5.2.2 DATA BUS HAMMING CODE FIELD - XRAMEC(3:0)$$P(7:0)

	7	6	5	4	3	2	1	0
00 :=				X			X	X
01 :=			X				X	X
02 :=		X					X	X
03 :=	X						X	X
04 :=				X		X		X
05 :=			X			X		X
06 :=		X				X		X
07 :=	X					X		X
08 :=				X		X	X	
09 :=			X			X	X	
10 :=		X			X	X		
11 :=	X				X	X		
12 :=				X	X	X		
13 :=			X		X	X		
14 :=		X			X			X
15 :=	X				X			X
16 :=				X	X		X	
17 :=			X		X		X	
18 :=		X			X		X	
19 :=	X				X		X	
20 :=		X		X				X
21 :=		X		X			X	
22 :=		X		X		X		
23 :=		X		X	X			
24 :=	X			X				X
25 :=	X			X			X	
26 :=			X	X		X		
27 :=			X	X	X			
28 :=			X	X				X
29 :=			X	X			X	
30 :=		X	X			X		
31 :=		X	X		X			
32 :=	X		X					X
33 :=	X		X				X	
34 :=	X		X			X		
35 :=	X		X		X			
36 :=	X	X						X
37 :=	X	X					X	
38 :=	X	X				X		
39 :=	X	X			X			

FIGURE 5.4 DATA BUS ERROR CORRECTION CODE

5.2.2 DATA BUS HAMMING CODE FIELD (Continued)

	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
76543		0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0000		*	40	41	D	42	D	D	M	43	D	D	M	D	M	M	D
0001		44	D	D	00	D	04	08	D	D	M	16	D	12	D	D	M
0010		45	D	D	01	D	05	09	D	D	M	17	D	13	D	D	M
0011		D	28	29	D	26	D	D	M	27	D	D	M	D	M	M	D
0100		46	D	D	02	D	06	M	D	D	14	18	D	10	D	D	M
0101		D	20	21	D	22	D	D	M	23	D	D	M	D	M	M	D
0110		D	M	M	D	30	D	D	M	31	D	D	M	D	M	M	D
0111		M	D	D	M	D	M	M	D	D	M	M	D	M	D	D	M
1000		47	D	D	03	D	07	M	D	D	15	19	D	11	D	D	M
1001		D	24	25	D	M	D	D	M	M	D	D	M	D	M	M	D
1010		D	32	33	D	34	D	D	M	35	D	D	M	D	M	M	D
1011		M	D	D	M	D	M	M	D	D	M	M	D	M	D	D	M
1100		D	36	37	D	38	D	D	M	39	D	D	M	D	M	M	D
1101		M	D	D	M	D	M	M	D	D	M	M	D	M	D	D	M
1110		M	D	D	M	D	M	M	D	D	M	M	D	M	D	D	M
1111		D	M	M	D	M	D	D	M	M	D	D	M	D	M	M	D

* = NO BITS IN ERROR (Note: ECC bits 0 and 7 are inverted on the backplane)
nn = SINGLE BIT IN ERROR, where 'nn' is the bad bit
D = MULTIPLE ERRORS (even number)
M = MULTIPLE ERRORS (odd number)
Note that bits 40 thru 47 are the check bits.

FIGURE 5.5 DATA BUS SYNDROME TO BIT-IN-ERROR DECODE TABLE

5.3 STATUS INTERFACE

5.3.1 CABINET SOURCE - XMCABSRC$$$P

Memory cabinet source is asserted true the bus cycle immediately prior to any bus cycle in which the MDC will be driving the backplane.

5.3.2 SOURCE ID (5 bits) - XKMSRCID$P(4:0)

This field contains the physical ID number of the module which is driving the backplane. This ID field shall be driven by the memory data cards on the 5th and 6th clock of a memory read cycle. All modules which source read data must also source the ID field at the same time as the read data.

5.3.3 SOURCE PARITY - XMEMSRCPAR$P

The source parity is calculated such that the parity over XKMSRCID$P(3:0), XNMBRMDC$P(3:0), and XMEMSRCPAR$P is even. The source parity is enabled onto the backplane during the 5th and 6th clocks of all non-inhibited memory cycles.

5.3.4 ERROR FIELD - XMEMERR$P(3:0)

The four error lines are always sourced on the fIfth and sixth clock of every memory cycle. It is up to the requestor to look at them and take some appropriate action.

0000	See Below
0001	NO ERROR
0010	SINGLE BIT DATA ERROR
0100	SINGLE BIT ADDRESS ERROR
0111	MULTIPLE SINGLE DATA ERRORS
1000	undefined
1011	MULTI-BIT DATA ERROR
1101	MULTI-BIT ADDRESS ERROR
1110	INTERNAL MALFUNCTION

FIGURE 5.6 ENCODING OF ERROR FIELD

The error '0000' is a special case and can happen for many reasons:

- It is not the fifth or sixth clock of any cycle. This is a normal condition. Error checking must be qualified to ignore this case.
- The addressed MDC is not present.
- The addressed MDC is present but OFFLINE.
- The addressed MDC is broken. The operation should be retried. If the retry is unsuccessful, then the MP will have to take this MDC offline and reconfigure the memory subsystem before halt-loading the processor.

Errors have precedence according to their binary weight. For example, if a single bit adddess error is detected on a read cycle (and corrected) and then a multi-bit DATA error is found, the DATA error will be reported.

5.4 SPY INTERFACE - XINHDATXFRnP (n=A,B)

The inhibit signal is a pair of duplicate-and-compare wires that signal the MDC's to disable their data bus and status bus drivers on the 5th and 6th clocks of a memory cycle. The inhibit signal is presented on clock T4 (see timing chart in section 6.2).

5.5 CONFIGURATION BUS

The configuration bus consists of 2 fields: the number of memory data cards, and the type of memory data cards. These fields are always driven onto the backplane by all of the online MDC's in the system.

5.5.1 NUMBER OF MEMORY DATA CARDS - XNMBRMDC$P(3:0)

The boards field specifies how many memory data cards are currently plugged into the V500 backplane. This value only includes those cards that are online and not any cards that have been put into an offline condition by the maintenance subsystem.

# Online cards	Kind of interleaving 4-way	2-way	1-way
1			1
2		2	
3		2	1
4	4		
5	4		1
6	4	2	
7	4	2	1
8	2x4		
9	2x4		1
10	2x4	2	
11	2x4	2	1
12	3*4		
13	3*4		1
14	3*4	2	
15	3*4	2	1
16	4*4		

FIGURE 5.7 TYPES OF INTERLEAVING

5.5.2 TYPE OF MEMORY DATA CARD - XCRDTYPE$P(3:0)

The memory subsystem is designed to allow great flexibility in the number and type of memory data cards which may be utilized in any V500 system. This allows different storage capacities and the ability to take advantage of faster and denser DRAM chips as they become available.

This bus is a one out of four code for the type of MDC on the system.

1	-	Half populated	256K DRAM
2	-	Full populated	256K DRAM
4	-	Half populated	1M DRAM
8	-	Full populated	1M DRAM

6 DESCRIPTION OF OPERATION

6.1 POWER-ON-CLEAR CONFIGURATION

There are 5 things that must be done by the maintenance processor at power-on-clear.

1- Clear all 16 YS chains (8 in each cabinet). This will set the ONLINE flip-flop in all the MDC's.

2- Read all 16 YS chains (8 in each cabinet) to determine the number of MDC's that are present.

3- Set the NMBR register in each of the present MDC's to the total number of MDC's in the system. Each MDC drives this field to the MCACM's and IOMC's. The number of MDC's determines the memory mapping.

4- Set the LOGADR registers in each of the present MDC's. These registers determine the logical I.D. of each MDC; each must be unique, with no missing codes. Each MDC compares the XMODSEL$$P(3:0) lines to the LOGADR registers to detect its requests.

5- Set the BUSYSELECT register in each of the present MDC's with the 1-out-of-16 encoded value of the LOGADR registers. This register is used to drive the proper backplane BUSY line. These BUSYSELECT flip-flops reside in the data arrays (MDECC2) - 2 registers in each array for a total of sixteen.

6.2 MEMORY TIMING

V500 MDC Timing - All Cycles Combined
Terms and Definitions
(Reference diagram pages 24-25)

CLOCK - System clock.

MEMBUS - Maintenance controller bus time clock. Can be considered as final qualifier for a MDC request.

REQUEST - The Requestor Command Field. True when MEMBUS is true and MODSEL equals the logical address.

REQSTH - Signal is internal to control array. Denotes the beginning of a request.

REQS0H - Signal is internal to control array. State generator BIT0 which is clocked REQSTH.

REQS1H - Signal is internal to control array. State generator BIT1, which is clocked BIT0.

REQS2H - Signal is internal to control array. State generator BIT2, which is clocked BIT1.

REQS3H - Signal is internal to control array. State generator BIT3, which is clocked BIT2. The state generator is cleared by this bit active on the following clock edge.

REQOFFH - Signals the end of a request cycle. True for REQS3H plus one system clock. Used to de-glitch state decoding at the end of memory cycles.

RASL - Row address strobe out of the control array to the Dynamic RAM matrix.

ROWADREN - From control array to address array. A high to low transition on this signal will switch the RAM address out of the address array from row to column. The address array uses this signal to generate CASL (column address strobe).

WRTENL - Write enable out of the control array. The Dynamic RAMs require a low active write signal.

6.2 MEMORY TIMING (Continued)

HOLDADR - From control array to address array. Holds the address, command, and command parity registers.

HOLD - From control array to all data arrays. Holds data and error registers.

MCABSRC - From control array to backplane. Memory cabinet source is asserted true the bus cycle immediately prior to any bus cycle in which the MDC will be driving the backplane.

INHIBIT - From backplane to control array. Signals the MDC to abort driving the backplane, which normally would have occurred during the next bus cycle.

RAMSEL - From control array to all data arrays. Selects either backplane data (low) or RAM data (high) as the input bus to the ECC logic and data registers.

READ - From control array to all data arrays. Controls the direction of the XRAMDATA I/O bus. On a Read, the bus receives read data from the RAMs. On a Write, the bus is driven with the write data.

SORCEBPL - From control array to all data arrays. Controls the gating of read data onto the backplane. On normal reads, the signal goes low as shown on the timing diagram. It is then latched in the data arrays and extended an extra clock period in order to present the data to the backplane for two clock periods.

BPDATA - The 192 bit bi-direction backplane memory data bus (160 bits data - 32 bits ECC).

6.2 MEMORY TIMING (Continued)

ID & ERR - Two fields driven onto the bus by the control array.

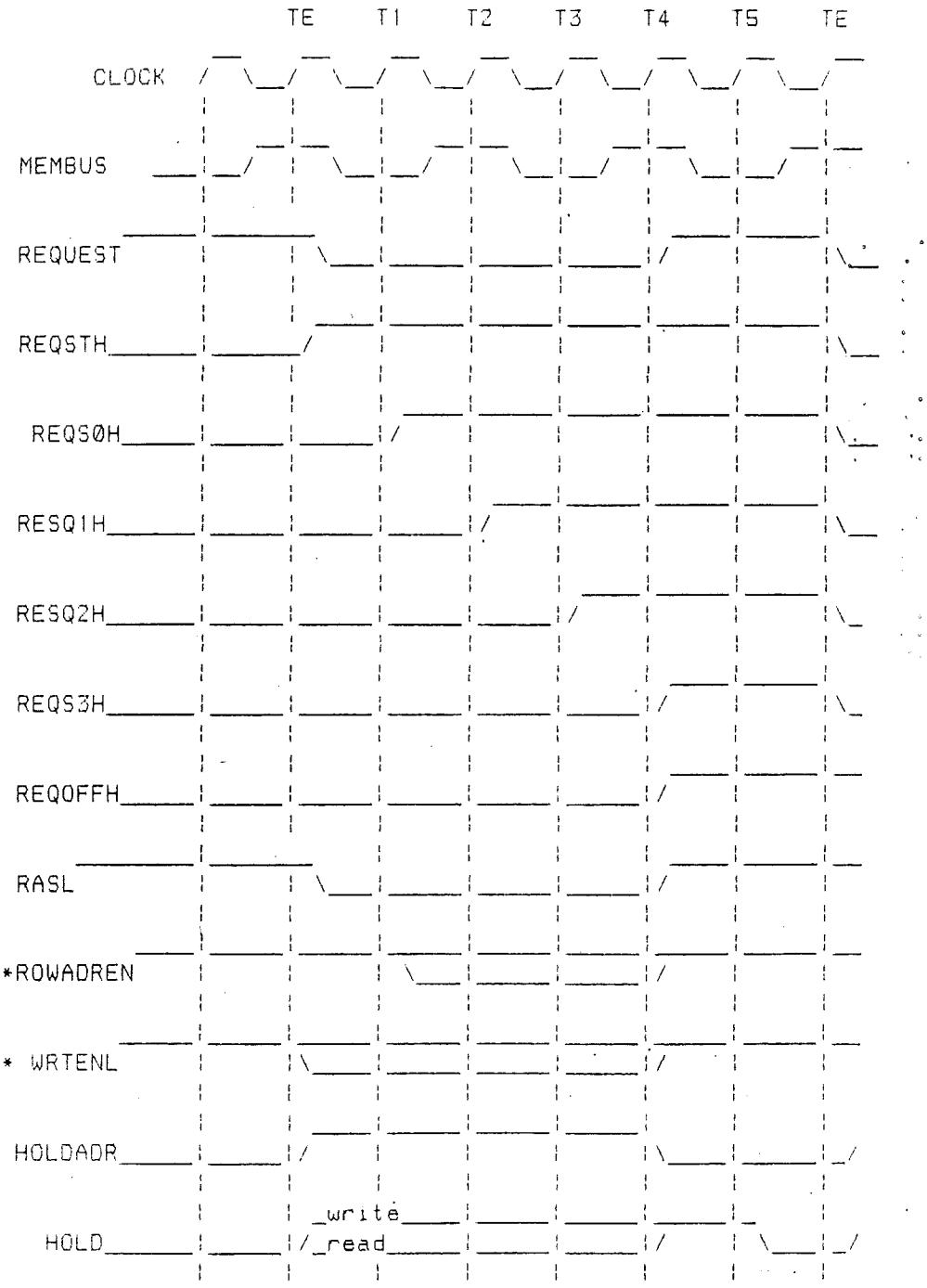
a) KMSRCID(5:0) - The MDC physical address
KMSRCID(2:0) The physical backplane 1 of 8 slot.
KMSRCID(3) The cabinet number 0 or 1.
KMSRCID(4) Always high for MDC's.
KMSRCID(5) Physical address parity. The exclusive Or of KMSRCID(4:0) and the internally latched NMBR(3:0) bus.

b) RAMEMERR(3:0) - The MDC error code:
0000 - Special case, no MDC presently drives these lines.
0001 - No error
0010 - Single bit data error
0100 - Single bit address error
0111 - Multiple single bit data errors
1000 - Undefined
1011 - Multi-bit data errors, uncorrectable
1101 - Multi-bit address errors, uncorrectable
1110 - Internal malfunction

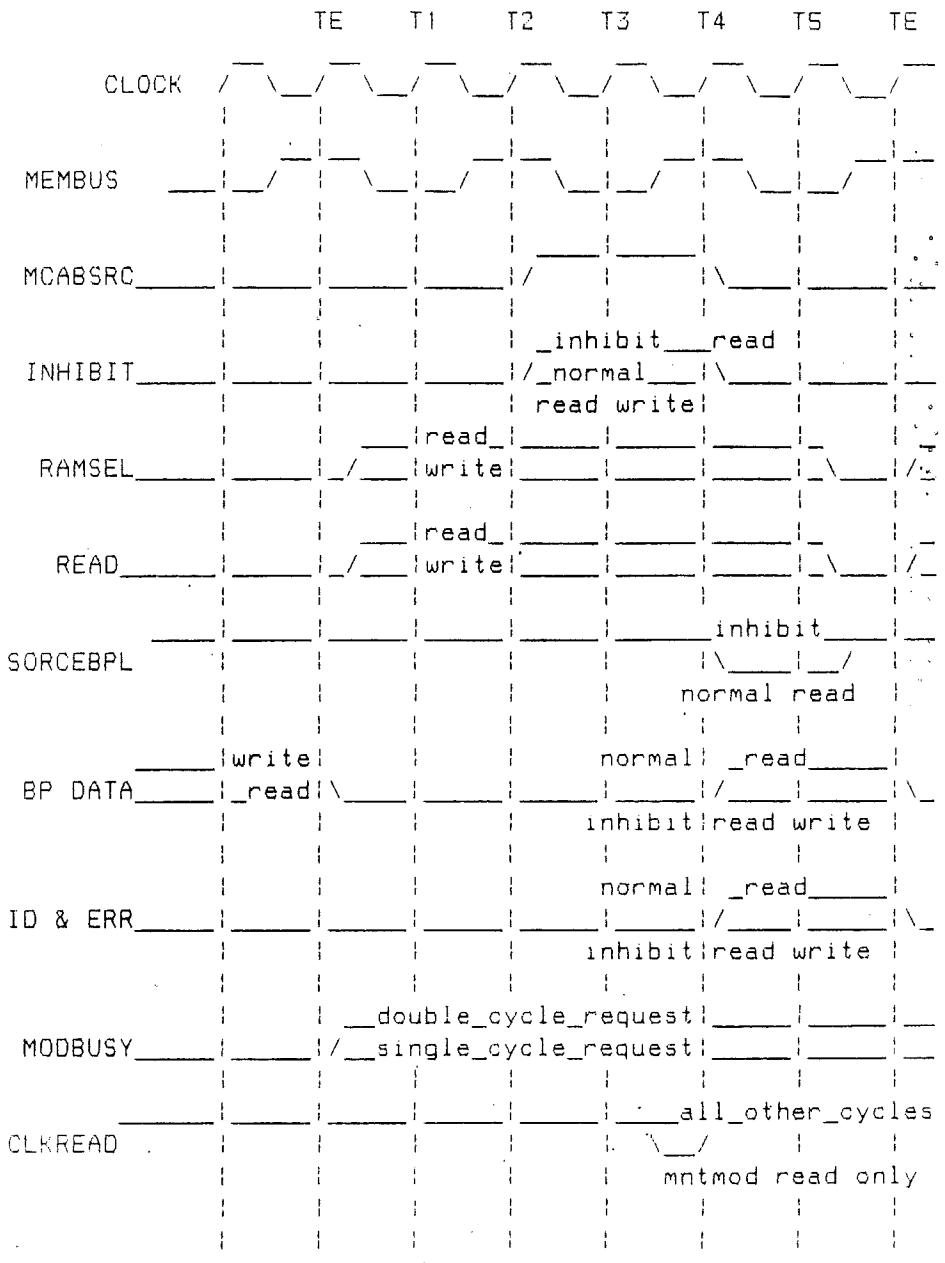
MODBUSY - From control array (BUSY) to all data arrays. Each data array will have two busy flip-flops for a total of 16. The SMC will shift a 1 into the flip-flop corresponding to the MDC's logical address. BUSY will then gate 1 of 16 MODBUSY's onto the backplane indicating a Double cycle is in progress. The signal will stay high until the write portion of the double is executed or it times out - 32 Mclocks. BUSY will also go high for six cycles on receipt of an on-card refresh request.

