



1. Scope	
1.1. Overview .....	2
1.2. Related Documents .....	2
2. Terminology.....	3
3. Module Description	
3.1. MBUS Receivers/Transceivers/Drivers .....	4
3.1.1 M-bus Signals .....	4
3.1.1.1. Bus Arbitration .....	5
3.1.1.1.1. MBRQ .....	5
3.1.1.2. Data Transfer .....	5
3.1.1.2.1. MCMD .....	6
3.1.1.2.2. MSTATUS .....	6
3.1.1.2.3. MDAL .....	7
3.1.1.2.4. MPARITY .....	7
3.1.1.2.5. MSHARED .....	7
3.1.1.2.6. MDATINV .....	7
3.1.1.3. Control Signals .....	7
3.1.1.3.1. MID .....	7
3.1.1.3.2. MRESET .....	8
3.1.1.3.3. MDCOK .....	8
3.1.1.3.4. MIRQ .....	8
3.1.1.3.5. MABORT .....	8
3.1.1.4. Clocks .....	8
3.1.1.4.1. MCLKA .....	8
3.1.1.4.2. MCLKB .....	8
3.1.1.4.3. M-bus MCLKA/MCLKB Waveforms .....	9
3.2. DRAM Array .....	9
3.3. FMCM (Firestarter Memory Controller Module) .....	10
3.3.1. ADDRESS Path .....	10
3.3.2. DATA Path .....	10
3.3.2.1. Write Data Buffer .....	10
3.3.2.2. Read Data Buffer .....	11
3.3.3. Memory M-bus Interface .....	11
3.3.3.1 Memory Space Read (unshared) .....	11
3.3.3.2 Memory Space Read (shared) .....	12
3.3.3.3 Memory Space Write (unshared) .....	12
3.3.3.4 Memory Space Write (shared) .....	13
3.3.3.5 Interlocked Transactions .....	13
3.3.3.6 I/O Read Transaction .....	13
3.3.3.7 I/O Write Transaction .....	13
3.3.3.8 Un-owned Transactions .....	13
3.3.3.9 No Slave Response .....	13
3.3.4. DRAM Controller .....	13

3.3.5. Error Strategy ..... 14  
3.3.5.1 M-bus Errors ..... 14  
3.3.5.2 Internal Memory Errors ..... 15

4. Register Description

4.1. FMCM Registers ..... 16  
4.2. FMCM Register Map ..... 16  
4.3. FMCM Register Summary ..... 16  
4.3.1. MODTYPE Register..... 16  
4.3.2. BUSCSR Register..... 17  
4.3.3. FMDCSR Register..... 19  
4.3.4. BASEADDR Register..... 20

5. Power Requirements

5.1 Module Power Requirements ..... 21

## 1. Scope

This specification is intended to provide all the details necessary to utilize the Firestarter 8 MByte Memory Module as the prototype memory during the early stages of Firefox system debug.

### 1.1. Overview

The Firestarter 8M Byte Memory Module is a main memory which responds to the MBUS protocol and retains/retrieves data to be used by the Firefox processors and I/O subsystem. This module's storage capacity is 8 megabytes of DRAM memory which may be accessed via the M-bus protocol specified in the "Firefox M-Bus Specification". This module performs all of the translations necessary to interface the MBUS to the DRAMs from a timing and protocol standpoint during normal operation. This module DOES NOT have battery backup capability and all data will be lost in the event of a power failure.

This module DOES NOT have the capability of initializing its DRAM array on command from a processor.

### 1.2. Related Documents

The following documents contain material that is referenced:

- o Firefox System Specification
- o Firefox M-Bus Specification
- o Firefox Bus Interface Chip Specification
- o Dynamic MOS RAM, +5V, 256K (262144 X 1) Specification
- o 2 Megabyte daughter board specification (54-16500-BA)