CLKREADL - From control array to address array. Normally high. This is only pulsed low for one-half clock period if the MDC's clock is stopped during the execution of a read cycle. This pulse is used to clock the read data into the data arrays so that the RAM matrix is free for refresh and the data is not lost.

6.2 MEMORY TIMING (Continued)



6.2 MEMORY TIMING (Continued)



* ROWADREN and WRTENL will stay inactive (high) when there are multi-bit errors on the input address and/or data buses.

7 ERROR DETECTION AND HANDLING

7.1 REPORTING OF PHYSICAL CARD SLOT ON ERRORS

During the fifth and sixth clocks of a memory cycle (see timing diagrams), each MDC will source its physical backplane location (10 thru 1F) onto the memory bus. This can allow the maintenance processor (or SMC) to identify the cause of an error for fault recovery purposes. Note that the physical address is not always the same as a MDC logical address; a MDC may respond to module select (n), but be in card slot (x).

Any error detected by a MDC during a cache sourced read will be lost.

7.2 ERROR CORRECTION

The MDC will correct single-bit-errors whenever possible subject to certain constraints.

7.2.1 ADDRESS BUS

Any single-bit error on the address bus (including module select, requestor I.D., and command) will be corrected when KADRCORREN$P is active (high). This is a backplane pin on the MDC; it will normally be strapped to an always-true signal provided by the MDC. When this strap is not present (i.e., in manufacturing, during initial board level debug), no error correction of the address bus will be done to allow detecting "stuck-at" faults.

7.2.2 DATA BUS

The V500 system level of protection of the data bus is very high. Consider that 4 single-bit-errors could be corrected at both ends of the data bus during both reads and writes. This protection can also be disabled under certain conditions.

7.2.2.1 WRITE

Any single-bit-error on the data bus will be corrected during a write cycle.

7.2.2.2 READ

Single-bit-errors occuring in RAM are normally not corrected. This allows the MCACM or IOMC to calculate the syndrome and correct the data; reporting the bit in error to maintenance.

A bit on the MDC maintenace chain has been provided to force the correction of single-bit-errors in RAM. This bit, CORRECT, will enable continuous processing in the presence of a bad bus line. Single-bit-errors which occur in the MDC while CORRECT is active (high) will not be reported to maintenance.

8 SYSTEM MAINTENANCE INTERFACE

The interface to the system maintenance will allow the internal state of the MDC to be set and examined through the use of a shift chain which contains all of the flip-flops.

8.1 CLOCK-MAINTENANCE ARRAY

The MDC shall use the CLKMNT2 array that is being used by all the other system modules. The MDC will use a subset of the available control signals (CLOCK, CLEAR, SHIFT, and BROKEN). The MEMJMPRH signal will be tied active (high) on the MDC which forces the data chain clocks to be always enabled.

8.2 MAINTENANCE AND DATA SHIFT CHAINS

The "maintenance" chain will provide the primary access to the MDC. The timing and refresh logic shall be contained on the data chain. This is necessary so that the refresh logic will continue to operate even when the internal state of the MDC is being read by the maintenance system.

8.3 SINGLE CLOCK OPERATION

The MDC shall be single-clockable, that is, any command may be started from a MCACM or IOMC, single clocked through to completion, with no difference caused by delays between clocks. No refresh or RAM timing requirements require the clocks to the maintenance chain to be continuous. When clocks are stopped to the maintenance chain, it is required that at least 12 data chain clocks occur before the next maintenance chain clock. This is required because the dynamic RAM chips used in the MDC cannot really do single-clock operations, they must be refreshed on a continuous basis, so the MDC single-clock operation is faked. This "magic" is done with mirrors that take some time to setup.

8.4 SUSPENDED OPERATIONS

The MDC operation shall be suspendable and restartable. That is, at any time the maintenance clocks to the MDC may be stopped, the contents of the maintenance chain saved in the maintenance system, another operation shifted in and performed, the original contents of the maintenance chain restored from the maintenance system, and the original operation be completed. When this is done, the maintenance system is required to follow 7 steps.

1- Stop the clocks to the maintenance chain of the MDC.

2- Save the maintenance chain for future restoration.

3- Shift in the new operation, or allow another operation to be started from a MCACM or IOMC.

4- Complete this new operation. Some operations, like direct reading of the DRAM's on the MDC, may be completed with a single maintenance clock. This is listed as a separate step since there is no MDC design reason that this, interrupting of operations in progress, could not be done repeatedly.

5- Shift in a "phantom" read command and issue a single maintenance chain clock. This read is called "phantom" because it gets executed but never used; it is obliterated in step 6.

6- Restore the saved contents of the maintenance chain from step 2.

7- Continue the maintenance clocks as prior to step 1.

8.5 FORCE REFRESH

Because of the asynchronous nature of refresh, there is a backplane signal between the system maintenance controller and the MDC's that forces refresh. This signal must be asserted at least 12 clocks before starting clocks to MDC's. This holds true even if only 1 clock is to be issued.

8.6 BUS DRIVERS

The MDC shall not disable backplane drivers except when shifting the maintenance chain. Naturally, only those drivers that are required by the current operation will be enabled.

9 BACKPLANE DEFINITION

XRAMDATm-nnP - Memory data buffered, from/to central buffer cards. One of four banks, m = 3:0 each 40 bits wide, nn =39:0.

XRAMECCm-nnP - ECC for bank m from/to central buffer cards. 8 bits, nn = 7:0, m = 3:0.

XRAMEMER$-nP - Error field, from Memory data cards, reports errors detected by the card; encoded, n = 3:0.

XKMSRCID$-nP - Memory bus source; indicates the origin of data on the bus; encoded. n = 4:0

XMEMSRCPAR$P - Memory bus source parity. Even parity over KMSRCID$P(4:0) and XNMBRC$P(3:0). Source on the 5th and 6th clocks of non-inhibited memory cycles.

KMEMCMD$$-nP - Memory bus command from central buffer. n = 1:0

KMODSEL$$-nP - Memory module select from central buffer. n = 4:0

KADDRECC$-nP - ECC for Memory address and memory command fields, from central buffer. n = 6:0

KRAMADR$-nnP - Memory address bus from central buffer. nn = 21:0

X3SHIFTOUT$P - Maintenance serial data out, third subgroup of ORed signals. It goes to SMC.

ASHIFTINF03P - Maintenance serial data in, third fanout. It comes from the SMC.

XMODBRRAM$P - Module broken signal from the memory cards as a group to the SMC.

9 BACKPLANE DEFINITION (Continued)

A3CARDEN$-nP - Module card enable bus from the SMC, third fanout, n = 3:0

ARAMEMODEN$P - Memory module enable from the SMC.

ASHIFTENFO3P - Maintenance shift enable from SMC, third fanout.

ADANGER$FO3P - Maintenance danger signal from SMC which indicates non-standard run condition, third fanout.

ACLEAR$$FO3P - Maintenance clear from the SMC, third fanout.

AnOENRAMCRDP - Zero level clock enable to the memory data card in location A (AxOENRAMCRDP x = A to H) from the SMC.

A$OCKECL$nnP - Zero level clock differential pair N = inverse of the pair, P = positive of the pair. From the SMC, to the memory data card in location A (AxOCKRAMCRDP x = A to H)

Vt - Backplane and card pin connection to Vtt (terminator voltage, -2V).

SPARE - Pin reserved for board repairs because of its limited accessibility.

GND(HARD) - Pin tied to ground, directly through buried ground plane.

GND(SOFT) - Pin tied to ground via a surface level etch to an adjacent buried level ground.

RMNTCARDJ-nP - Maintenance card location jumpers: indicate to the MDC which card location it is in. n = 2:0

RCABNETJMPRP - A backplane jumper indicating which cabinet the MDC resides in.

9 BACKPLANE DEFINITION (Continued)

KMODBUSY-nnP - MDC busy lines. n = 15:0

XNMBRMDC$-nP - The number of MDCs available in the system sourced by all MDCs. n = 3:0

RnJMPRTRUEOP - An ECL logical True sent to the backplane to allow backplane jumpers to be tied True.

KADRCORREN$P - Enable address bus error correction in MDCs.

XINHDATXFRnP - Inhibit data transfer from MDCs, two concurrent signals (i.e. should be equal), n=A,B.

KMEMREQID-nP - Memory requester ID, n = 2:0.

XRCRDTYPE-nP - Memory card type indicator lines, n = 3:0
Each card will drive one and only one.
XRCRDTYP(0)P if 256k half populated card
XRCRDTYP(1)P if 256k full populated card
XRCRDTYP(2)P if 1Meg half populated card
XRCRDTYP(3)P if 1Meg full populated card

XFORCERFRSHP - Forces the MDC to refresh the DRAMS. Used before coming out of maintenance in order to prevent command and refresh overlap.

APOWERUPOK3N - Power supply levels O.K. indicator.

RMEMCABSRC$P - Asserted true the clock cycle immediately prior to any bus cycle in which the MDC will be driving the backplane.

9 BACKPLANE DEFINITION (Continued)

1	2	3	4	5
BACKPLANE	P4	P5	P6	P7
A3	*Vttl*	*Vttl*	*Vttl*	*Vttl*
A4	*Vttl*	*Vttl*	*Vttl*	*Vttl*
A5	*Vttl*	*Vttl*	*Vttl*	*Vttl*
A6	*SPARE*	GND(HARD)	KMODBUSY-15P	*SPARE*
A7	GND(SOFT)	KMODBUSY-14P	GND(SOFT)	GND(SOFT)
A8	KMODBUSY-13P	*Vt*	*SPARE*	*SPARE*
A9	GND(SOFT)	*SPARE*	GND(SOFT)	GND(SOFT)
B0	KMODBUSY-11P	*Vt*	*SPARE*	KMODBUSY-12P
B1	GND(SOFT)	*SPARE*	GND(SOFT)	GND(HARD)
B2	KMODBUSY-09P	*Vt*	*SPARE*	KMODBUSY-10P
B3	GND(SOFT)	*SPARE*	GND(SOFT)	GND(HARD)
B4	*SPARE*	*Vt*	KMODBUSY-08P	*SPARE*
B5	GND(SOFT)	KMODBUSY-07P	GND(SOFT)	GND(HAHD)
B6	*SPARE*	*Vt*	KMODBUSY-06P	*SPARE*
B7	GND(SOFT)	KMODBUSY-05P	GND(SOFT)	GND(HARD)
B8	*SPARE*	*Vt*	KMODBUSY-04P	*SPARE*
B9	GND(SOFT)	KMODBUSY-03P	GND(SOFT)	GND(HARD)
C0	KMODBUSY-02P	*Vt*	*SPARE*	*SPARE*
C1	GND(SOFT)	*SPARE*	GND(SOFT)	GND(HARD)
C2	KMODBUSY-00P	*Vt*	*SPARE*	KMODBUSY-01P
C3	GND(SOFT)	XNMBRMDC$-3P	GND(SOFT)	GND(HARD)
C4	XNMBRMDC$-0P	*Vt*	XNMBRMDC$-1P	XNMBRMDC$-2P
C5	GND(SOFT)	XRAMDAT3-39P	GND(SOFT)	GND(HARD)
C6	XRAMDAT3-37P	*Vt*	XRAMDAT3-38P	*SPARE*
C7	GND(SOFT)	XRAMDAT3-36P	GND(SOFT)	GND(HARD)
C8	XRAMDAT3-33P	*Vt*	XRAMDAT3-34P	XRAMDAT3-35P
C9	GND(SOFT)	XRAMDAT3-32P	GND(SOFT)	GND(HARD)
D0	XRAMDAT3-29P	*Vt*	XRAMDAT3-30P	XRAMDAT3-31P
D1	GND(SOFT)	XRAMDAT3-28P	GND(SOFT)	GND(HARD)
D2	XRAMDAT3-26P	*Vt*	XRAMDAT3-27P	*SPARE*
D3	GND(SOFT)	XRAMDAT3-25P	GND(SOFT)	GND(HARD)
D4	XRAMDAT3-22P	*Vt*	XRAMDAT3-23P	XRAMDAT3-24P
D5	GND(SOFT)	XRAMDAT3-21P	GND(SOFT)	GND(HARD)
D6	XRAMECC3-06P	*Vt*	XRAMECC3-07P	XRAMDAT3-20P
D7	GND(SOFT)	XRAMECC3-05P	GND(SOFT)	GND(HARD)
D8	XRAMECC3-03P	*Vt*	XRAMECC3-04P	*SPARE*
D9	GND(SOFT)	XRAMECC3-02P	GND(SOFT)	GND(HARD)

9 BACKPLANE DEFINITION (Continued)

E0	XRAMDAT3-19P	*Vt*	XRAMECC3-00P	XRAMECC3-01P
E1	GND(SOFT)	XRAMDAT3-18P	GND(SOFT)	GND(HARD)
E2	XRAMDAT3-15P	*Vt*	XRAMDAT3-16P	XRAMDAT3-17P
E3	GND(SOFT)	XRAMDAT3-14P	GND(SOFT)	GND(HARD)
E4	XRAMDAT3-12P	*Vt*	XRAMDAT3-13P	*SPARE*
E5	GND(SOFT)	XRAMDAT3-11P	GND(SOFT)	GND(HARD)
E6	XRAMDAT3-08P	*Vt*	XRAMDAT3-09P	XRAMDAT3-10P
E7	**NO.pin**	**NO.PIN**	**NO.PIN**	**NO.PIN**
E8	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
E9	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
F0	XRAMDAT3-05P	*Vt*	XRAMDAT3-06P	XRAMDAT3-07P
F1	GND(SOFT)	XRAMDAT3-04P	GND(SOFT)	GND(HARD)
F2	XRAMDAT3-02P	*Vt*	XRAMDAT3-03P	*SPARE*
F3	GND(SOFT)	XRAMDAT3-01P	GND(SOFT)	GND(HARD)
F4	XRAMDAT2-38P	*Vt*	XRAMDAT2-39P	XRAMDAT3-00P
F5	GND(SOFT)	XRAMDAT2-37P	GND(SOFT)	GND(HARD)
F6	XRAMDAT2-34P	*Vt*	XRAMDAT2-35P	XRAMDAT2-36P
F7	GND(SOFT)	XRAMDAT2-33P	GND(SOFT)	GND(HARD)
F8	XRAMDAT2-31P	*Vt*	XRAMDAT2-32P	*SPARE*
F9	GND(SOFT)	XRAMDAT2-30P	GND(SOFT)	GND(HARD)
G0	XRAMDAT2-27P	*Vt*	XRAMDAT2-28P	XRAMDAT2-29P
G1	GND(SOFT)	XRAMDAT2-26P	GND(SOFT)	GND(HARD)
G2	XRAMDAT2-23P	*Vt*	XRAMDAT2-24P	XRAMDAT2-25P
G3	GND(SOFT)	XRAMDAT2-22P	GND(SOFT)	GND(HARD)
G4	XRAMDAT2-20P	*Vt*	XRAMDAT2-21P	*SPARE*
G5	GND(SOFT)	XRAMECC2-07P	GND(SOFT)	GND(HARD)
G6	XRAMECC2-04P	*Vt*	XRAMECC2-05P	XRAMECC2-06P
G7	GND(SOFT)	XRAMECC2-03P	GND(SOFT)	GND(HARD)
G8	XRAMECC2-00P	*Vt*	XRAMECC2-01P	XRAMECC2-02P
G9	GND(SOFT)	XRAMDAT2-19P	GND(SOFT)	GND(HARD)
H0	XRAMDAT2-17P	*Vt*	XRAMDAT2-18P	*SPARE*
H1	GND(SOFT)	XRAMDAT2-16P	GND(SOFT)	GND(HARD)
H2	XRAMDAT2-13P	*Vt*	XRAMDAT2-14P	XRAMDAT2-15P
H3	GND(SOFT)	XRAMDAT2-12P	GND(SOFT)	GND(HARD)
H4	XRAMDAT2-09P	*Vt*	XRAMDAT2-10P	XRAMDAT2-11P
H5	GND(SOFT)	XRAMDAT2-08P	GND(SOFT)	GND(HARD)
H6	XRAMDAT2-06P	*Vt*	XRAMDAT2-07P	KADRCORREN$P
H7	GND(SOFT)	XRAMDAT2-05P	GND(SOFT)	GND(HARD)
H8	XRAMDAT2-02P	*Vt*	XRAMDAT2-03P	XRAMDAT2-04P
H9	GND(SOFT)	XRAMDAT2-01P	GND(SOFT)	GND(HARD)

9 BACKPLANE DEFINITION (Continued)

I0	XINHDATXFRAP	*Vt*	XINHDATXFRBP	XRAMDAT2-00P
I1	GND(SOFT)	XRCRDTYPE-3P	GND(SOFT)	GND(HARD)
I2	XRCRDTYPE-1P	*Vt*	XRCRDTYPE-2P	*SPARE*
I3	GND(SOFT)	XRCRDTYPE-0P	GND(SOFT)	GND(HARD)
I4	XRAMEMER$-1P	*Vt*	XRAMEMER$-2P	XRAMEMER$-3P
I5	GND(SOFT)	XRAMEMER$-0P	GND(SOFT)	GND(HARD)
I6	XKMSRCID$-3P	*Vt*	XKMSRCID$-4P	XMEMSRCPAR$P
I7	GND(SOFT)	XKMSRCID$-2P	GND(SOFT)	GND(HARD)
I8	XKMSRCID$-0P	*Vt*	XKMSRCID$-1P	*SPARE*
I9	GND(SOFT)	KMODSEL$$-4P	GND(SOFT)	GND(HARD)
J0	KMODSEL$$-1P	GND(HARD)	KMODSEL$$-2P	KMODSEL$$-3P
J1	*Vecl*	*Vecl*	*Vecl*	*Vecl*
J2	*Vecl*	*Vecl*	*Vecl*	*Vecl*
J3	*Vecl*	*Vecl*	*Vecl*	*Vecl*
J4	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
J5	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
J6	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
J7	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
J8	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
J9	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
K0	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
K1	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
K2	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
K3	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
K4	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
K5	*Vecl*	*Vecl*	*Vecl*	*Vecl*
K6	*Vecl*	*Vecl*	*Vecl*	*Vecl*
K7	*Vecl*	*Vecl*	*Vecl*	*Vecl*
K8	KMEMCMD$$-0P	GND(HARD)	KMEMCMD$$-1P	KMODSEL$$-0P
K9	GND(SOFT)	KADDRECC-6P	GND(SOFT)	GND(HARD)
L0	KADDRECC-4P	*Vt*	KADDRECC-5P	*SPARE*
L1	GND(SOFT)	KADDRECC-3P	GND(SOFT)	GND(HARD)
L2	KADDRECC-0P	*Vt*	KADDRECC-1P	KADDRECC-2P
L3	GND(SOFT)	KRAMADR$-21P	GND(SOFT)	GND(HARD)
L4	KRAMADR$-18P	*Vt*	KRAMADR$-19P	KRAMADR$-20P
L5	GND(SOFT)	KRAMADR$-17P	GND(SOFT)	GND(HARD)
L6	KRAMADR$-15P	*Vt*	KRAMADR$-16P	*SPARE*
L7	GND(SOFT)	KRAMADR$-14P	GND(SOFT)	GND(HARD)
L8	KRAMADR$-11P	*Vt*	KRAMADR$-12P	KRAMADR$-13P
L9	GND(SOFT)	KRAMADR$-10P	GND(SOFT)	GND(HARD)

9 BACKPLANE DEFINITION (Continued)

M0	KRAMADR$-07P	*Vt*	KRAMADR$-08P	KRAMADR$-09P
M1	GND(SOFT)	KRAMADR$-06P	GND(SOFT)	GND(HARD)
M2	KRAMADR$-04P	*Vt*	KRAMADR$-05P	KMEMBUS$T1$P
M3	GND(SOFT)	KRAMADR$-03P	GND(SOFT)	GND(HARD)
M4	KRAMADR$-00P	*Vt*	KRAMADR$-01P	KRAMADR$-02P
M5	GND(SOFT)	KMEMREQID-2P	GND(SOFT)	GND(HARD)
M6	XRAMDAT1-39P	*Vt*	KMEMREQID-0P	KMEMREQID-1P
M7	GND(SOFT)	XRAMDAT1-38P	GND(SOFT)	GND(HARD)
M8	XRAMDAT1-36P	*Vt*	XRAMDAT1-37P	*SPARE*
M9	GND(SOFT)	XRAMDAT1-35P	GND(SOFT)	GND(HARD)
N0	XRAMDAT1-32P	*Vt*	XRAMDAT1-33P	XRAMDAT1-34P
N1	GND(SOFT)	XRAMDAT1-31P	GND(SOFT)	GND(HARD)
N2	XRAMDAT1-28P	*Vt*	XRAMDAT1-29P	XRAMDAT1-30P
N3	GND(SOFT)	XRAMDAT1-27P	GND(SOFT)	GND(HARD)
N4	XRAMDAT1-25P	*Vt*	XRAMDAT1-26P	*SPARE*
N5	GND(SOFT)	XRAMDAT1-24P	GND(SOFT)	GND(HARD)
N6	XRAMDAT1-21P	*Vt*	XRAMDAT1-22P	XRAMDAT1-23P
N7	GND(SOFT)	XRAMDAT1-20P	GND(SOFT)	GND(HARD)
N8	XRAMECC1-05P	*Vt*	XRAMECC1-06P	XRAMECC1-07P
N9	GND(SOFT)	XRAMECC1-04P	GND(SOFT)	GND(HARD)
00	XRAMECC1-02P	*Vt*	XRAMECC1-03P	*SPARE*
01	GND(SOFT)	XRAMECC1-01P	GND(SOFT)	GND(HARD)
02	XRAMDAT1-18P	*Vt*	XRAMDAT1-19P	XRAMECC1-00P
03	GND(SOFT)	XRAMDAT1-17P	GND(SOFT)	GND(HARD)
04	XRAMDAT1-14P	*Vt*	XRAMDAT1-15P	XRAMDAT1-16P
05	GND(SOFT)	XRAMDAT1-13P	GND(SOFT)	GND(HARD)
06	XRAMDAT1-11P	*Vt*	XRAMDAT1-12P	*SPARE*
07	GND(SOFT)	XRAMDAT1-10P	GND(SOFT)	GND(HARD)
08	XRAMDAT1-07P	*Vt*	XRAMDAT1-08P	XRAMDAT1-09P
09	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
P0	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
P1	**NO.PIN**	**NO.PIN**	**NO.PIN**	**NO.PIN**
P2	XRAMDAT1-04P	*Vt*	XRAMDAT1-05P	XRAMDAT1-06P
P3	GND(SOFT)	XRAMDAT1-03P	GND(SOFT)	GND(HARD)
P4	XRAMDAT1-01P	*Vt*	XRAMDAT1-02P	*SPARE*
P5	GND(SOFT)	XRAMDAT1-00P	GND(SOFT)	GND(HARD)
P6	XRAMDAT0-37P	*Vt*	XRAMDAT0-38P	XRAMDAT0-39P
P7	GND(SOFT)	XRAMDAT0-36P	GND(SOFT)	GND(HARD)
P8	XRAMDAT0-33P	*Vt*	XRAMDAT0-34P	XRAMDAT0-35P
P9	GND(SOFT)	XRAMDAT0-32P	GND(SOFT)	GND(HARD)

9 BACKPLANE DEFINITION (Continued)

Q0	XRAMDAT0-30P	*Vt*	XRAMDAT0-31P	*SPARE*
Q1	GND(SOFT)	XRAMDAT0-29P	GND(SOFT)	GND(HARD)
Q2	XRAMDAT0-26P	*Vt*	XRAMDAT0-27P	XRAMDAT0-28P
Q3	GND(SOFT)	XRAMDAT0-25P	GND(SOFT)	GND(HARD)
Q4	XRAMDAT0-22P	*Vt*	XRAMDAT0-23P	XRAMDAT0-24P
Q5	GND(SOFT)	XRAMDAT0-21P	GND(SOFT)	GND(HARD)
Q6	XRAMECC0-07P	*Vt*	XRAMDAT0-20P	*SPARE*
Q7	GND(SOFT)	XRAMECC0-06P	GND(SOFT)	GND(HARD)
Q8	XRAMECC0-03P	*Vt*	XRAMECC0-04P	XRAMECC0-05P
Q9	GND(SOFT)	XRAMECC0-02P	GND(SOFT)	GND(HARD)
R0	XRAMDAT0-19P	*Vt*	XRAMECC0-00P	XRAMECC0-01P
R1	GND(SOFT)	XRAMDAT0-18P	GND(SOFT)	GND(HARD)
R2	XRAMDAT0-16P	*Vt*	XRAMDAT0-17P	*SPARE*
R3	GND(SOFT)	XRAMDAT0-15P	GND(SOFT)	GND(HARD)
R4	XRAMDAT0-12P	*Vt*	XRAMDAT0-13P	XRAMDAT0-14P
R5	GND(SOFT)	XRAMDAT0-11P	GND(SOFT)	GND(HARD)
R6	XRAMDAT0-08P	*Vt*	XRAMDAT0-09P	XRAMDAT0-10P
R7	GND(SOFT)	XRAMDAT0-07P	GND(SOFT)	GND(HARD)
R8	XRAMDAT0-05P	*Vt*	XRAMDAT0-06P	*SPARE*
R9	GND(SOFT)	XRAMDAT0-04P	GND(SOFT)	GND(HARD)
S0	XRAMDAT0-01P	*Vt*	XRAMDAT0-02P	XRAMDAT0-02P
S1	GND(SOFT)	XRAMDAT0-00P	GND(SOFT)	GND(HARD)
S2	RMNTCARDJ-1P	*Vt*	RMNTCARDJ-2P	RCABNETJMPRP
S3	GND(SOFT)	R0JMPRTRUE0P	GND(SOFT)	GND(SOFT)
S4		*Vt*	RMNTCARDJ-0P	*SPARE*
S5	GND(SOFT)	*SPARE*	GND(SOFT)	GND(HARD)
S6	ASHIFTINFO3P	*Vt*	X3SHIFTOUT$P	XMODBRKRAM$P
S7	GND(SOFT)	A3CARDEN$-3P	GND(SOFT)	GND(HARD)
S8	A3CARDEN$-0P	*Vt*	A3CARDEN$-1P	A3CARDEN$-2P
S9	GND(SOFT)	ARAMEMODEN$P	GND(SOFT)	GND(HARD)
T0	ASHIFTENFO3P	*Vt*	ARANGER$FO3P	*SPARE*
T1	GND(SOFT)	ACLEAR$$FO3P	GND(SOFT)	GND(HARD)
T2	AAOCKRAMCRDN	GND(HARD)	AAOCKRAMCRDP	AAOENRAMCRDP
T3	*Vtt*	*Vtt*	*Vtt*	*Vtt*
T4	*Vtt*	*Vtt*	*Vtt*	*Vtt*
T5	*Vtt*	*Vtt*	*Vtt*	*Vtt*

10 GATE ARRAY DEFINITIONS

The MDC uses 4 different MCAII options. These options are: CLKMNT2, MCTRL, MDADD, and eight MDECC2.

CLOCK-MAINTENANCE - CLKMNT2

This is the same clock-maintenance array as used by all of the system modules in the V500.

CONTROL LOGIC - MCTRL

The control logic array provides the timing and control signals for the other arrays on the MDC. The functions of this array include: RAM timing, internal refresh, error and I.D. reporting, and board level fault detection.

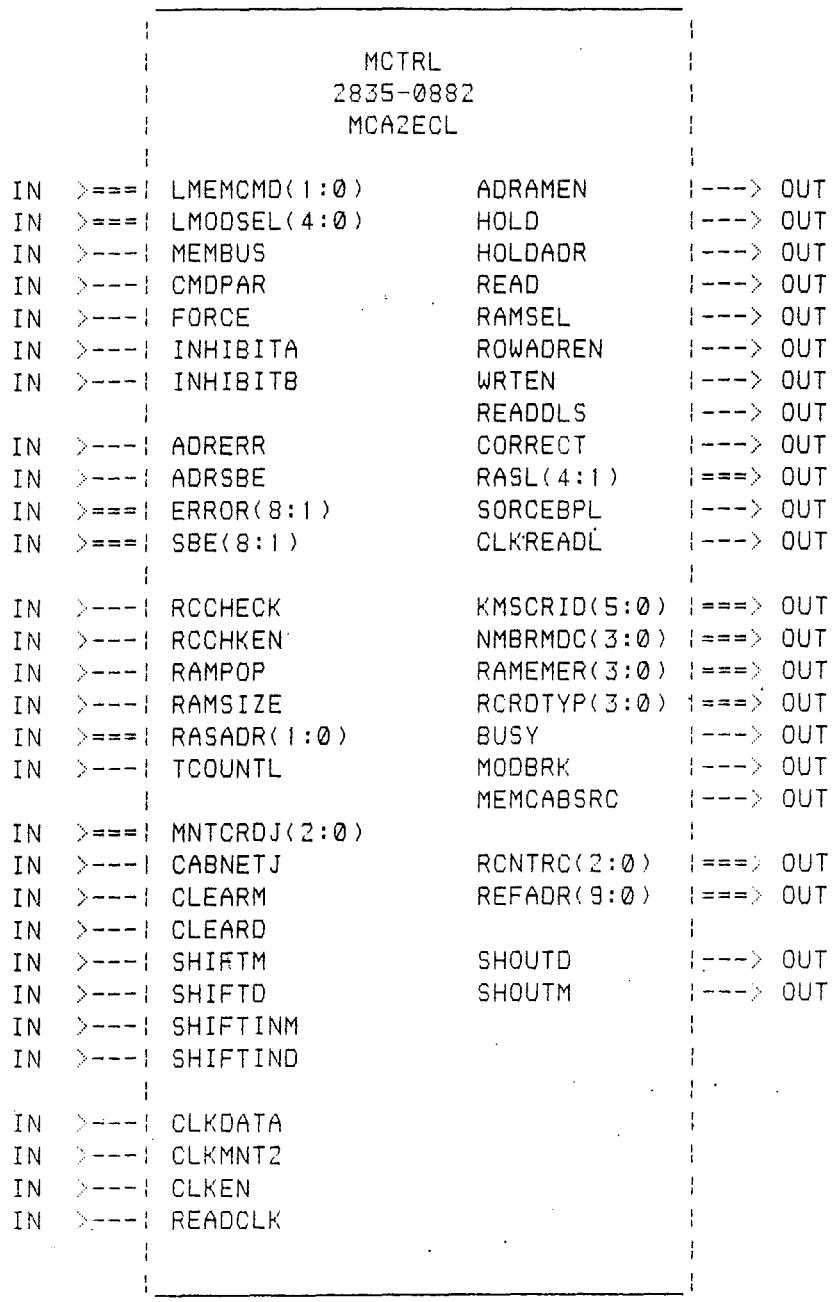
BACKPLANE ADDRESS AND COMMAND RECEIVER - MDADD

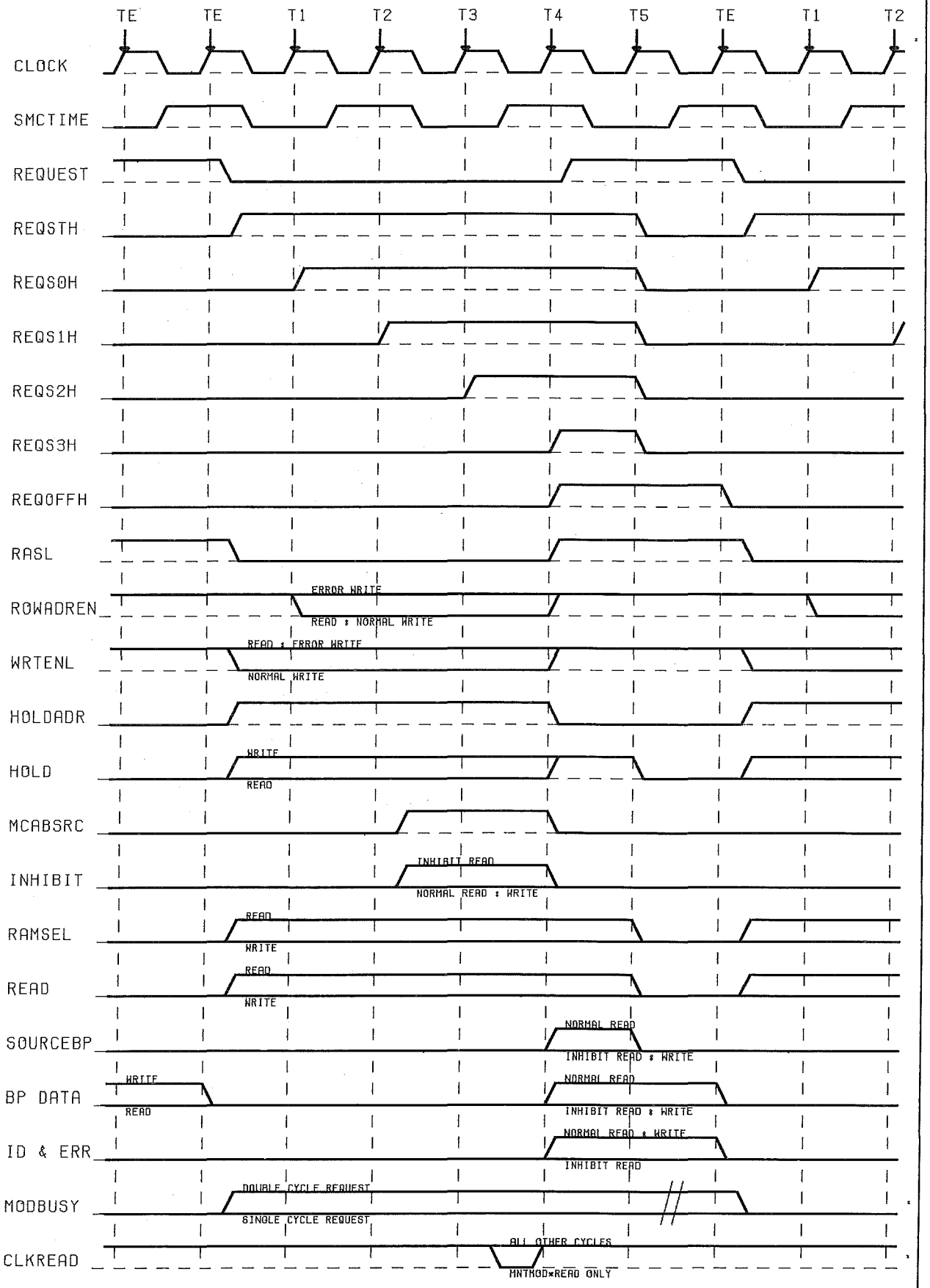
The address, module select, requestor I.D., and command are received and corrected in this array. This array contains the row/column address multiplexor.

BACKPLANE DATA TRANSCEIVER - MDECC2

Two MDECC2 arrays, in a mirror-image pair, are used for each 40 digits of data. They provide the 25 ohm bus drivers, bus receivers, single-bit error correction logic, and RAM data registers for the MDC.

10.1 CONTROL ARRAY - MCTRL





Burroughs Corporation
MICRO COMPONENTS GROUP
MEMORY ENGINEERING
RANCHO BERNARDO PLANT
SAN DIEGO, CALIFORNIA 92127
U.S. AMERICA

TITLE: TIMING - V5 MDC
ALL CYCLES COMBINED

ASSEMBLY NO.	DRAWING NO.	REV.
—	—	**
REL. DOC. - DATE <UNREL> 05/21/85		
— —		
CLASS CODE 2-9520	PAGE 1 OF 1	

10.1.1 MCTRL FUNCTIONAL OVERVIEW

The MCTRL has 4 main functions:

1) Control the memory data card(MDC) - memory backplane interface.

2) Decode and execute the four basic memory commands.

3) Provide for on-card refresh control.

4) Detect and report memory errors and backplane errors as well as MDC hardware errors.

The MCTRL is the heart of the MDC as it controls all signals that drive that backplane as well as all RAM control signals.

The MCTRL first compares the logical address of the MDC to the MODSEL lines for a match. If it is the requested board, the array contains all logic necessary to decode and execute the given command.

In conjunction with a discrete refresh interval counter, the MCTRL array furnishes the refresh address and refresh control signals to the RAM matrix. A FORCED refresh can be requested from the backplane, while the MDC is in maintenance mode, in order to accommodate uninterupted maintenance endeavors.

The control array also has the necessary logic to detect and report errors which happen anywhere on the board. Fault detection approaches 100% through the use of parity, ECC logic on the backplane buses, and various other schemes. The errors are reported in a prioritized manner to the system maintenance processor.

10.1.2 MCTRL INTERFACES

Inputs:

LMEMCMD(1:0)

The latched command bits from the address array. Encoding of the command field is as follows:

00 - READ-MODIFY-WRITE (DOUBLE)
01 - WRITE
10 - READ PUBLIC
11 - READ PRIVATE

The MDC makes no distinction between a read private and read public. It sources read data on the backplane in both cases.