## 2. Terminology

CAS	Column Address Strobe. This term is used to describe a control signal sent to a DRAM. One function of this signal is to indicate that the DRAM should interpret the signals on its address pins as the second dimension of its two dimensional addressing scheme and latch them internally. Another function is to serve as an "output enable" which controls the driving of the DRAM's Data Out pin during a READ operation. This signal, with RAS and WE, controls the operation of the DRAM.
Column Address, or Column	This term is used to describe the second half of the two dimensional, time multiplexed addressing scheme used to access data in most DRAMs. The addresses are present on the DRAM address input pins and their interpretation is controlled by CAS. The first half of the address was previously presented as the Row Address.
DRAM	Dynamic Random Access Memory. A form of Read/Write memory whose contents must be periodically refreshed so that data will be maintained.
MBZ	Must Be Zero. This term is used to indicate that a field of an I/O register must be driven as all "0's". This mechanism is used to reserve register bits for future enhancements and make new versions of a product backward compatible.
RAS	Row Address Strobe. This term is used to describe a control signal sent to a DRAM. One function of this signal is to indicate that the DRAM should interpret the signals on its address pins as one dimension of its two dimensional addressing scheme and latch them internally. Another function is to serve as an flag to start an internal set of timing sequences that lead to a memory access. This signal, with CAS and WE, controls the operation of the DRAM.
Refresh	This term is used to describe an operation that must be performed periodically on a DRAM so that it will retain the data stored in it. In the Firestarter memory system, the FMCM performs this function and makes it nearly invisible to the MBUS.
Row Address, or Row	This term is used to describe the first half of the two dimensional, time multiplexed addressing scheme used to access data in most DRAMs. The addresses are present on the DRAM address input pins and are latched by RAS. The second half of the address is presented as the Column Address.
WE	Write Enable. This signal, when asserted and in conjunction with RAS and CAS, signals the DRAM to store data from its Data IN pin into its internal array.

3.1 M-bus Receivers/Transceivers/Drivers

Table 3-1 lists the transceiver/driver components that are utilized on the Memory M-bus interface. All transceiver/driver components are in DIP packages. Specified termination will be connected in series immediately after the bus driver transceiver output (bus side of transceivers). All specified pullups are implemented on the backplane.

Component	Termination	Pullup	Signal(s)
74F244	20 ohm	4.7K ohm	MBRQ
74F245	20 ohm	4.7K ohm	MDAL<31:24>
74F245	20 ohm	4.7K ohm	MDAL<23:16>
74F245	20 ohm	4.7K ohm	MDAL<15:08>
74F245	20 ohm	4.7K ohm	MDAL<07:00>
74F245	20 ohm	4.7K ohm	MDPARITY
74F245	20 ohm	4.7K ohm	MCMD<3:0>, MCPARITY
74F245	20 ohm	4.7K ohm	MSTATUS<1:0>, MSPARITY
74AS760		143/768 ohm	MSHARED, MBUSY, MABORT

TABLE 3-1 M-bus TRANSCEIVER/DRIVER COMPONENTS

Table 3-2 lists the per memory module loading for each M-bus signal. Loading is in addition to the output driver, if appropriate. For example, the MABORT signal has 1 74AS760 output driver and 1 74F244 input load per memory array module.

Loading	Signal(s)
1 74F244 RECEIVER	MBRM<6:0>, MBUSY
1 74F245 TRANSCEIVER	MDAL, MDPARITY
1 74F245 TRANSCEIVER	MCMD, MCPARITY
1 74F245 TRANSCEIVER	MSTATUS, MSPARITY
1 74F244 RECEIVER	MSHARED, MDATINV, MABORT, MRESET
4 74F244 RECEIVER	MCLKA, MCLKB
4 74F00 GATE	MCLKA, MCLKB
1 74F240 RECEIVER	MCLKA, MCLKB

Table 3-2 Memory Array Module Input Loading

3.1.1. M-bus Signals

Since the memory interfaces directly to the M-bus, the signal list and functions are defined in the Firefox M-bus Specification. The signal descriptions contained herein are as they relate to the memory specifically.

## 3.1.1.1. Bus Arbitration

In contrast to the operation of the CPU, the memory does not, strictly speaking, arbitrate for the M-bus. The reasons for this are:

- a) The memory is never a bus master and therefore does not need to arbitrate for mastership of the M-bus.
- b) During shared transactions, the memory knows one cycle in advance that it should not assert its MBRQ.
- c) There should never be a case for more than one memory module being a responder for a given M-bus address. The only exceptions are in the case of a slot MID failure or memory resource allocation error (ie. The BASEADDR registers of two or more memory modules were programmed such that their address spaces overlap). The latter case would signal a MABORT during P4 or P7 because of a multiple MBRQ error.