LMODSEL(4:0)

The latched module select lines from the address array. When the mod select is "10" hex through "1F" hex, a memory data card has been selected. When the module select is "00" hex through "0F" hex, I/O memory concentrators are selected. LMODSEL(3:0) is compared with the logical address to determine if it is the requested MDC. The logical address is shifted into the MDC by the maintenance processor during system initialization.

DLMEMBUS

Delayed latched MEMBUS from the address array. The final qualifier for a memory request. LMEMBUS must be high in order to generate a request.

MEMBUS

MEMBUS clock from backplane. Used to deglitch RAS lines and qualify requests.

CMDPAR

Command parity bit over the "command" field sent across the board from the address array. CMDPAR is the XOR of the following: LMEMBUS, MEMCMD(1:0), and MODSEL(4:0).

10.1.2 MCTRL INTERFACES (Continued)

FORCE

Signal originates at the backplane and is used to force a refresh of the memory matrix beginning on the next rising edge of CLKDATA. FORCE may stay High for more than one clock period, however only one refresh will take place. This is used prior to the SMC releasing the MDC from maintenance mode to operating mode to assure that the MDC is not executing an on-board refresh at the time of release A forced refresh will of course increment the refresh address counter and may only be invoked when the MDC is in maintenance mode.

INHIBITA

Originates on the backplane. This signal is used on read commands to inform the MDC that the requested data has been found in the cache. Upon receipt of an INHIBITA, the MDC will prevent data from being driven on the backplane. However, the read data will still be latched into the data registers. This signal must be true prior to the 5th clock of the cycle.

INHIBITB

Originates on the backplane. Even parity on INHIBITA. If there is a discrepency between inhibits, MODBRK will go high indicating a hardware fault.

ADRERR

Address error indicator from the address array. ADRERR going true in the absence of ADRSBE means a multi-bit address error has occurred. Also prohibits WEN/ from going low and feeds the error reporting circuitry See Outputs: RAMEMER(3:0). ADRERR going true in the presence ADRSBE indicates a hardware failure and MODBRK will be activated.

10.1.2 MCTRL INTERFACES (Continued)

AORSBE

Address single bit error flag driven by the address array. Input to the error reporting circuitry. See Outputs: RAMEMER(3:0).

ERROR(8:1)

Data error signals, one from each data array. These lines work in conjunction with the SBE(8:1) field to report on the status of the data bus. Actually, the data arrays work in pairs. For example, the possible combinations for data array pair 1 and 2 are as follows:

ERROR1 = 0	SBE1 = 0	No error.
ERROR2 = 0	SBE2 = 0	
ERROR1 = 0	SBE1 = 1	Single bit error
ERROR2 = 1	SBE2 = 0	in D.A. 1.
ERROR1 = 1	SBE1 = 0	Single bit error
ERROR2 = 0	SBE2 = 1	in D.A. 2.
ERROR1 = 1	SBE1 = 0	Multi-bit error.
ERROR2 = 1	SBE2 = 0	

These errors will be reported by RAMEMER(3:0). All other possible combinations are invalid and will cause the MODBRK hardware fault indicator to go true. The other 3 data array pairs function in a similar fashion.

SBE(8:1)

The single bit error detection lines from the 8 data arrays. See DBE above.

10.1.2 MCTRL INTERFACES (Continued)

RCCHECK

This input comes from the discrete prom which is addressed by the refresh interval counter. The counter is incremented every other system clock, and therefore the address of the prom change every other system clock. The LSB of the prom, which is RCCHECK, was programmed to toggle on every address change. It is then compared to the output of a flip-flop inside the control array which does nothing but toggle also. If the two do not agree, MODBRK will go high as there must be an error in the hardware. This adds a level of fault detection to the counter and prom circuitry.

RCCHKEN

Refresh counter check enable, active high. A low on this input will disable the above mentioned checking scheme. The signal is normally tied high on the MDC.

RAMPOP

A low indicates a board that is half populated with sips. A high implies a fully populated board. This input is jumpered on the MDC accordingly. The status of this line is reported to the system maintenance controller on the CRDTYP(3:0) bus. See Outputs: CRDTYP(3:0).

RAMSIZE

A low indicates a RAMSIZE of 256k-bit. A high implies utilization of the 1M-bit chips. This input is also jumper configurable on the MDC. RAMSIZE is also used to determine the card type.

RASADR(1:0)

RAS address lines from the address array. Decoded to fire a RAS line to 1 of 4 banks of memory.

10.1.2 MCTRL INTERFACES (Continued)

TCOUNTL - active low

A low going pulse indicates that the refresh interval counter has reached the terminal count set by the prom programming and requests that a refresh be done. BUSY will go high as the MDC waits for 6 clocks before doing the refresh. If the card receives a command within these six clocks, it will service the request before executing the refresh. If TCOUNTL goes active during an operation, the command will be serviced as usual with the refresh occurring afterwards. The refresh address counter will be incremented.

MNTCRDJ(2:0)

The maintenance card jumpers or physical address. These lines originate at the backplane and go to the CLKMNT2 array for maintenance identification and to the control array so that it may report the physical i.d. to the requestor. See Outputs: KMSCRID(5:0).

CABNETJ

The cabinet jumper. True signifies cabinet 1 and a logic low corresponds to cabinet 0 of a possible two cabinet configuration.

CLEARM

Maintenance chain reset signal. This is an asynchronous clear provided CLKMNT2 is high.

CLEARD

Data chain reset signal. This is an asynchronous clear provided CLKDATA is high.

SHIFTM

Maintenance chain serial shift enable sign from the clock and maintenance chip.

10.1.2 MCTRL INTERFACES (Continued)

SHIFTD

Data chain serial shift enable signal from the clock and maintenance chip.

SHIFTINM

The maintenance chain serial shift input.

SHIFTIND

The data chain serial shift input.

CLKDATA

The normal system data chain clock out of the clock and maintenance chip.

CLKMNT2

The normal system maintenance chain clock out of the clock and maintenance chip.

CLKEN

This input is driven by ENRAMCRDP on the backplane. The signal is used by the CLKMNT2 array to enable the clocks to the data and maintenance chains for both running and maintenance modes. A low will notify the control array to place the card in maintenance mode on the next rising edge of CLKDATA.

READCLK

Data clock from the CLKMNT2 array. Signal is used to generate CLKREADL so that the rising edge will occur at the same time as a normal clock, i.e. when not in maintenance mode.

10.1.2 MCTRL INTERFACES (Continued)

Outputs:

ADRAMEN

Goes low on a refresh to signal the address array to gate off the row address and allow the refresh address from the control array to OR through the address array.

HOLD

When true, places all registers in the data arrays into the hold state.

HOLDADR

Informs the address array that a request has been received and to therefore hold the address registers.

READ

Controls the bi-directional interface between the data arrays and the ram matrix. A true allows ram data to enter the data arrays on a read command.

RAMSEL

Controls the input to the ECC circuitry and registers in the data arrays. A high selects ram data whereas a low selects the backplane data.

ROWADREN

To the address array. Is high at the start of a cycle, which sends out the row address, then goes low to ship out the column address. Also fires PRECASN on a high-to-low transition.

WRTEN - active low

The ram matrix write enable signal.

10.1.2 MCTRL INTERFACES (Continued)

READDLS

Enables the TTL-ECL level shifters to drive the bi-directional ram data bus on read commands.

CORRECT

A true permits the data arrays to correct single bit errors before the data word is latched. This signal is controlled by a flip-flop on the maintenance chain whose state is determined and shifted in by the system maintenance controller.

RASL(4:0) - active low

The row address strobes to the ram matrix. On refresh operations all 4 are active at once. Normally the RASADR(1:0) lines are decoded to 1 of 4.

SORCEBPL

To the data arrays. Controls the gating of read data onto the backplane. Goes true on the fifth clock of a read cycle. It is the latched in the data arrays and extended another clock period in order to present the data to the backplane for two clock periods.

CLKREADL

Normally high. This signal is used to register the read data if the MDC was placed in maintenance mode (Mclks stopped) before the operation was allowed to complete. This signal will provide a low-to-high transition on what would normally be the fifth system clock of the read cycle.

10.1.2 MCTRL INTERFACES (Continued)

KMSCRID(5:0)

During the fifth and sixth clocks of a memory cycle, the MDC will source its physical backplane location onto the memory bus. This can allow the system maintenance processor to determine the cause of an error. KMSCRID(2:0) are the card jumpers from the backplane. KMSRCID(3) is the cabinet jumper. KMSRCID(4) is always high for a MDC. KMSCRID(5) is the XOR of KMSRCID(4:0) and NMBR(3:0). See below for an explanation of NMBR(3:0). The physical address should not be confused with the logical address. It is quite possible for an MDC to respond to module select (n) but be in card slot (x).

NMBR(3:0)

All MDC's drive the backplane with the number of memory cards in the system. Each memory card has a flip-flop (ONLINE) on its maintenance chain which, upon clearing the chain, will be shifted out as a one. At configuration time, the SMC will count the number of ONLINE's it shifted out and determine the number of cards in the system. It will then shift in this number to all MDC's.

RAMEMER(3:0)

On the fifth and sixth clock of a memory cycle, the MDC will report its error status on the memory bus. The error code is as follows:

0000 - special case, no MDC presently drives these lines
0001 - no error
0010 - single bit data error
0100 - single bit address error
0111 - multiple single bit data errors
1000 - undefined
1011 - multi-bit data errors, uncorrectable
1101 - multi-bit address errors, uncorrectable
1110 - internal malfunction

Errors have precedence according to their binary weight.

10.1.2 MCTRL INTERFACES (Continued)

RCRDTYP(3:0)

One of four decode of RAMPOP and RAMSIZE:

RCRDTYP0 256K RAMS AND HALF POPULATED
RCRDTYP1 256K RAMS AND FULLY POPULATED
RCRDTYP2 1M RAMS AND HALF POPULATED
RCRDTYP3 1M AND FULLY POPULATED

These lines are held low while in maintenance mode.

BUSY

To the data arrays in order to gate one of sixteen BUSIES on to the backplane. BUSY will go true on DOUBLE commands and on a refresh request. In the case of a DOUBLE, BUSY will go true at the start of the read and stay high until the write back, or any other instruction has been requested. Meanwhile, It is understood that the requestor of the DOUBLE will ignore the BUSY on the write portion. It should be noted that if the requestor of the Double does not complete the write-back within 32 Mclocks, the operation will timeout and Busy will be reset to a low. If an on-board request for a refresh requires servicing, BUSY will go true for six clocks to flag the requestors of an impending refresh operation. If a command is received during this period, it must be executed and the refresh will have to wait until after completion of the operation.

MODBRK

Module broken signal tells the CLKMNT2 array that there exists a hardware error and if appropriate, stop the clocks at once. MODB will go true on the following occurrences:

a) Data ECC logic failure
b) Address ECC logic failure
c) An inhibit parity error
d) A RAMCYCLE overlapping a REFCYCLE
e) A refresh counter error
f) A refresh address error
g) A refresh request error
h) A command parity (CPARERR) error

10.1.2 MCTRL INTERFACES (Continued)

MEMCABSRC

Each MDC drives memory cabinet source on the backplane. The signal is asserted true the bus cycle immediately prior to any bus cycle in which the MDC will be driving the backplane.

RCNTRC(2:0)

These lines control the discrete refresh interval counters as follows:

000 - Parallel load
011 - Count up
101 - Clear
110 - Shift right
111 - Hold

The counters load all zeros after a refresh has been completed, i.e. start the interval over. The counters only count every other clock. They are cleared on a CLEARD, and are shiftable.

REFADR(9:0)

The refresh address sent to the address array. The refresh address counter is incremented by TCOUNTL or FORCE, immediately prior to gating out REFADR(9:0).

SHOUTD

The data chain serial shift output.

SHOUTM

The maintenance chain serial shift output.

10.1.3 MCTRL FAULT DETECTION

The "command" lines from the address array to the control array are parity protected. The INHIBIT line from the backplane is also parity protected.

Faults in the refresh address counter are detected by parity prediction. The refresh interval counter has a fault detection scheme as described under Inputs: RCCHECK. The on-board refresh request signal, TCOUNTL, is checked for proper operation by making sure it goes away two clocks later.

The physical address and NMBR(3:0) are XOR-ed and reported as KMSCRCID(5).

Of course the control array is responsible for reporting data and address errors. Their fault detection techniques are covered in their respective specifications and will not be repeated here.

10.1.4 MCTRL BOARD TEST FACILITIES

Besides the serial shift chains, the control array has a CORRECT flip-flop which when loaded with a 1 will allow the data arrays to correct single bit READ data errors from the ram matrix.

10.1.5 MCTRL SHIFT CHAIN DEFINITION

THE TOTAL DATA CHAIN (through the MCTRL and the discrete refresh interval counter) IS PRESENTED HERE:

FF#	NAME	DESCRIPTION
---	ASHIFTINFO3P	SHIFT IN TO BOARD (BP PIN S4P4)
---	----------	ENTER CONTROL ARRAY
001	DREQS0H	DATA CHAIN REQUEST STATE 0
002	DREQS1H	DATA CHAIN REQUEST STATE 1
003	DREQS2H	DATA CHAIN REQUEST STATE 2
004	DREQS3H	DATA CHAIN REQUEST STATE 3
005	DREQS4H	DATA CHAIN REQUEST STATE 4
006	MNTMODH	MAINTENANCE MODE
007	MODBRKH	MODULE FAIL
008	ONLINEH	ONLINE CONTROL FF
009	REFBUSYH	REFRESH BUSY
010	RWAIT1H	REFRESH WAIT STATE 1
011	RWAIT2H	REFRESH WAIT STATE 2
012	RWAIT3H	REFRESH WAIT STATE 3
013	RWAIT4H	REFRESH WAIT STATE 4
014	RWAIT5H	REFRESH WAIT STATE 5
015	REFCYC1H	REFRESH CYCLE STATE 1
016	REFCYC2H	REFRESH CYCLE STATE 2
017	REFCYC3H	REFRESH CYCLE STATE 3
018	REFCYC4H	REFRESH CYCLE STATE 4
019	REFRASEH	REFRESH RAS CYCLE END
020	RCNTR0H	REFRESH ADDRESS COUNTER BIT 0
021	RCNTR1H	REFRESH ADDRESS COUNTER BIT 1
022	RCNTR2H	REFRESH ADDRESS COUNTER BIT 2
023	RCNTR3H	REFRESH ADDRESS COUNTER BIT 3
024	RCNTR4H	REFRESH ADDRESS COUNTER BIT 4
025	RCNTR5H	REFRESH ADDRESS COUNTER BIT 5
026	RCNTR6H	REFRESH ADDRESS COUNTER BIT 6
027	RCNTR7H	REFRESH ADDRESS COUNTER BIT 7
028	RCNTR8H	REFRESH ADDRESS COUNTER BIT 8
029	RCNTR9H	REFRESH ADDRESS COUNTER BIT 9
030	RCNTRPH	REFRESH ADDRESS COUNTER PARITY BIT
031	RICCHKBH	REFRESH INTERVAL COUNTER CHECK BIT
032	RICHOLDL	REFRESH INTERVAL COUNTER HOLD COUNT
033	FRCSTRBH	FORCE REFRESH STROBE
034	FRCREFH	CLOCKED FORCE REFRESH SIGNAL
---	----------	EXIT CONTROL ARRAY

10.1.5 MCTRL SHIFT CHAIN DEFINITION (Continued)

DATA CHAIN DEFINITION (CONTINUED)

---	----------	ENTER DISCRETE REFRESH INTERVAL COUNTER
035	RICNTR0H	REFRESH INTERVAL COUNTER BIT 0
036	RICNTR1H	REFRESH INTERVAL COUNTER BIT 1
037	RICNTR2H	REFRESH INTERVAL COUNTER BIT 2
038	RICNTR3H	REFRESH INTERVAL COUNTER BIT 3
039	RICNTR4H	REFRESH INTERVAL COUNTER BIT 4
040	RICNTR5H	REFRESH INTERVAL COUNTER BIT 5
041	RICNTR6H	REFRESH INTERVAL COUNTER BIT 6
042	RICNTR7H	REFRESH INTERVAL COUNTER BIT 7
---	----------	EXIT DISCRETE REFRESH INTERNAL COUNTER
---	----------	ENTER CLOCK/MAINT ARRAY
---	----------	EXIT CLOCK/MAINT ARRAY
---	X3SHIFTOUT$P	SHIFT OUT FROM BOARD (BP PIN S6P6)

THE MAINTENANCE CHAIN:

FF#	NAME	DESCRIPTION
---	----------	----------
---	ASHIFTINFO3P	SHIFT IN TO BOARD (BP PIN S4P4)
---	----------	ENTER CONTROL ARRAY
001	MREQS0H	MAINTAINENCE CHAIN REQUEST STATE 0
002	MREQS1H	MAINTAINENCE CHAIN REQUEST STATE 1
003	MREQS2H	MAINTAINENCE CHAIN REQUEST STATE 2
004	MREQS3H	MAINTAINENCE CHAIN REQUEST STATE 3
005	MREQSOFFH	MAINTAINENCE CHAIN REQUEST OFF
006	SOURCEH	SOURCE DATA BUS
007	REPORTH	REPORT NOT MDC ID AND ERROR CODE
008	ECORRENH	ERRROR CORRECT ENABLE
009	LNMBR0H	LATCHED MDC COUNT NUMBER BIT 0
010	LNMBR1H	LATCHED MDC COUNT NUMBER BIT 1
011	LNMBR2H	LATCHED MDC COUNT NUMBER BIT 2
012	LNMBR3H	LATCHED MDC COUNT NUMBER BIT 3
013	LLOGAD0H	LATCHED CARD LOGICAL ADDRESS BIT 0
014	LLOGAD1H	LATCHED CARD LOGICAL ADDRESS BIT 1
015	LLOGAD2H	LATCHED CARD LOGICAL ADDRESS BIT 2
016	LLOGAD3H	LATCHED CARD LOGICAL ADDRESS BIT 3
017	DBLBSYH	DOUBLE CYCLE BUSY
018	MONLINEH	MONITOR OF DATA CHAIN ONLINE FF
019	BUSYCTR0H	BUSY TIMEOUT COUNTER BIT 0

10.1.5 MCTRL SHIFT CHAIN DEFINITION (Continued)

MAINTENANCE CHAIN DEFINITION (CONTINUED)

FF#	NAME	DESCRIPTION
020	BUSYCTR1H	BUSY TIMEOUT COUNTER BIT1
021	BUSYCTR2H	BUSY TIMEOUT COUNTER BIT2
022	BUSYCTR3H	BUSY TIMEOUT COUNTER BIT3
023	BUSYCTR4H	BUSY TIMEOUT COUNTER BIT4
024	BUSYCTR5H	BUSY TIMEOUT COUNTER BIT5
025	BUSYCTRPH	BUSY TIMEOUT COUNTER PARITY BIT
026	LMEMBUSH	LATCHED MEMBUS
---	----------	EXIT CONTROL ARRAY