For the above reasons, the memory is not required to check the priority level of other requesters. It is required to assure that there is one and only one requestor on the bus during non-arbitration cycles.

## 3.1.1.1.1. MBRQ

The memory drives one of the eight MBRQ signals to show that it is the one responding to an M-bus transaction. The MBRQ that is driven is a function of the M-bus slot where that module is located. The module in slot #0 will drive MBRQ<0>, etc.

The memory module which is selected by the transaction address, either I/O space address or memory space address, asserts its MBRQ signal for as long as it drives the bus. In the case of shared memory space transactions, the memory will forfeit driving the M-bus to the highest priority cache.

Although the memory does not arbitrate for the bus, it must monitor all the other seven MBRQ signals. The reasons for this are:

- a) It must determine if it is the bus slave
- b) It must keep in synchronization with the rest of the system
- c) It must check for multiple MBRQs during a non-arbitration cycles.

The memory will assert its MBRQ during the time that the system is asserting MRESET. The system will use this feature to determine which slots are occupied. The MBRQ assertion and deassertion will be one cycle delayed from MRESET because of the pipelined nature of the bus protocol.

## 3.1.1.2. Data Transfer

Data transfer to and from the memory is accomplished via the 32 bit bidirectional MDAL bus under control of the MCMD and MSTATUS signals. The MXPARITY signals enable single bit error detection of the MCMD, MSTATUS, and MDAL signals.

The MDAL signals are driven by the bus master to specify address and write data. The MDAL signals are driven by the memory to specify read data. The MCMD signals are driven by bus masters to specify the transaction type and I/O space byte masks. The MSTATUS signals are driven by the memory to indicate status of the current transaction.

## 3.1.1.2.1. MCMD

The MCMD lines, which are driven by the bus master, serve two different purposes during various transaction times: first, they indicate transaction type during P2 of all M-bus transactions; second, they indicate I/O read/write byte mask during P3 of I/O space transactions.

The following table lists the encoding of the MCMD lines during cycle P2 of a M-bus transaction:

Value	Mnemonic	Function
0101	READ	I/O read or memory space read request
0111	WRITET	Memory space write through request
1001	READI	Interlocked memory space read request
1010	WRITE	I/O write or memory space write request
1011	WRITEU	Memory space write unlock request
* 1101	INTACK	Interrupt acknowledge request
1110	READU	Memory space unshared read request

\* The INTACK transaction is only monitored by the memory for proper bus protocol.

Note that a memory space versus an I/O space transaction is specified by MDAL<31> of a read or write address.

For an I/O space read or write transactions, the MCMD lines specify byte mask for the longword being accessed. MCMD<3:0> correspond to longword bytes <3:0>. If MCMD<n> is asserted, then the corresponding byte of the longword contains valid data. If MCMD<n> is deasserted, then the corresponding byte of the longword contains undefined data. Although some of the bytes in a longword may be undefined, all 32 lines plus the parity line must be driven. The parity will always be calculated across the entire longword.

## 3.1.1.2.2. MSTATUS

The MSTATUS lines are asserted by the selected memory during I/O transactions and unshared memory space read transactions. The memory must actively assert WAIT status to stall a transaction until it is ready with the read data. The following table describes the values and functions of the MSTATUS lines.

Value	Mnemonic	Function
00	WAIT	Stall transaction
01	GOOD	I/O write complete/good read data
10	CORRECTED	(this status unused by this memory)
11	ERROR	(this status unused by this memory)

3.1.1.2.3. MDAL

The MDAL are bidirectional lines which are used by the memory to receive I/O address, memory space address, and write data. They are also used by the memory to return the requested I/O or memory read data. In addition, they indicate interrupt level and vector for modules which perform interrupt acknowledge transactions. Note that although the memory does not perform interrupt transactions, it does check parity on the MDAL during the INTACK.

3.1.1.2.4. M\*PARITY

The MCPARITY signal specifies even parity for the MCMD signals. The MSPARITY specifies even parity for the MSTATUS signals. The MDPARITY signal specifies even parity for the MDAL signals. When a bus master is driving the MCMD ,MSTATUS or the MDAL signals the memory will check those fields for correct parity. When the memory is driving the MSTATUS or the MDAL signals it will drive the entire field and the corresponding parity. See section 3.3.5.1 for a detailed explanation of when the various fields are checked for valid parity.

3.1.1.2.5. MSHARED

The MSHARED signal is used by the caches to maintain data integrity throughout the system. The Firestarter memory module does not utilize this signal.

3.1.1.2.6. MDATINV

Whenever a module drives data onto the MDAL that is known to contain errors, it asserts MDATINV. This memory DOES NOT use this signal to modify the data that it stores in its DRAM array. Furthermore, if bad data is read out of the DRAM array, this module will not assert MDATINV. It is likely that a single bit error (either soft or hard) in a DRAM will cause bad parity to be detected on the M-bus when read data is presented during a memory read transaction.

3.1.1.3. Control Signals

The MID, MRESET, MDCOK, MIRQ, and MABORT signals are used to initialize and coordinate activity on the M-bus.

3.1.1.3.1. MID

The MID signals uniquely identify each backplane slot with a value from 0 to 7. The table below lists the connections for each slot. The MID value is used by the address decode logic in the memory to determine which I/O addresses the memory will respond to. In this way, no jumpers or switches will be required for configuring a module.

Slot	MID<2>	MID<1>	MID<0>
0	Gnd	Gnd	Gnd
1	Gnd	Gnd	+5V
2	Gnd	+5V	Gnd
3	Gnd	+5V	+5V
4	+5V	Gnd	Gnd
5	+5V	Gnd	+5V
6	+5V	+5V	Gnd
7	+5V	+5V	+5V

## 3.1.1.3.2. MRESET

The MRESET signal will be used to set the memory control logic and selected I/O register fields to a known state. When the MRESET signal is asserted, the memory will terminate any current transaction. Consult section 4.0 to determine the effects of MRESET on the FIRESTARTER I/O register fields.

One effect of MRESET is that the memory address space of the module will be made inaccessible. The memory address space of each memory module will remain inaccessible until the BASEADDR register has been loaded to specify the starting address of a module's memory address space and the MEMSPEN bit of that register has been asserted.

## 3.1.1.3.3. MDCOK

Since the memory is not equipped with a battery backup feature, it will not respond or react in any way to the MDCOK signal. This signal will not be received on a finger pin of this module.

## 3.1.1.3.4. MIRQ

Since the memory does not utilize the interrupt function, it will never assert or be required to react in any way to the MIRQ signal. These signals will not be received on the finger pins of this module.

## 3.1.1.3.5. MABORT

The MABORT signal will be asserted by the memory if it detects an error condition during any M-bus transaction. The possible transaction errors detected by this module are:

- a) parity errors on MDAL, MCMD, or MSTATUS
- b) reserved values on the MCMD during P2

## 3.1.1.4. Clocks

The MCLKA and MCLKB master clocks are used by the memory to control all of its timing. The MCLKI signal functions as an interval timer for the system and is not utilized by the memory. The MCLKI signal will not be received on a finger pin of this module.

## 3.1.1.4.1. MCLKA

MCLKA is the master clock for the M-bus. All M-bus signal transitions and bus interface state machines are referenced to MCLKA. The M-bus cycles, P<sub>n</sub>, are defined by the rising edge of MCLKA, i.e., MCLKA is used by the memory to clock registers and enable latches driving the M-bus. MCLKA is radially distributed to each module in the backplane in order to minimize skew between modules.

## 3.1.1.4.2. MCLKB

MCLKB is the slave clock for the M-bus. M-bus receiver latches and registers in the memory will be clocked by MCLKB. MCLKB is radially distributed to each module slot in order to minimize skew between modules.

3.1.1.4.3 M-bus MCLKA/MCLKB Waveforms

Figure 3-1 shows the waveforms expected by the memory subsystem. The clock generator is based on a divide-by-6 circuit of the master oscillator. The clock generator is free-running and self-initializing from an arbitrary power-up state within four oscillator cycles. The memory receives only the MCLKA and MCLKB signals. The OSC trace is included only to show the six phase nature of the clocking system. The memory worst case design will assume a minimum cycle period of 70nS (MCLKA rising to MCLKA rising). The maximum clock cycle period will be 120nS. The memory will operate at cycle periods greater than 120nS but the refresh period of the DRAMs will rise to a value greater than specifications allow. As long as the occurrence of DRAM data loss is taken into account, the memory may be operated at cycle periods greater than 120nS.