10.1.6 MCTRL PIN-OUT

CONTROL ARRAY PIN NAME	PIN #	I/O	DESCPITION OF PIN SIGNAL
ADRAMEN	A2	O5Ø	RAM ADDRESS ENABLE TO AA.
ADRERR	N12	I-R	ADDRESS ERROR FROM AA.
ADRS8E	N11	I-R	ADDRESS SINGLE BIT ERROR FROM AA.
BUSY	E3	O25	BUSY TO ALL DA.
CABNETJ	N1	I	CABINET LOCATION JUMPER BIT (BP-S2P7).
CLEARD	B1	I	DATA CLEAR FROM CLKA.
CLEARM	A3	I-R	MAINT CLEAR FROM CLKA.
CLKDATA	B3	I-R	D-CLOCK FOR CTRLA D-CHAIN FROM CLKA.
CLKEN	P1	I	CLOCK ENABLE FROM BP - AAØENRAMCRDP.
CLKMNT	A4	I-R	M-CLOCK FOR CTRLA M-CHAIN FROM CLKA.
CLKREADL	J14	O5Ø	M-MODE CLK PULSE TO AA (CLK READ DATA).
CMDPAR	P2	I-R	COMMAND PARITY BIT FROM AA.
CORRECT	D13	O25	CORRECT TO ALL DA.
ERROR1	B13	I-R	ERROR SIGNAL FROM DA1.
ERROR2	B6	I-R	" " " DA2.
ERROR3	B12	I-R	" " " DA3.
ERROR4	B7	I-R	" " " DA4.
ERROR5	B11	I-R	" " " DA5.
ERROR6	B8	I-R	" " " DA6.
ERROR7	B1Ø	I-R	" " " DA7.
ERROR8	B9	I-R	" " " DA8.
FORCE	R4	I-A	FORCE REFRESH FROM BP.
HOLD	D14	O25	HOLD SIGNAL TO ALL DA.
HOLDADR	R2	O5Ø	HOLD SIGNAL TO AA.
INHDATXA	D2	I-A	INHIBIT DRIVING D-BUS SIGNAL BIT (BP-I0P4).
INHDATXB	E2	I-A	INHIBIT DRIVING D-BUS EVEN PARITY BIT (BP-I0P6).
KMSRCID0	E1	O25	CARD IDENTIFICATION BUS BIT 0 TO BP.
KMSRCID1	L3	O25	" " " 1 TO BP.
KMSRCID2	F2	O25	" " " 2 TO BP.
KMSRCID3	L2	O25	" " " 3 TO BP.
KMSRCID4	F1	O25	" " " 1(MEM) TO BP.
KMSRCID5	M3	O25	" " " PARITY TØ BP.
LMEMCMD0	L14	I-R	LATCHED MEM COMMAND BIT Ø FROM AA.
LMEMCMD1	M15	I-R	LATCHED MEM COMMAND BIT 1 FROM AA.
LMODSEL0	M14	I-R	MDC SELECT (LOG ADR) BIT Ø FROM AA.
LMODSEL1	R3	I-R	" " (" ") " 1 " ".
LMODSEL2	P13	I-R	" " (" ") " 2 " ".
LMODSEL3	P15	I-R	" " (" ") " 3 " ".

10.1.6 MCTRL PIN-OUT (Continued)

CONTROL ARRAY PIN NAME	PIN #	I/O	DESCPITION OF PIN SIGNAL
LMODSEL4	R13	I-R	MDC SELECT (LOG ADR) BIT 0 FROM AA.
DLMEMBUS	N13	I-R	DELAYED LATCHED MEMBUS FROM AA.
MCABSRC	C2	025	MEMORY CABINET SOURCE TO BP PIN R2P7.
MEMBUS	B12	I	MEMBUS CLOCK OFF BACKPLANE.
MODBRK	N3	050	MODULE BROKEN SIGNAL TO CLKA.
MNTCRDJ0	P4	I	MAINT CARD LOC JMPR BIT 0 FROM BP.
MNTCRDJ1	N4	I	" " " " " 1 " " .
MNTCRDJ2	N5	I	" " " " " 2 " " .
NMBRMDC0	K2	025	COUNT OF ON MDCS, BIT 0 TO BP.
NMBRMDC1	M2	025	" " " " " 1 " ".
NMBRMDC2	K1	025	COUNT OF ON MDCS, BIT 2 TO BP.
NMBRMDC3	N2	025	" " " " " 3 " ".
RAMEMER0	F15	025	RAM MEMORY ERROR BUS BIT 0 TO BP.
RAMEMER1	F14	025	" " " " " 1 " ".
RAMEMER2	K15	025	" " " " " 2 " ".
RAMEMER3	K14	025	" " " " " 3 " ".
RAMPOP	C5	I	FULL/HALF POP BOARD FROM BD JUMPER.
RAMSEL	E13	025	RAM SELECT TO ALL DA.
RAMSIZE	J2	I	1M/256K RAM DIE FROM BD JUMPER.
RASL1	B14	050	RAS THRU LS&B TO FIRST BLOCK OF RAM.
RASL2	D15	050	RAS THRU LS&B TO SECOND BLOCK OF RAM.
RASL3	G15	050	RAS THRU LS&B TO THIRD BLOCK OF RAM.
RASL4	H14	050	RAS THRU LS&B TO FOURTH BLOCK OF RAM.
RASADR0	C12	I-R	RAS ADDRESS BIT 0 FROM AA.
RASADR1	C11	I-R	RAS ADDRESS BIT 1 FROM AA.
RCCHECK	C15	I-R	COUNTING CHECK BIT FROM RC.
RCCHKEN	G14	I	RC CHECK ENABLE FROM AA - JMPRTRU2.
RCNTRC0	E15	050	REFRESH CNTR CTRL BIT 0 TO RC.
RCNTRC1	R14	050	" " " " 1 " ".
RCNTRC2	P14	050	" " " " 2 " ".
RCRDTYP0	L15	025	256K RAMS & HALF POP BOARD TO BP.
RCRDTYP1	L13	025	256K RAMS & FULL POP BOARD TO BP.
RCRDTYP2	M13	025	1M RAMS & HALF POP BOARD TO BP.
RCRDTYP3	N14	025	1M RAMS & FULL POP BOARD TO BP.
READ	C14	025	READ RAM DATA TO ALL DA.
READCLK	D3	025	DATA CLOCK FROM CLKMNT2.
READDLS	C13	050	READ TO RDATA-TE.
REFADR0	M1	050	REFRESH ROW ADR BIT 0 TO AA.
REFADR1	L1	050	" " " " 1 " ".
REFADR2	J1	050	" " " " 2 " ".

10.1.6 MCTRL PIN-OUT (Continued)

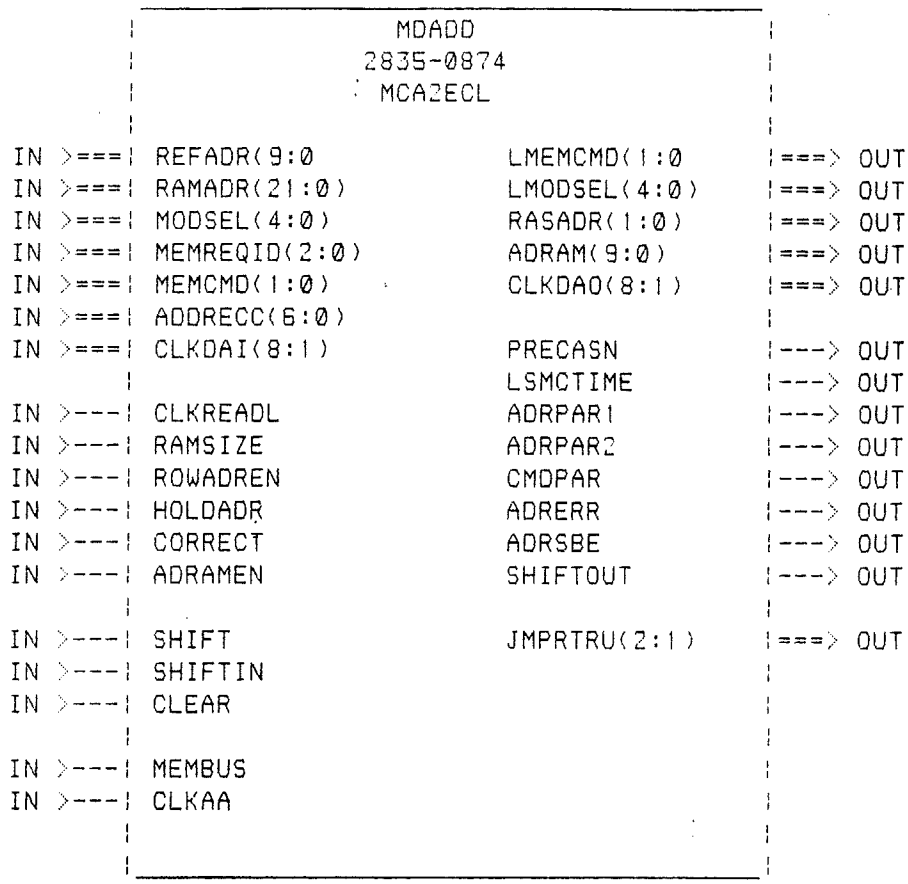
CONTROL ARRAY PIN NAME	PIN #	I/O	DESCPITION OF PIN SIGNAL
REFADR3	H2	O50	REFRESH ROW ADR BIT 3 TO AA.
REFADR4	H1	O50	" " " " 4 " ".
REFADR5	G2	O50	" " " " 5 " ".
REFADR6	G1	O50	" " " " 6 " ".
REFADR7	D1	O50	" " " " 7 " ".
REFADR8	C1	O50	" " " " 8 " ".
REFADR9	B2	O50	" " " " 9 " ".
ROWADEN	J15	O50	ROW ADDRESS ENABLE TO AA.
SBE1	A13	I-R	SINGLE BIT ERROR FROM DA1.
SBE2	A6	I-R	" " " " DA2.
SBE3	A12	I-R	" " " " DA3.
SBE4	A7	I-R	" " " " DA4.
SBE5	A11	I-R	" " " " DA5.
SBE6	A8	I-R	" " " " DA6.
SBE7	A10	I-R	" " " " DA7.
SBE8	A9	I-R	" " " " DA8.
SHIFTD	B4	I	DATA SHIFT FROM CLKA.
SHIFTM	C4	I-R	MAINT SHIFT FROM CLKA.
SHIFTIND	A5	I	DATA SHIFT CHAIN INPUT FROM BP SHIFTIN.
SHIFTINM	B5	I-A	MAINT SHIFT CHAIN INPUT FROM BP SHIFTIN.
SHOUTD	H15	O50	DATA SHIFTOUT TO AA SHIFTIN.
SHOUTM	A14	O50	MAINT SHIFTOUT TO RC.
SORCEBPL	E14	O25	DRIVE BACKPLANE DATA BUS TO ALL DATA ARRAYS
TCOUNTL	B15	I-R	TERMINAL COUNT LOW FROM RC.
WRTENL	N15	O50	WRITE SIGNAL THRU LS&B TO ALL RAM SIPS.

10.1.6 MCTRL PIN-OUT (Continued)

DEFINITION OF ABBREVIATIONS:

AA	= ADDRESS ARRAY.
CLKA	= CLOCK ARRAY.
CTRLA	= CONTROL ARRAY.
BD	= CIRCUIT BOARD.
BP	= BACKPLANE.
DA#	= DATA ARRAY, WHERE # IS THE USAGE NUMBER (1 TO 8).
I	= INPUT, NO TERMINATION.
I-A	= INPUT, WITH ACTIVE TERMINATOR LOCATED NEAR CTRLA.
I-B	= INPUT, WITH ACTIVE TERMINATOR LOCATED NEAR BP.
I-R	= INPUT, WITH 50 OHM RESISTIVE TERMINATORS.
LS&B	= ECL-TTL LEVEL SHIFTERS & TTL BUFFERS.
RC	= DISCRETE REFRESH COUNTER.
RDATA###	= BIDIRECTIONAL DATA BUS BETWEEN THE DATA ARRAYS AND THE ECL/TTL LEVEL SHIFTERS. ### IS THE BIT NUMBER.
RDATA-TE	= TTL-ECL LEVEL SHIFTERS ON THE RDATA### BUS.
XDATA###	= BIDIRECTIONAL DATA BUS BETWEEN THE DATA ARRAYS AND THE MDC'S BACKPLANE CONNECTOR. ### IS THE BIT NUMBER.

10.2 ADDRESS ARRAY - MDADD



10.2.1 MDADD FUNCTIONAL OVERVIEW

The MDADD array has 4 primary functions:

1) Register the address and command field off the memory backplane.

2) Provide row, column, and refresh address muxing to dynamic ram main memory.

3) Distribute clocks to the data arrays.

4) Detect and report errors that occur on the backplane.

The MDADD captures the address and command field off the memory backplane. The command and modsel lines are transmitted, along with a command parity bit, to the control array for decoding and comparison.

On receipt of a valid command, the MDADD is responsible for muxing the row and column address to the dynamic rams. The array will function for both 256k and 1M drams.

The refresh address is sent from the control array and must be gated out to the rams when a refresh is requested by the control array.

Eight maintenance clocks are received from the CLKMNT2 array and are distributed to the data arrays. This is done in order to capture the read data if the MDC has been put in maintenance mode (Mclks stopped) while in the middle of a read instruction. The Mclks are "anded" with a signal called CLKREAD to accomplish this. CLKREADL is generated at the proper time in the control array from the free running data clock. Due to various reasons including number of outputs and board location, it was convenient to due this gating in the MDADD.

The MDADD array reports single (correctable) and multi-bit (uncorrectable) errors to the control array. An input has been provided which allows the MDADD to correct the address and command field before registering, or to latch the data as received off the backplane.

10.2.2 MDADD INTERFACES

Inputs:

REFADR(9:0)

Refresh address inputs from the refresh address counter in the control array. The counter will be incremented by either an on card refresh interval counter overflow or a backplane forced refresh command. These lines are low except when a refresh is taking place.

RAMADR(21:0)

This is the memory backplane block address. The MDC requires 19 to 22 bits of address depending on the type and number of RAM chips it contains A board that is half-populated with 256k-bit chips requires 19 bits, while a fully populated board with 1meg-bit chips requires 22 bits.

MODSEL(4:0)

When the module select is "10" hex through "1F" hex, memory data cards are selected. This field is latched in the MDADD and sent to the control array.

MEMREQID(2:0)

This is the memory requestor identification field. These bits are used in the address error correction-detection logic and serve no other function on MDC.

MEMCMD(1:0)

The command bits are latched in the MDADD and transmitted to the control array. Valid commands are:

00 - READ-MODIFY-WRITE (DOUBLE)
01 - WRITE
10 - READ PUBLIC
11 - READ PRIVATE

10.2.2 MDADD INTERFACES (Continued)

The MDC makes no distinction between a read private and read public. The different read types are used by MCACM and IOMC to maintain cache consistency.

ADDRECC(6:0)

The 32 bit address bus is protected by 7 bits of ECC. The ECC encoding encompasses the following 4 fields: command, requestor i.d., module select, and block address.

CLKDAI(8:1)

Eight Mclks from the CLKMNT2 array. These clocks are gated with CLOCKREADL and distributed to the data arrays.

CLKREADL

Normally high. This signal is used to register the read data if the MDC was placed into maintenance mode (Mclks stopped) before the read operation was allowed to complete. This signal will provide a low-to-high transistion on the fifth system clock of a read command.

RAMSIZE

A low indicates 256k-bit RAMS are on the board. A high indicates that 1M-bit RAMS are utilized.

ROWADREN

A high-to-low transition on this input will switch the DRAM address outputs, (ADRAM), from row to column. It will also fire PRECASN.

HOLDADR

From the control array. A high on this input implies a request and therefore all registers in the MDADD are to be held. The only exception is the MEMBUS flip-flop.

10.2.2 MDADD INTERFACES (Continued)

CORRECT

A high on this signal will allow the MDADD to correct all single bit address errors received off the backplane before they are registered. This pin is controlled by a backplane jumper.

ADRAMEN

On a refresh, this signal should go low to allow the refresh address to pass out on ADRAM(9:0).

SHIFT

Serial shift chain enable signal from the clock and maintenance chip.

SHIFTIN

The serial shift chain input.

CLEAR

Maintenance reset signal. This is an asynchronous clear provided CLKAA is high.

MEMBUS

Half the frequency of the system clock. Can be considered as the final qualifier for a request. This signal is latched in the MDADD and sent to the control array.

CLKAA

The normal system Mclk out of the clock and maintenance chip.

10.2.2 MDADD INTERFACES (Continued)

Outputs:

LMEMCMD(1:0)

The latched command bits. Sent to control array.

LMODSEL(4:0)

The latched module select field. Sent to control array.

RASADR(1:0)

These are the RAM address lines which are sent to the control array. There they are decoded to send a RAS to only 1 of 4 banks of memory.

ADRAM(9:0)

The multiplexed row, column, and refresh address to the RAMs.

CLKDAO(8:1)

The eight clocks for the data arrays. The logical "and" of CLKREADL and a normal Mclk.

PRECASN - Low active

This output goes low when the ROWADREN input goes low, i.e. it is time to send the column address out to the RAMS. The signal is then run into a delay line before it becomes the column address strobe (CAS) to the RAMS. The delay line allows time for the address to get to the sips.

LMEMBUS

Latched MEMBUS. This signal is sent to a delay line and then to the control array. This is the only register in MDADD which is not held by HOLDADR.

10.2.2 MDADD INTERFACES (Continued)

ADRPAR1

Address parity bit 1 is the exclusive or of the MODSEL field and RAMADR bits 00, 01, 02, 03, 04, 05, 06, 07, 08, 19, and 20. When CORRECT is high this parity bit is derived from the corrected address field. If not, ADPAR1 will be calculated with the above listed bits as they are received off the backplane. This bit is sent to all data arrays and is XOR'd with data bits 00 and 39 when both writing and reading to RAM memory. This was done to detect an address short in the RAM array. For instance, suppose data was not written to its target address because of a stuck-at-zero address line. Then when reading the stuck-at address, the ADRPAR bits (there are two) will be different (one or both) and the corresponding data bits will not be flipped to their true states. Thus, a double bit error is reported.

ADRPAR2

The XOR of RAMADR bits 09, 10, 11, 12, 13, 14, 15, 16, 17, 18 and 21. This bit is also sent to all data arrays and is XOR'd with data bits 01 and 38. See ADRPAR1 above.

CMDPAR

This is the exclusive-or of the "command" field which is sent across the board to the control array. It is the XOR of SMCTIME, MEMCMD1, MEMCMD0, and the MODSEL bits at the time ADRHOLD goes true. The control array can use this parity bit to check for single bit errors on these inputs from the address array.