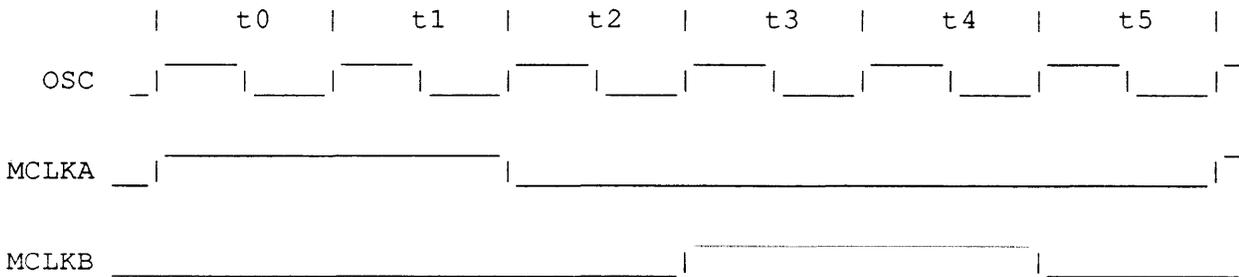


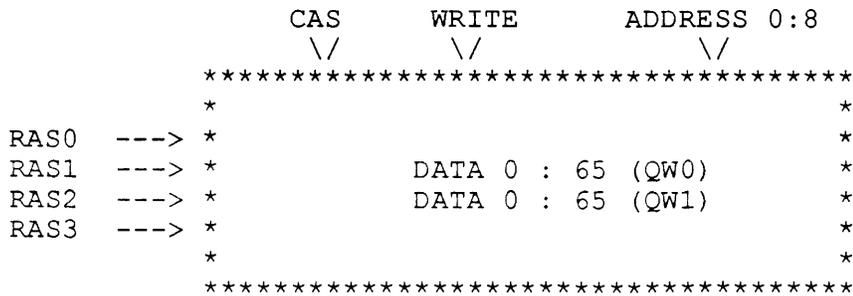
Figure 3-1

3.2. DRAM ARRAY

The Firestarter memory module uses four - 2 megabyte daughter boards. Each board consists of 78, 256Kx1 DRAMs, of which only 66 per daughter board are used (264 total). The module uses 4 bank selects to read/write the memory. This does not mean that each daughter board stores one longword. Data is transferred to and from the memory in two quadwords. Each quadword is distributed among all four daughter boards. The four bank selects, are connected to each of the daughter boards.

8 MB MODULES

DRAM ARRAY ORGANIZATION



All memory chips used in our 8MB designs will be page mode 256K X 1 DRAMS. These DRAM arrays are double sided surface mounted PC boards.

### 3.3. FMCM (Firestarter Memory Controller Module)

The following sections describe the characteristics of component pieces comprising the Firestarter memory system.

#### 3.3.1 ADDRESS PATH

The memory module's addressing consists of row and column DRAM addresses as well as DRAM bank select.

Address Signals Assignments :

-----

MDAL <12:04> = Row Addresses

MDAL <20:13> = Column Addresses

MDAL <22:21> = Bank Select (normalized using BASEADDR)

MDAL <30:23> = Module Offset Address \*

MDAL <31> = Indicates I/O Space Address transaction

\* Offset Address lines are used by the array (with the BASEADDR register), to determine whether or not to respond to the current M-bus transaction. DRAM addresses in this range are normalized to zero before they are used.

#### 3.3.2 Data Path

The Data Path section of the Memory Array module does the following:

- o Carries write data from the M-bus to the DRAMS
- o Carries read data from the DRAMS to the M-bus
- o Stores a line of write data, in the case of a refresh in progress, until the DRAMS are ready
- o Maintains data integrity between the M-bus and the DRAMS

##### 3.3.2.1 Write Data Buffer

As a result of a WRITE,WRITET or WRITEU, write data is loaded from the MDAL lines into the write data buffers. Parity is checked on each 32 bit "long word" as it is received from the bus. The data is then assembled into quadwords and is presented to the DRAM Array one quadword at a time. The two 66 bit words (64 data lines plus 2 parity bits) are held in the data path, in the event of an in progress RAM refresh cycle, until the REFRESH cycle has completed.