ADRERR

Signals the control array that an address error has occurred.

10.2.2 MDADD INTERFACES (Continued)

ADRSBE

Signals the control array that a single bit address error was received.

SHIFTOUT

The serial shift chain output.

JMPRTRU(2:1)

Provides ECL true levels to backplane (BP-S3P5) and to board jumpers.

10.2.3 MDADD FAULT DETECTION

The MDADD can correct all single bit address errors on the backplane. It can detect multi-bit errors. Both errors are reported to the control array and subsequently to the system maintenance controller.

The bits of the command field which cross the pc board to the control array are protected by parity. An error detected here would be reported by the control array as a hardware error - MODBRK.

The MDADD also provides a mechanism for detecting address faults in the RAM matrix. The scheme is described in Section 10.1.2 under ADRPAR1 and will not be repeated here.

10.2.4 MDADD BOARD TEST FACILITIES

The CORRECT input, which in normal operation should be tied true, is provided to latch the address directly off the backplane by bypassing the ECC logic.

10.2.5 MDADD SHIFT CHAIN DEFINITION

FF#	NAME	DESCRIPTION
001	LRAMADR00H	- LATCHED RAM ADDRESS BIT 00
002	LRAMADR01H	- LATCHED RAM ADDRESS BIT 01
003	LRAMADR02H	- LATCHED RAM ADDRESS BIT 02
004	LRAMADR03H	- LATCHED RAM ADDRESS BIT 03
005	LRAMADR04H	- LATCHED RAM ADDRESS BIT 04
006	LRAMADR05H	- LATCHED RAM ADDRESS BIT 05
007	LRAMADR06H	- LATCHED RAM ADDRESS BIT 06
008	LRAMADR07H	- LATCHED RAM ADDRESS BIT 07
009	LRAMADR08H	- LATCHED RAM ADDRESS BIT 08
010	LRAMADR09H	- LATCHED RAM ADDRESS BIT 09
011	LRAMADR10H	- LATCHED RAM ADDRESS BIT 10
012	LRAMADR11H	- LATCHED RAM ADDRESS BIT 11
013	LRAMADR12H	- LATCHED RAM ADDRESS BIT 12
014	LRAMADR13H	- LATCHED RAM ADDRESS BIT 13
015	LRAMADR14H	- LATCHED RAM ADDRESS BIT 14
016	LRAMADR15H	- LATCHED RAM ADDRESS BIT 15
017	LRAMADR16H	- LATCHED RAM ADDRESS BIT 16
018	LRAMADR17H	- LATCHED RAM ADDRESS BIT 17
019	LRAMADR18H	- LATCHED RAM ADDRESS BIT 18
020	LRAMADR19H	- LATCHED RAM ADDRESS BIT 19
021	LRAMADR20H	- LATCHED RAM ADDRESS BIT 20
022	LRAMADR21H	- LATCHED RAM ADDRESS BIT 21
023	LMODSEL0H	- LATCHED MODULE SELECT BIT 0
024	LMODSEL1H	- LATCHED MODULE SELECT BIT 1
025	LMODSEL2H	- LATCHED MODULE SELECT BIT 2
026	LMODSEL3H	- LATCHED MODULE SELECT BIT 3
027	LMODSEL4H	- LATCHED MODULE SELECT BIT 4
028	LMEMCMD0H	- LATCHED MEMORY COMMAND BIT 0
029	LMEMCMD1H	- LATCHED MEMORY COMMAND BIT 1
030	LMEMBUSH	- LATCHED BUS CYCLE TIMING SIGNAL (MEMBUS)
031	AERRORH	- ADDRESS BUS ERROR
032	ASBEH	- ADDRESS BUS SIGNAL BIT ERROR
033	HLDERRH	- HOLD ERROR
034	ADRPAR1H	- ADDRESS BUS PARITY BIT 1 TO DATA ARRAYS
035	ADRPAR2H	- ADDRESS BUS PARITY BIT 2 TO DATA ARRAYS
036	CMDPARH	- COMMAND PARITY BIT TO CONTROL ARRAY

10.2.6 MDADD PIN-OUT

ADDRESS ARRAY PIN NAME	PIN #	I/O	DESCPITION OF PIN SIGNAL
ADDRECC0	R11	I-A	ADDRESS ECC BIT 0 FROM BP.
ADDRECC1	P12	I-A	" " " 1 " ".
ADDRECC2	R12	I-A	" " " 2 " ".
ADDRECC3	P13	I-A	" " " 3 " ".
ADDRECC4	R13	I-A	" " " 4 " ".
ADDRECC5	P14	I-A	" " " 5 " ".
ADDRECC6	R14	I-A	" " " 6 " ".
ADRAM0	A14	O50	ADDRESS BIT 0 THUR LS&B TO RAM SIPS.
ADRAM1	C13	O50	" " 1 " " " " " .
ADRAM2	E15	O50	" " 2 " " " " " .
ADRAM3	B1	O50	" " 3 " " " " " .
ADRAM4	B15	O50	" " 4 " " " " " .
ADRAM5	H15	O50	" " 5 " " " " " .
ADRAM6	D13	O50	" " 6 " " " " " .
ADRAM7	B14	O50	" " 7 " " " " " .
ADRAM8	E3	O50	" " 8 " " " " " .
ADRAM9	J15	O50	" " 9 " " " " " .
ADRAMEN	C4	I-R	RAM ADDRESS ENABLE FROM CTRLA.
ADRERR	B2	O50	ADDRESS ERROR SIGNAL TO CTRLA.
ADRPAR1	E14	O25	ADDRESS PARITY BIT 1 TO ALL DA.
ADRPAR2	F14	O25	ADDRESS PARITY BIT 2 TO ALL DA.
ADRSBE	C3	O50	ADDRESS SINGLE BIT ERROR TO CTRLA.
CLEAR	J2	I-R	CLEARD FROM CLKA.
CLKAA	C12	I-R	D-CLOCK FOR AA D-CHAIN FROM CLKA.
CLKDAI1	G1	I-R	D-CLOCK FOR ROUTING TO DA FROM CLKA.
CLKDAI2	H1	I-R	" " " " " " " .
CLKDAI3	J1	I-R	" " " " " " " .
CLKDAI4	K1	I-R	" " " " " " " .
CLKDAI5	H14	I-R	" " " " " " " .
CLKDAI6	J14	I-R	" " " " " " " .
CLKDAI7	L13	I-R	" " " " " " " .
CLKDAI8	M13	I-R	" " " " " " " .
CLKDAO1	E2	O50	D-CLOCK * CLKREADL TO DA1 CLOCK.
CLKDAO2	F2	O50	" * " " DA2 " .
CLKDAO3	G2	O50	" * " " DA3 " .
CLKDAO4	H2	O50	" * " " DA4 " .
CLKDAO5	F15	O50	" * " " DA5 " .
CLKDAO6	D15	O50	" * " " DA6 " .
CLKDAO7	D14	O50	" * " " DA7 " .
CLKDAO8	G15	O50	" * " " DA8 " .

10.2.6 MDADD PIN-OUT (Continued)

ADDRESS ARRAY PIN NAME	PIN #	I/O	DESCPITION OF PIN SIGNAL
CLKREADL	A7	I-R	PULSE FROM CTRLA (M-MODE CLK READ DATA).
CORRECT	B7	I	ACTIVELY TERMINATED ON MDC.
HOLDADR	A8	I-R	HOLD SIGNAL FROM CTRLA.
CMDPAR	N3	O50	COMMAND PARITY BIT TO CTRLA.
JMPRTRU1	K14	O50	ECL TRUE TO BP JMPRTRUE (BP-S3P5).
JMPRTRU2	K15	O50	ECL TRUE TO ON BOARD JUMPERS.
LMEMCMD0	C1	O50	LATCHED COMMAND BIT 0 TO CTRLA.
LMEMCMD1	D1	O50	LATCHED COMMAND BIT 1 TO CTRLA.
LMODSEL0	C2	O50	LATCHED MODSEL ADDRESS BIT 0 TO CTRLA.
LMODSEL1	D2	O50	" " " " 1 " "
LMODSEL2	D3	O50	LATCHED MODSEL ADDRESS BIT 2 TO CTRLA.
LMODSEL3	E1	O50	" " " " 3 " "
LMODSEL4	F1	O50	" " " " 4 " "
LMEMBUS	N13	O50	LATCHED MEMBUS THRU DELAY LINE TO CTRLA.
MEMCMD0	P15	I-A	COMMAND BIT 0 FROM BP.
MEMCMD1	N14	I-A	COMMAND BIT 1 FROM BP.
MEMRQID0	L1	I-A	REQUESTOR ID BIT 0 FROM BP.
MEMRQID1	L2	I-A	" " " 1 " ".
MEMRQID2	M1	I-A	" " " 2 " ".
MODSEL0	N15	I-A	MODULE SELECT BIT 0 FROM BP.
MODSEL1	M14	I-A	" " " 1 " ".
MODSEL2	M15	I-A	" " " 2 " ".
MODSEL3	L14	I-A	" " " 3 " ".
MODSEL4	L15	I-A	" " " 4 " ".
PRECASN	C14	O50	PRECASN TO DELAY LINE TO GENERATE CASN.
RAMADR00	M2	I-A	ADRRESS BIT 00 FROM BP.
RAMADR01	N1	I-A	" " 01 " ".
RAMADR02	N2	I-A	" " 02 " ".
RAMADR03	P1	I-A	ADDRESS BIT 03 FROM BP.
RAMADR04	P2	I-A	" " 04 " ".
RAMADR05	R2	I-A	" " 05 " ".
RAMADR06	R3	I-A	" " 06 " ".
RAMADR07	P4	I-A	" " 07 " ".
RAMADR08	R4	I-A	" " 08 " ".
RAMADR09	P5	I-A	" " 09 " ".
RAMADR10	P5	I-A	" " 10 " ".
RAMADR11	P6	I-A	" " 11 " ".
RAMADR12	R6	I-A	" " 12 " ".

10.2.6 MDADD PIN-OUT (Continued)

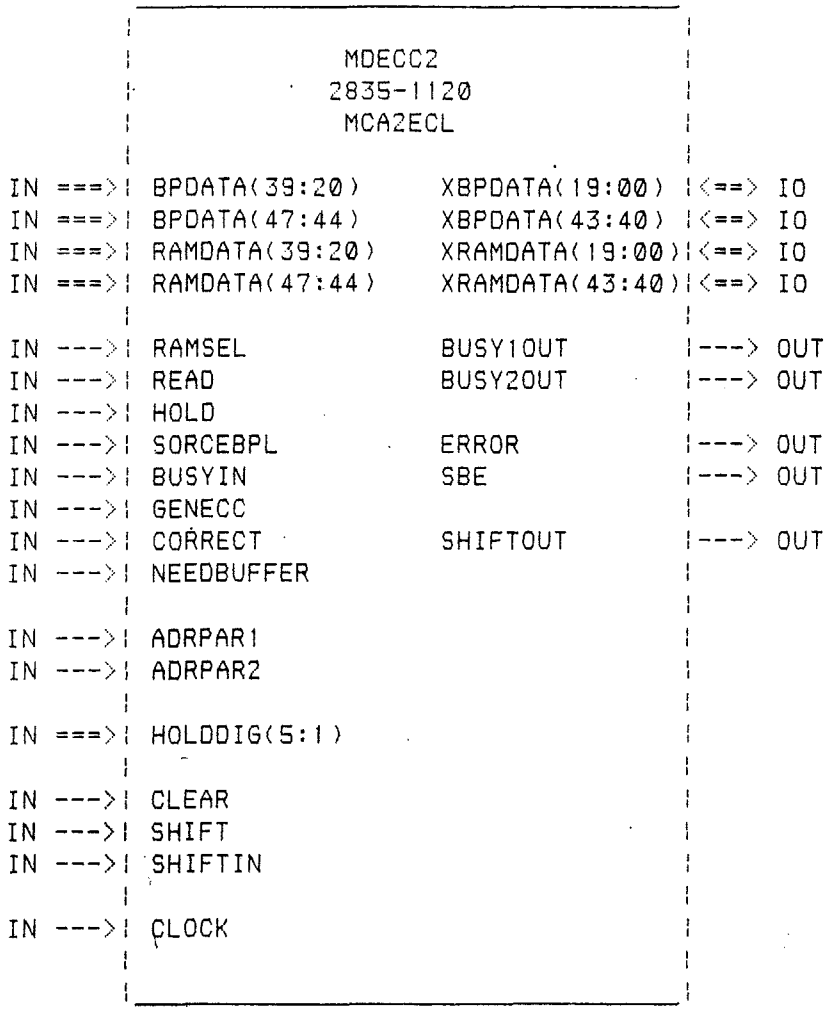
ADDRESS ARRAY PIN NAME	PIN #	I/O	DESCPITION OF PIN SIGNAL
RAMADR13	P7	I-A	ADDRESS BIT 13 FROM BP.
RAMADR14	R7	I-A	" " 14 " ".
RAMADR15	P8	I-A	" " 15 " ".
RAMADR16	R8	I-A	" " 16 " ".
RAMADR17	P9	I-A	" " 17 " ".
RAMADR18	R9	I-A	" " 18 " ".
RAMADR19	P10	I-A	" " 19 " ".
RAMADR20	R10	I-A	" " 20 " ".
RAMADR21	P11	I-A	" " 21 " ".
RAMSIZE	A13	I	RAM SIZE (256K/1M) FROM BD JMPR.
RASADR0	E13	O50	RAS ADDRESS BIT 0 TO CTRLA.
RASADR1	G14	O50	RAS ADDRESS BIT 1 TO CTRLA.
REFADR0	A5	I-R	REFRESH ADDRESS BIT 0 FROM CTRLA.
REFADR1	B5	I-R	REFRESH ADDRESS BIT 1 FROM CTRLA.
REFADR2	A6	I-R	REFRESH ADDRESS BIT 2 FROM CTRLA.
REFADR3	B6	I-R	REFRESH ADDRESS BIT 3 FROM CTRLA.
REFADR4	A9	I-R	REFRESH ADDRESS BIT 4 FROM CTRLA.
REFADR5	B9	I-R	REFRESH ADDRESS BIT 5 FROM CTRLA.
REFADR6	A10	I-R	REFRESH ADDRESS BIT 6 FROM CTRLA.
REFADR7	B10	I-R	REFRESH ADDRESS BIT 7 FROM CTRLA.
REFADR8	A12	I-R	REFRESH ADDRESS BIT 8 FROM CTRLA.
REFADR9	B12	I-R	REFRESH ADDRESS BIT 9 FROM CTRLA.
ROWADREN	N4	I-R	RAM ROW ADDRESS ENABLE FROM CTRLA.
SHIFT	N5	I-R	SHIFTD FROM CLKA.
SHIFTIN	A3	I-R	SHIFTIN FROM DA8 SHIFTOUT.
SHIFTOUT	A2	O50	SHIFTOUT TO CLKA SHIFTIN.
MEMBUS	P3	I-A	MEMBUS FROM BP.

10.2.6 MDADD PIN-OUT (Continued)

DEFINITION OF ABBREVIATIONS:

AA	= ADDRESS ARRAY.
CLKA	= CLOCK ARRAY.
CTRLA	= CONTROL ARRAY.
BD	= CIRCUIT BOARD.
BP	= BACKPLANE.
DA#	= DATA ARRAY, WHERE # IS THE USAGE NUMBER (1 TO 8).
I	= INPUT, WITHOUT TERMINATION.
I-A	= INPUT, WITH ACTIVE TERMINATION.
I-R	= INPUT, WITH 50 OHM RESISTIVE TERMINATION.
LS&B	= ECL-TTL LEVEL SHIFTERS & TTL BUFFERS.
RDATA###	= BIDIRECTIONAL DATA BUS BETWEEN THE DATA ARRAYS AND THE ECL/TTL LEVEL SHIFTERS. ### IS THE BIT NUMBER (000 TO 191).
XDATA###	= BIDIRECTIONAL DATA BUS BETWEEN THE DATA ARRAYS AND THE MDC'S BACKPLANE CONNECTOR. ### IS THE BIT NUMBER (000 TO 191).

10.3 DATA ARRAY - MDECC2



10.3.1 MDECC2 FUNCTIONAL OVERVIEW

The MDECC2 option is an ECC generator and SEC-DED-4ED bidirectional 48 bit bus transceiver. Two MDECC2 PGA's in a mirror image configuration are required to implement the function because of the limited number of 25 ohm drivers on the MCA2ECL chip. Each PGA must see all data and check bits in order for the ECC circuitry to function properly, however, each PGA is only responsible for driving 20 data bits and 4 check bits. See Section 10.3.6 for pin-out and data array pair interconnect.

10.3.2 MDECC2 INTERFACES

Inputs:

BPDATA(39:20)

Twenty bits of backplane data.

BPDATA(47:44)

Four check bits off the backplane.

RAMDATA(39:20)

Twenty bits of ram data.

RAMDATA(47:44)

Four check bits from ram.

RAMSEL

Selects the input to the ECC circuitry and data registers. A high chooses the RAMDATA whereas a low will select BPDATA. Remember that due to the number of output drivers half the ram data will enter on inputs RAMDATA(39:20) and RAMDATA(47:44), while the other 24 bits will invade I/O lines XRAMDATA(19:0) and XRAMDATA(43:0). Therefore, his I/O bus should be in receive mode (READ HIGH) while RAMSEL is high. See READ below.

10.3.2 MDECC2 INTERFACES (Continued)

READ

A true will allow the read data from ram memory to be received on I/O pins XRAMDATA (19:0) and XRAMDATA(43:40).

HOLD

A high on HOLD will place all data and error registers into the hold mode.

SORCEBPL - active low

A low on source backplane allows I/O pins XBPDATA(19:0) and XBPDATA(43:40) to drive data onto the backplane. This signal from the control array goes low on the fifth clock of a read cycle. It is then latched and extended another clock period in order to present data to the backplane for two clock periods.

BUSYIN

From the control array. There are two "busy" flip-flops in the MDECC2, for a total of sixteen on the board. Upon system initialization, the maintenance processor will shift a one into the flip-flop which corresponds to the MDC's logical address. BUSYIN is received by all data arrays and will gate one of sixteen "busies" onto the backplane. If NEEDBUFFER is high, BUSY1OUT and BUSY2OUT will be the noninverted buffered output of BUSYIN.

GENECC

A true on GENECC forces the data array pair to generate 8 check bits from the 40 bit data word. The data present on the syndrome input lines are ignored when GENECC is active. See Section 10.1.3 for the check bit encode table.

10.3.2 MDECC2 INTERFACES (Continued)

CORRECT

From the control array. When CORRECT is high, single bit errors will be detected and corrected before the data is clocked into the registers. On the MDC this signal is always high when write data is captured off the backplane. In order to correct read data errors it is necessary to shift a one into the CORRECT flip-flop in the control array. See Section 10.3.4 for the syndrome decode table.

NEEDBUFFER

Tied to VTT on the MDC. The 25 ohm outputs BUSY1OUT and BUSY2OUT will, when NEEDBUFFER is high, be the noninverted buffered output of BUSYIN delayed by roughly 2.5 ns.

ADRPAR1

Address parity bit 1 is the exclusive or of the MODSEL field and RAMADR bits 00, 01, 02, 03, 04, 05, 06, 07, 08, 19, and 20. When CORRECT on the address array is high, this bit is derived from the corrected address field. If address correction is not utilized ADPAR1 will be calculated with the above listed bits as they are received off the backplane. This bit is sent to all data arrays and is XOR'd with data bits 00 and 39 when both writing and reading to RAM memory. This was done to detect an address short in the RAM array. For instance, suppose data was not written to its target address because of a stuck-at-zero address line. Then when reading the stuck-at address, the ADRPAR bits (there are two) will be different (one or both) and the corresponding data bits will not be flipped to their true states. Thus, a double bit error is reported.

ADRPAR2

The XOR of RAMADR bits 09, 10, 11, 12, 13, 14, 15, 16, 17, 18 and 21. This bit is also sent to all data arrays and is XOR'd with data bits 01 and 38. See ADRPAR1 above.

10.3.2 MDECC2 INTERFACES (Continued)

HOLDDIG(5:1)

Each data array has 24 registers - 20 data and 4 check bit. HOLDDIG(1) will place a digit, data registers 00,01,02,03, into the hold state. All five digits have their own individual HOLD line. The check bits, of course, have no hold control. These lines are tied to VTT on the MDC.

CLEAR

Maintenance chain reset signal originating at the clock and maintenance chip. This signal is an asynchronous clear provided the clock is high.

SHIFT

Maintenance chain serial shift enable signal from the clock and maintenance chip.

SHIFTIN

The maintenance chain serial shift input.

10.3.2 MDECC2 INTERFACES (Continued)

I/O SIGNALS

XBPDATA(19:00)

Half of the 40 bit backplane data word. Each array handles 20 bits. Remember that each array "sees" all 48 bits but only drives half the data and half the check bits. XBPDATA(19:0) is the half of the data word that is driven onto the backplane on read commands.

XBPDATA(43:40)

Half of the 8 check bits. Driven onto the backplane on a read cycle.

XRAMDATA(39:20)

Half of the 40 bit ram data word. Driven to the ram matrix during a write cycle.

XRAMDATA(43:40)

Half of the 8 check bits. Driven to the ram matrix on write commands.

10.3.2 MDECC2 INTERFACES (Continued)

OUTPUTS

ERROR and SBE

ERROR and SBE work in conjunction with one another and in conjunction with the same signals of the other data array comprising the pair. The possible combinations for data array pair 1 and 2 are as follows:

ERROR1 = 0	SBE1 = 0	No error.
ERROR2 = 0	SBE2 = 0	
ERROR1 = 0	SBE1 = 1	Single bit error
ERROR2 = 1	SBE2 = 0	in D.A. 1.
ERROR1 = 1	SBE1 = 0	Single bit error
ERROR2 = 0	SBE2 = 1	in D.A. 2.
ERROR1 = 1	SBE1 = 0	Multi-bit error.
ERROR2 = 1	SBE2 = 0	

A single bit error in data array 1 means that the error in the 48 bit word is one of the 24 bits that gets registered in data array 1. Basically, the array pair look at the 48 bits, and if one array sees an error which is not a bit that is registered on chip, he will raise his ERROR line saying an error occurred but it is not in me. If it was a single-bit-error the other array will make SBE true. A multi-bit error will cause both error lines to go high. Theses error lines go to the control array for decoding and reporting.
It should be noted for the sake of completeness that the actual data in the error registers is different from the above (the array OUTPUTS). This is an important consideration when evaluating this information via the shift chain. The difference is, for a single bit error, both ERROR and SBE will be high if the bit in error has its flip-flop in that array. This discrepency is due to the use of an XOR on the output side of the error FF's.

10.3.2 MDECC2 INTERFACES (Continued)

BUSY1OUT and BUSY2OUT

There are 2 "busy" flip-flops in each data array for a total of 16 on the MDC. On system configuration, the system maintenance controller is responsible for shifting a 1 into the flip-flop, which corresponds to the cards' logical address. If a 1 was shifted into the BUSY1 flip-flop, BUSY1OUT will go true on receipt of a BUSYIN going high. Similarly with BUSY2OUT. These 25 ohm outputs may also be used as buffers. See Inputs: NEEDBUFFER. On the MDC, they are always used as "busies" as NEEDBUFFER is tied to VTT.

SHIFTOUT

The maintenance chain serial shift output.

10.3.3 MDECC2 CHECK BIT ENCODE TABLE

	7	6	5	4	3	2	1	0
00 :=				X			X	X
01 :=			X				X	X
02 :=		X					X	X
03 :=	X						X	X
04 :=				X		X		X
05 :=			X			X		X
06 :=		X				X		X
07 :=	X					X		X
08 :=				X		X	X	
09 :=			X			X	X	
10 :=		X			X	X		
11 :=	X				X	X		
12 :=				X	X	X		
13 :=			X		X	X		
14 :=		X			X			X
15 :=	X				X			X
16 :=				X	X		X	
17 :=			X		X		X	
18 :=		X			X		X	
19 :=	X				X		X	
20 :=		X		X				X
21 :=		X		X			X	
22 :=		X		X		X		
23 :=		X		X	X			
24 :=	X			X				X
25 :=	X			X			X	
26 :=			X	X		X		
27 :=			X	X	X			
28 :=			X	X				X
29 :=			X	X			X	
30 :=			X			X		
31 :=		X	X		X			
32 :=	X	X	X					X
33 :=	X		X				X	
34 :=	X		X			X		
35 :=	X		X		X			
36 :=	X	X						X
37 :=	X	X					X	
38 :=	X	X				X		
39 :=	X	X			X			

10.3.4 MDECC2 SYNDROME DECODE TO BIT-IN-ERROR TABLE

	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0:	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1:	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
2:	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
76543:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0000 :	*	40	41	D	42	D	D	M	43	C	D	M	D	M	M	D
0001 :	44	D	D	00	D	04	08	D	D	M	16	D	12	D	D	M
0010 :	45	D	D	01	D	05	09	D	D	M	17	D	13	D	D	M
0011 :	D	28	29	D	26	D	D	M	27	D	D	M	D	M	M	D
0100 :	46	D	D	02	D	06	N	D	D	14	18	D	10	D	D	M
0101 :	D	20	21	D	22	D	D	M	23	D	D	M	D	M	M	D
0110 :	D	M	M	D	30	D	D	M	31	D	D	M	D	M	M	D
0111 :	M	D	D	M	D	M	M	D	D	M	M	D	M	D	D	M
1000 :	47	D	D	03	D	07	M	D	D	15	19	D	11	D	D	M
1001 :	D	24	25	D	M	D	D	M	M	D	D	M	D	M	M	D
1010 :	D	32	33	D	34	D	D	M	39	D	D	M	D	M	M	D
1011 :	M	4D	4D	M	4D	M	M	D	4D	M	M	D	M	D	D	M
1100 :	D	36	37	D	38	D	D	M	39	D	D	M	D	M	M	D
1101 :	M	D	D	M	D	M	M	D	D	M	M	D	M	D	D	M
1110 :	M	D	D	M	D	M	M	D	D	M	M	D	M	D	D	M
1111 :	D	M	M	D	M	D	D	M	M	D	D	M	D	M	M	D

* = NO BITS IN ERROR
00 = SINGLE BIT IN ERROR, WHERE "nn" IS THE BAD BIT
D = MULTIPLE ERRORS (EVEN NUMBER)
M = MULTIPLE ERRORS (ODD NUMBER)

NOTE THAT BITS 40 THRU 47 ARE THE CHECK BITS.

10.3.5 MDECC2 SHIFT CHAIN DEFINITION

FF#	NAME	DESCRIPTION
001	LRAMDATA000H	LATCHED RAM DATA BUS DATA BIT 000
002	LRAMDATA001H	LATCHED RAM DATA BUS DATA BIT 001
003	LRAMDATA002H	LATCHED RAM DATA BUS DATA BIT 002
004	LRAMDATA003H	LATCHED RAM DATA BUS DATA BIT 003
005	LRAMDATA004H	LATCHED RAM DATA BUS DATA BIT 004
006	LRAMDATA005H	LATCHED RAM DATA BUS DATA BIT 005
007	LRAMDATA006H	LATCHED RAM DATA BUS DATA BIT 006
008	LRAMDATA007H	LATCHED RAM DATA BUS DATA BIT 007
009	LRAMDATA008H	LATCHED RAM DATA BUS DATA BIT 008
010	LRAMDATA009H	LATCHED RAM DATA BUS DATA BIT 009
011	LRAMDATA010H	LATCHED RAM DATA BUS DATA BIT 000
012	LRAMDATA011H	LATCHED RAM DATA BUS DATA BIT 011
013	LRAMDATA012H	LATCHED RAM DATA BUS DATA BIT 012
014	LRAMDATA013H	LATCHED RAM DATA BUS DATA BIT 013
015	LRAMDATA014H	LATCHED RAM DATA BUS DATA BIT 014
016	LRAMDATA015H	LATCHED RAM DATA BUS DATA BIT 015
017	LRAMDATA016H	LATCHED RAM DATA BUS DATA BIT 016
018	LRAMDATA017H	LATCHED RAM DATA BUS DATA BIT 017
019	LRAMDATA018H	LATCHED RAM DATA BUS DATA BIT 018
020	LRAMDATA019H	LATCHED RAM DATA BUS DATA BIT 019
021	LRAMECC000H	LATCHED RAM DATA ECC BUS DATA BIT 000
022	LRAMECC001H	LATCHED RAM DATA ECC BUS DATA BIT 001
023	LRAMECC002H	LATCHED RAM DATA ECC BUS DATA BIT 002
024	LRAMECC003H	LATCHED RAM DATA ECC BUS DATA BIT 003
025	SBE1H	SINGLE BIT ERROR DETECTED BY DATA ARRAY 1
026	ERROR1H	ERROR CONDITION DETECTED BY DATA ARRAY 1
027	SBPHOLD1H	SOURCE BACKPLANE HOLD, DATA ARRAY 1
028	BUSY(00)	MDC BACKPLANE BUSY
029	BUSY(01)	MDC BACKPLANE BUSY

10.3.6 MDECC2 DATA ARRAY PIN-OUT

DATA ARRAY SIGNAL NAME	PIN #	I/O	DATA ARRAY 1 (DA1) PIN CONNECTS TO -	DATA ARRAY 2 (DA2) PIN CONNECTS TO -
ADRPAR1	D2	I	AA - ADRPAR1	AA - ADRPAR1
ADRPAR2	E2	I	AA - ADRPAR2	AA - ADRPAR2
BPDATA20	R5	I	BD - XRAMDTm(20)P	BD - XRAMDTm(19)P
BPDATA21	A5	I	BD - XRAMDTm(21)P	BD - XRAMDTm(18)P
BPDATA22	R6	I	BD - XRAMDTm(22)P	BD - XRAMDTm(17)P
BPDATA23	A6	I	BD - XRAMDTm(23)P	BD - XRAMDTm(16)P
BPDATA24	R7	I	BD - XRAMDTm(24)P	BD - XRAMDTm(15)P
BPDATA25	A7	I	BD - XRAMDTm(25)P	BD - XRAMDTm(14)P
BPDATA26	R8	I	BD - XRAMDTm(26)P	BD - XRAMDTm(13)P
BPDATA27	A8	I	BD - XRAMDTm(27)P	BD - XRAMDTm(12)P
BPDATA28	R9	I	BD - XRAMDTm(28)P	BD - XRAMDTm(11)P
BPDATA29	A9	I	BD - XRAMDTm(29)P	BD - XRAMDTm(10)P
BPDATA30	R10	I	BD - XRAMDTm(30)P	BD - XRAMDTm(09)P
BPDATA31	A10	I	BD - XRAMDTm(31)P	BD - XRAMDTm(08)P
BPDATA32	R11	I	BD - XRAMDTm(32)P	BD - XRAMDTm(07)P
BPDATA33	A11	I	BD - XRAMDTm(33)P	BD - XRAMDTm(06)P
BPDATA3	R12		B - XRAMDTm(34)	B - XRAMDTm(05)P
BPDATA35	A12	I	BD - XRAMDTm(35)P	BD - XRAMDTm(04)P
BPDATA36	A13	I	BD - XRAMDTm(36)P	BD - XRAMDTm(03)P
BPDATA37	R13	I	BD - XRAMDTm(37)P	BD - XRAMDTm(02)P
BPDATA38	P15	I	BD - XRAMDTm(38)P	BD - XRAMDTm(01)P
BPDATA3	B15		B - XRAMDTm(39)	B - XRAMDTm(00)P
BPDATA44	R3	I	BD - XRAMECm(04)P	BD - XRAMECm(03)P
BPDATA45	A3	I	BD - XRAMECm(05)P	BD - XRAMECm(02)P
BPDATA46	R4	I	BD - XRAMECm(06)P	BD - XRAMECm(01)P
BPDATA47	A4	I	BD - XRAMECm(07)P	BD - XRAMECm(00)P
BUSYIN	N1	I	CTRLA - BUSY1H	BD - BUSY2H
BUSY1OUT	L3	O25	BD - BUSY0(0)	BD - BUSY0(2)
BUSY2OUT	M13	O25	BD - BUSY0(1)	BD - BUSY0(3)
CLEAR	G14	I	CLKA - CLEAR	CLKA - CLEAR
CLOCK	C12	I	CLKA - CLOCK	CLKA - CLOCK
CORRECT	H14	I	CTRLA - CORRECT	CTRLA - CORRECT
ERROR	N13	O50	CTRLA - ERROR1	CTRLA - ERROR2
GENECC	L14	I	BD - VTT (LOW)	BD - VTT (LOW)
HOLD	M14	I	CTRLA - HOLD	CTRLA - HOLD
HOLDDIG1	N11	I	BD - VTT (LOW)	BD - VTT (LOW)
HOLDDIG2	N12	I	BD - VTT (LOW)	BD - VTT (LOW)
HOLDDIG3	N5	I	BD - VTT (LOW)	BD - VTT (LOW)
HOLDDIG4	C11	I	BD - VTT (LOW)	BD - VTT (LOW)
HOLDDIG5	C5	I	BD - VTT (LOW)	BD - VTT (LOW)

10.3.6 MDECC2 DATA ARRAY PIN-OUT (Continued)

DATA ARRAY SIGNAL NAME	PIN #	I/O	DATA ARRAY 1 (DA1) PIN CONNECTS TO -	DATA ARRAY 2 (DA2) PIN CONNECTS TO -
NEEDBUFFER	C15	I	BD - VTT (LOW)	BD - VTT (LOW)
RAMDAT20	P1	I	BD - RDATAm(20)	BD - RDATAm(19)
RAMDAT21	B1	I	BD - RDATAm(21)	BD - RDATAm(18)
RAMDAT22	P3	I	BD - RDATAm(22)	BD - RDATAm(17)
RAMDAT23	B3	I	BD - RDATAm(23)	BD - RDATAm(16)
RAMDAT24	P4	I	BD - RDATAm(24)	BD - RDATAm(15)
RAMDAT25	B4	I	BD - RDATAm(25)	BD - RDATAm(14)
RAMDAT26	P5	I	BD - RDATAm(26)	BD - RDATAm(13)
RAMDAT27	B5	I	BD - RDATAm(27)	BD - RDATAm(12)
RAMDAT28	P6	I	BD - RDATAm(28)	BD - RDATAm(11)
RAMDAT29	B6	I	BD - RDATAm(29)	BD - RDATAm(10)
RAMDAT30	P7	I	BD - RDATAm(30)	BD - RDATAm(09)
RAMDAT31	B7	I	BD - RDATAm(31)	BD - RDATAm(08)
RAMDAT32	P8	I	BD - RDATAm(32)	BD - RDATAm(07)
RAMDAT33	B8	I	BD - RDATAm(33)	BD - RDATAm(06)
RAMDAT34	P9	I	BD - RDATAm(34)	BD - RDATAm(05)
RAMDAT35	B9	I	BD - RDATAm(35)	BD - RDATAm(04)
RAMDAT36	P10	I	BD - RDATAm(36)	BD - RDATAm(03)
RAMDAT37	B10	I	BD - RDATAm(37)	BD - RDATAm(02)
RAMDAT38	P11	I	BD - RDATAm(38)	BD - RDATAm(01)
RAMDAT39	B11	I	BD - RDATAm(39)	BD - RDATAm(00)
RAMDAT44	P12	I	BD - RDECCm(04)	BD - RDECCm(03)
RAMDAT45	B12	I	BD - RDECCm(05)	BD - RDECCm(02)
RAMDAT46	P13	I	BD - RDECCm(06)	BD - RDECCm(01)
RAMDAT47	B13	I	BD - RDECCm(07)	BD - RDECCm(00)
RAMSEL	H2	I	CTRLA - RAMSEL	CTRLA - RAMSEL
READ	G2	I	CTRLA - READ	CTRLA - READ
SBE	J14	050	CTRLA - SBE1	CTRLA - SBE2
SHIFT	N4	I	CLKA - SHIFT	CLKA - SHIFT
SHIFTIN	C4	I	CTRLA - SHOUTD	DA1 - SHIFTOUT
SHIFTOUT	C3	050	DA2 - SHIFTIN	DA3 - SHIFTIN
SORCEBPL	J2	I	CTRLA - SOURCEBPL	CTRLA - SOURCEBPL
XBPDAT00	D2	B25	BD - XRAMDTm(00)P	BD - XRAMDTm(39)P
XBPDAT01	D3	B25	BD - XRAMDTm(01)P	BD - XRAMDTm(38)P
XBPDAT02	E1	B25	BD - XRAMDTm(02)P	BD - XRAMDTm(37)P
XBPDAT03	E3	B25	BD - XRAMDTm(03)P	BD - XRAMDTm(36)P
XBPDAT04	F1	B25	BD - XRAMDTm(04)P	BD - XRAMDTm(35)P
XBPDAT05	F2	B25	BD - XRAMDTm(05)P	BD - XRAMDTm(34)P
XBPDAT06	K1	B25	BD - XRAMDTm(06)P	BD - XRAMDTm(33)P

10.3.6 MDECC2 DATA ARRAY PIN-OUT (Continued)

DATA ARRAY SIGNAL NAME	PIN #	I/O	DATA ARRAY 1 (DA1) PIN CONNECTS TO -	DATA ARRAY 2 (DA2) PIN CONNECTS TO -
XBPDAT07	K2	B2S	BD - XRAMDTm(07)P	BD - XRAMDTm(32)P
XBPDAT08	L2	B2S	BD - XRAMDTm(08)P	BD - XRAMDTm(31)P
XBPDAT09	M2	B2S	BD - XRAMDTm(09)P	BD - XRAMDTm(30)P
XBPDAT10	M3	B2S	BD - XRAMDTm(10)P	BD - XRAMDTm(29)P
XBPDAT11	N2	B2S	BD - XRAMDTm(11)P	BD - XRAMDTm(28)P
XBPDAT12	L13	B2S	BD - XRAMDTm(12)P	BD - XRAMDTm(27)P
XBPDAT13	N14	B2S	BD - XRAMDTm(13)P	BD - XRAMDTm(26)P
XBPDAT14	L15	B2S	BD - XRAMDTm(14)P	BD - XRAMDTm(25)P
XBPDAT15	K14	B2S	BD - XRAMDTm(15)P	BD - XRAMDTm(24)P
XBPDAT16	K15	B2S	BD - XRAMDTm(16)P	BD - XRAMDTm(23)P
XBPDAT17	F14	B2S	BD - XRAMDTm(17)P	BD - XRAMDTm(22)P
XBPDAT18	F15	B2S	BD - XRAMDTm(18)P	BD - XRAMDTm(21)P
XBPDAT19	E13	B2S	BD - XRAMDTm(19)P	BD - XRAMDTm(20)P
XBPDAT40	E14	B2S	BD - XRAMECm(00)P	BD - XRAMECm(07)P
XBPDAT41	D13	B2S	BD - XRAMECm(01)P	BD - XRAMECm(06)P
XBPDAT42	D14	B2S	BD - XRAMECm(02)P	BD - XRAMECm(05)P
XBPDAT43	C14	B2S	BD - XRAMECm(03)P	BD - XRAMECm(04)P
XRAMDT00	G1	B50	BD - RDATAm(00)	BD - RDATAm(39)
XRAMDT01	D1	B50	BD - RDATAm(01)	BD - RDATAm(38)
XRAMDT02	J1	B50	BD - RDATAm(02)	BD - RDATAm(37)
XRAMDT03	H1	B50	BD - RDATAm(03)	BD - RDATAm(36)
XRAMDT04	M1	B50	BD - RDATAm(04)	BD - RDATAm(35)
XRAMDT05	L1	B50	BD - RDATAm(05)	BD - RDATAm(34)
XRAMDT06	P2	B50	BD - RDATAm(06)	BD - RDATAm(33)
XRAMDT07	N3	B50	BD - RDATAm(07)	BD - RDATAm(32)
XRAMDT08	M15	B50	BD - RDATAm(08)	BD - RDATAm(31)
XRAMDT09	N15	B50	BD - RDATAm(09)	BD - RDATAm(30)
XRAMDT10	H15	B50	BD - RDATAm(10)	BD - RDATAm(29)
XRAMDT11	J15	B50	BD - RDATAm(11)	BD - RDATAm(28)
XRAMDT12	E15	B50	BD - RDATAm(12)	BD - RDATAm(27)
XRAMDT13	G15	B50	BD - RDATAm(13)	BD - RDATAm(26)
XRAMDT14	C13	B50	BD - RDATAm(14)	BD - RDATAm(25)
XRAMDT15	D15	B50	BD - RDATAm(15)	BD - RDATAm(24)
XRAMDT16	B14	B50	BD - RDATAm(16)	BD - RDATAm(23)
XRAMDT17	P14	B50	BD - RDATAm(17)	BD - RDATAm(22)
XRAMDT18	A14	B50	BD - RDATAm(18)	BD - RDATAm(21)
XRAMDT19	R14	B50	BD - RDATAm(19)	BD - RDATAm(20)
XRAMDT40	A2	B50	BD - RDECCm(00)	BD - RDECCm(07)
XRAMDT41	R2	B50	BD - RDECCm(01)	BD - RDECCm(06)
XRAMDT42	C1	B50	BD - RDECCm(02)	BD - RDECCm(05)
XRAMDT43	B2	B50	BD - RDECCm(03)	BD - RDECCm(04)

10.3.6 MDECC2 DATA ARRAY PIN-OUT (Continued)

DEFINITION OF ABBREVIATIONS:

AA	= ADDRESS ARRAY.
B25	BI-DIRECTIONAL SIGNAL WITH 25 OHM OUTPUT DRIVE.
B50	= BI-DIRECTIONAL SIGNAL WITH 50 OHM OUTPUT DRIVE.
CLKA	= CLOCK ARRAY.
CTRLA	= CONTROL ARRAY.
BD	= CIRCUIT BOARD.
DA#	= DATA ARRAY, WHERE - DA1 IS FOR DATA BITS 00 THRU 19 AND DA2 IS FOR DATA BITS 20 THRU 39.
I	= INPUT SIGNAL.
O25	= OUTPUT SIGNAL WITH 25 OHM OUTPUT DRIVE.
O50	= OUTPUT SIGNAL WITH 50 OHM OUTPUT DRIVE.
RDATAm(##)	= BIDIRECTIONAL DATA BUS BETWEEN THE DATA ARRAYS AND THE ECL/TTL LEVEL SHIFTERS. ##=00:39, m=0:3.
RDECCm(##	BIDIRECTIONAL ECC DATA BETWEEN THE DATA ARRAYS AND THE ECL/TTL LEVEL SHIFTERS. ##=00:39, m=0:3.
XRAMDTm(##)P	= BIDIRECTIONAL DATA BUS BETWEEN THE DATA ARRAYS AND THE MDC'S BACKPLANE CONNECTOR. ##=00: m=0:3.
XRAMECm(##)	BIDIRECTIONAL ECC DATA BETWEEN THE DATA ARRAYS AND THE MDC"S BACKPLANE CONNECTOR. ##=00: m=0:3.

11 TERMS AND DEFINITIONS

CAS Column Address Strobe. The CAS is one of the chip select pins on Dynamic RAMS. It is used to latch the column address from the multiplexed address pins.

DRAM Dynamic random access memory, commonly available in 16k bit, 64k bit, 256k bit, and 1M bit varieties. They are generally inexpensive , medium speed (around 120 nS), volatile storage devices.

IOMC The I/O memory concentrator. The IOMC buffers data transfers from the Data Transfer Modules and the I/O Translator. It also maintains the time of day clock for the V500 System.

IOT The I/O Translator. The IOT manages the queuing and dequeuing of I/O requests from the MCP. It schedules I/O operations for the Data Transfer Modules.

MCACM Memory Control and Cache Module. The interface from the processor to the main memory storage structure.

MCP The Master Control Program. This is the V500 operating system.

MDC Memory Data Card as defined in this specification.

RAS Row Address Strobe. The RAS is one of the chip select pins on Dynamic RAMS. It is used to latch in the row address through the multiplexed address pins.

SMC The System Maintenance Controller. The SMC is responsible for controlling clocks and shifting data and maintenance chains in the V500 System.

APPENDIX A

CHAIN LIST

V500 MDC MAINTENANCE CHAIN

```
DATE     1102786;

CHAIN    YS;                    % Maintenance chain

                                % exit board

                                % exit CLKMNT2 array
         ERRORIG        1;      % error ignore ff
         OVERRUN        1;      %
         CNTRFF         1;      %
         CLKBAD         1;      %
         SKEWL          1;      %
         SKEWH          1;      %
         MODIFYSKEW     1;      %
         SKEWREG        9;      %
                                % enter CLKMNT2 array

                                % exit MDADDR array
         CMDPARH        1;      % command parity bit
         ADRPARH        2;      % address bus parity bits
         HLDERRH        1;      % hold error
         ASBEH          1;      % address bus single bit error
         AERRORH        1;      % address bus error
         LMEMBUSH       1;      % latched bus cycle timing signal
         LMEMCMDH       2;      % latched memory command
         LMODSELH       5;      % latched module select
         LRAMADRH      22;      % latched ram address
                                % enter MDADDR array

                                % exit MDDATA array 8
         BUSY15H        1;      % backplane busy
         BUSY14H        1;      % backplane busy
         SBPHOLD8H      1;      % source backplane hold
         ERROR8H        1;      % error ff
         SBE8H          1;      % single bit error
         LRAMECC34      1;      % latched ram data ECC
         LRAMECC35      1;
         LRAMECC36      1;
         LRAMECC37      1;
```

APPENDIX A (Continued)

```
LRAMDAT320    1;        % latched ram data bus
LRAMDAT321    1;
LRAMDAT322    1;
LRAMDAT323    1;
LRAMDAT324    1;
LRAMDAT325    1;
LRAMDAT326    1;
LRAMDAT327    1;
LRAMDAT328    1;
LRAMDAT329    1;
LRAMDAT330    1;
LRAMDAT331    1;
LRAMDAT332    1;
LRAMDAT333    1;
LRAMDAT334    1;
LRAMDAT335    1;
LRAMDAT336    1;
LRAMDAT337    1;
LRAMDAT338    1;
LRAMDAT339    1;
                        % enter MDDATA 8

                        % exit MDDATA 7
BUSY13H       1;        % backplane busy
BUSY12H       1;        % backplane busy
SBPHOLD7H     1;        % source backplane hold
ERROR7H       1;        % error detected
SBE7H         1;        % single bit error
LRAMECC3X     4;        % latched ram data ECC
LRAMDAT3X    20;        % latched ram data
                        % enter MDDATA 7

                        % exit MDDATA array 6
BUSY11H       1;        % backplane busy
BUSY10H       1;        % backplane busy
SBPHOLD6H     1;        % source backplane hold
ERROR6H       1;        % error ff
SBE6H         1;        % single bit error
LRAMECC24     1;        % latched ram data ECC
LRAMECC25     1;
LRAMECC26     1;
LRAMECC27     1;
```

APPENDIX A (Continued)

```
LRAMDAT220    1;        % latched ram data bus
LRAMDAT221    1;
LRAMDAT222    1;
LRAMDAT223    1;
LRAMDAT224    1;
LRAMDAT225    1;
LRAMDAT226    1;
LRAMDAT227    1;
LRAMDAT228    1;
LRAMDAT229    1;
LRAMDAT230    1;
LRAMDAT231    1;
LRAMDAT232    1;
LRAMDAT233    1;
LRAMDAT234    1;
LRAMDAT235    1;
LRAMDAT236    1;
LRAMDAT237    1;
LRAMDAT238    1;
LRAMDAT239    1;
                        % enter MDDATA 6

                        % exit MDDATA 5
BUSY09H       1;        % backplane busy
BUSY08H       1;        % backplane busy
SBPHOLD5H     1;        % source backplane hold
ERROR5H       1;        % error detected
SBE75         1;        % single bit error
LRAMECC2X     4;        % latched ram data ECC
LRAMDAT2X    20;        % latched ram data
                        % enter MDDATA 5

                        % exit MDDATA array 4
BUSY07H       1;        % backplane busy
BUSY06H       1;        % backplane busy
SBPHOLD4H     1;        % source backplane hold
ERROR4H       1;        % error ff
SBE4H         1;        % single bit error
LRAMECC14     1;        % latched ram data ECC
LRAMECC15     1;
LRAMECC16     1;
LRAMECC17     1;
```

APPENDIX A (Continued)

```
LRAMDAT120    1;        % latched ram data bus
LRAMDAT121    1;
LRAMDAT122    1;
LRAMDAT123    1;
LRAMDAT124    1;
LRAMDAT125    1;
LRAMDAT126    1;
LRAMDAT127    1;
LRAMDAT128    1;
LRAMDAT129    1;
LRAMDAT130    1;
LRAMDAT131    1;
LRAMDAT132    1;
LRAMDAT133    1;
LRAMDAT134    1;
LRAMDAT135    1;
LRAMDAT136    1;
LRAMDAT137    1;
LRAMDAT138    1;
LRAMDAT139    1;
                        % enter MDDATA 4

                        % exit MDDATA 3
BUSY05H       1;        % backplane busy
BUSY04H       1;        % backplane busy
SBPHOLD3H     1;        % source backplane hold
ERROR3H       1;        % error detected
SBE3H         1;        % single bit error
LRAMECC1X     4;        % latched ram data ECC
LRAMDAT1X    20;        % latched ram data
                        % enter MDDATA 3

                        % exit MDDATA array 2
BUSY03H       1;        % backplane busy
BUSY02H       1;        % backplane busy
SBPHOLD2H     1;        % source backplane hold
ERROR2H       1;        % error ff
SBE2H         1;        % single bit error
LRAMECC04     1;        % latched ram data ECC
LRAMECC05     1;
LRAMECC06     1;
LRAMECC07     1;
```

APPENDIX A (Continued)

```
LRAMDAT020    1;        % latched ram data bus
LRAMDAT021    1;
LRAMDAT022    1;
LRAMDAT023    1;
LRAMDAT024    1;
LRAMDAT025    1;
LRAMDAT026    1;
LRAMDAT027    1;
LRAMDAT028    1;
LRAMDAT029    1;
LRAMDAT030    1;
LRAMDAT031    1;
LRAMDAT032    1;
LRAMDAT033    1;
LRAMDAT034    1;
LRAMDAT035    1;
LRAMDAT036    1;
LRAMDAT037    1;
LRAMDAT038    1;
LRAMDAT039    1;
                        % enter MDDATA 2

                        % exit MDDATA 1
BUSY01H       1;        % backplane busy
BUSY00H       1;        % backplane busy
SBPHOLD1H     1;        % source backplane hold
ERROR1H       1;        % error detected
SBE1H         1;        % single bit error
LRAMECC0X     4;        % latched ram data ECC
LRAMDAT0X    20;        % latched ram data
                        % enter MDDATA 1

                        % exit MDCNTL array
LMEMBUSH      1;        % latched MEMBUS timing signal
BUSYCTRPH     1;        % busy timer parity predict
BUSYCTR       6;        % busy timeout counter ff's
MONLINEH      1;        % monitor of online ff
DBLBSYH       1;        % double cycle busy
LLOGADH       4;        % latched card logical address
LNMBRL        4;        % latched MDC count number
ECORRENH      1;        % error correct enable (data)
REPORTH       1;        % report MDC ID and error code
```

APPENDIX A (Continued)

```
        SOURCEH      1;        % source data bus
        MREQSOFFH    1;        % maint chain request off
        MREQSH       4;        % maint chain request state
                               % enter MDCNTL array

                               % enter board

    CHNEND;                    % End of chain



    REGLIST     YS;            % MDC Maintenance chain

        LRAMECC3   5;          % ECC for block 3

           LRAMECC37;
           LRAMECC36;
           LRAMECC35;
           LRAMECC34;
           LRAMECC3X#

        LRAMDAT3   21;         % data for block 3
           LRAMDAT339;
           LRAMDAT338;
           LRAMDAT337;
           LRAMDAT336;
           LRAMDAT335;
           LRAMDAT334;
           LRAMDAT333;
           LRAMDAT332;
           LRAMDAT331;
           LRAMDAT330;
           LRAMDAT329;
           LRAMDAT328;
           LRAMDAT327;
           LRAMDAT326;
           LRAMDAT325;
           LRAMDAT324;
           LRAMDAT323;
           LRAMDAT322;
```

APPENDIX A (Continued)

```
      LRAMDAT321;
      LRAMDAT320;
      LRAMDAT3X#

   LRAMECC2   5;          % ECC for block 2

      LRAMECC27;
      LRAMECC26;
      LRAMECC25;
      LRAMECC24;
      LRAMECC2X#

   LRAMDAT2   21;         % data for block 2
      LRAMDAT239;
      LRAMDAT238;
      LRAMDAT237;
      LRAMDAT236;
      LRAMDAT235;
      LRAMDAT234;
      LRAMDAT233;
      LRAMDAT232;
      LRAMDAT231;
      LRAMDAT230;
      LRAMDAT229;
      LRAMDAT228;
      LRAMDAT227;
      LRAMDAT226;
      LRAMDAT225;
      LRAMDAT224;
      LRAMDAT223;
      LRAMDAT222;
      LRAMDAT221;
      LRAMDAT220;
      LRAMDAT2X#

   LRAMECC1   5;          % ECC for block 1

      LRAMECC17;
      LRAMECC16;
      LRAMECC15;
      LRAMECC14;
      LRAMECC1X#
```

APPENDIX A (Continued)

```
LRAMDAT1   21;          % data for block 1
   LRAMDAT139;
   LRAMDAT138;
   LRAMDAT137;
   LRAMDAT136;
   LRAMDAT135;
   LRAMDAT134;
   LRAMDAT133;
   LRAMDAT132;
   LRAMDAT131;
   LRAMDAT130;
   LRAMDAT129;
   LRAMDAT128;
   LRAMDAT127;
   LRAMDAT126;
   LRAMDAT125;
   LRAMDAT124;
   LRAMDAT123;
   LRAMDAT122;
   LRAMDAT121;
   LRAMDAT120;
   LRAMDAT1X#

LRAMECC0   5;           % ECC for block 0

   LRAMECC07;
   LRAMECC06;
   LRAMECC05;
   LRAMECC04;
   LRAMECC0X#

LRAMDAT0   21;          % data for block 0
   LRAMDAT039;
   LRAMDAT038;
   LRAMDAT037;
   LRAMDAT036;
   LRAMDAT035;
   LRAMDAT034;
   LRAMDAT033;
   LRAMDAT032;
   LRAMDAT031;
```

APPENDIX A (Continued)

```
LRAMDAT030;
LRAMDAT029;
LRAMDAT028;
LRAMDAT027;
LRAMDAT026;
LRAMDAT025;
LRAMDAT024;
LRAMDAT023;
LRAMDAT022;
LRAMDAT021;
LRAMDAT020;
LRAMDAT0X#

REGEND;                                  %End of register list
                                         %End of file
```

V500 MDC DATA (TIMING) CHAIN

```
DATE     ]102786;

CHAIN     S                  % Data (timing) chain
                             % exit board

                             % exit CLKMNT2 array
                             % enter CLKMNT2 array

                             % exit discrete counter
     RICNTRH     8;          % refresh interval counter
                             % enter discrete counter

                             % exit control array
     FRCREFH     1;          % clocked force refresh signal
     FRCSTRBH    1;          % force refresh strobe
     RICHOLDL    1;          % refresh interval counter hold count
     RICCHKBH    1;          % refresh interval counter check bit
     RCNTRPH     1;          % refresh address counter parity bit
     RCNTRH     10;          % refresh address counter
     REFRASEH    1;          % refresh RAS cycle end
     REFCYCH     4;          % refresh cycle state
     RWAITH      5;          % refresh wait state
     REFBUSYH    1;          % refresh busy
     ONLINEH     1;          % online control ff
     MODBRKH     1;          % module fail
     MNTMODH     1;          % maintenance mode
     DREQSH      5;          % data chain request state
                             % enter control array

                             % enter board


CHNEND;                      %End of chain
```