### 3.3.2.2 Read Data Buffer

As a result of a READ, READI or READU, an octaword of read data is taken from the dynamic RAMS by means of a double page mode read cycle. The two 66 bit words are taken out of the DRAMs, one at a time. Each of the quadwords will then be divided into 32 bit longwords with parity. To complete the READ transaction, the four 33 bit words (data + parity) are sent to the M-bus along with "good" status to reflect no error.

### 3.3.3 Memory M-bus Interface

Since the memory array interfaces directly to the Firefox M-bus, (it doesn't utilize a private memory bus) it must respond to M-bus transactions and adhere to the M-bus specification and protocol. The memory M-bus interface does differ from some of the other M-bus modules in that it never takes the role of bus master. This implies that the memory can only respond to bus transactions or monitor the bus. The purpose of monitoring the M-bus, in the case of "Un-owned" transactions, is to stay in synchronization with bus transactions, to aid in checking bus protocol and bus information integrity.

The Memory M-bus interface responds to the following transaction types:

- o READ (memory space read or I/O read)
- o READI (read interlocked)
- o READU (read unshared)
- o WRITE (memory space write or I/O write)
- o WRITET (write through)
- o WRITEU (write unlock)

The memory will never initiate a INTACK (interrupt acknowledge) command. It will monitor an INTACK in order to check bus protocol and bus parity.

#### 3.3.3.1 Memory Space Read (unshared)

During P2 of an M-bus transaction, with MDAL bit 31 unasserted, the memory will interpret a READ command as a memory space read. The memory M-bus interface will then decode the upper address bits to determine, based on the base address register and the memory size, if the address falls within its domain. If the address is "owned," the M-bus interface will send MY\_CYCLE and READ to the DRAM controller section during P3 to signal an octaword read request. During RP4-RP6 the M-bus interface will monitor the bus. If the DRAM state machine is caught up with the M-bus state machine (i.e. the DRAM s.m. hasn't fallen behind due a Refresh cycle), then it will send DATA\_RDY to the MBUS state machine. During P7, the M-bus interface will assert its MBRQ and read status on the bus. During P7 through P10, four consecutive long words get driven onto the MDAL lines. At the end of P9, the M-bus interface will remove its MBRQ.

If there had been a refresh in progress at the time of the octaword request, the Data\_Rdy signal would be delayed until a later cycle allowing the Refresh cycle to complete. In this case, WAIT status and MBRQ will be asserted during P7 and during each of the following cycles until the DRAM state machine asserts Data\_Rdy. In the four cycles following the assertion of Data\_Rdy, read status, read data, and MBRQ will be asserted.

The maximum number of excess WAIT cycles that can be generated as a result of an Refresh cycle during a memory read is five.

### 3.3.3.2 Memory Space Read (shared)

During P2 of an M-bus transaction, with MDAL bit 31 unasserted, the memory will interpret a READ command as a memory space read. The memory M-bus interface will then decode the upper address bits to determine, based on the base address register and the memory size, if the address falls within its domain. If the address is "owned," the M-bus interface will send MY\_CYCLE and READ to the DRAM controller section during P3 to signal an octaword read request. During P4-P6, the M-bus interface will monitor the bus. If the DRAM state machine is caught up with the MBUS state machine (i.e. the DRAM s.m. hasn't fallen behind due a Refresh cycle), then it will send DATA\_RDY to the M-bus state machine.

In the case of a shared cycle, the M-bus interface will see a MBRQ during P6. The MBRQ indicates that one of the caches will supply read data for this transaction. The memory's M-bus interface will not drive the M-bus but will allow the requesting cache to drive the read data, status, and MBRQ. Also, in the case of a shared read, the M-bus interface will monitor the M-bus/CACHE transaction, rather than the DRAM controller response, to determine the end of the transaction.

### 3.3.3.3 Memory Space Write (unshared)

During P2 of an M-bus transaction, with MDAL bit 31 unasserted, the memory will interpret a WRITE, WRITET or WRITEU command as a memory space write. The memory M-bus interface will then latch the address and decode the upper bits to determine, based on the base address register and the memory size, if the address falls within its domain. If the address is "owned," the M-bus interface will send MY\_CYCLE and WRITE (during P3) to the DRAM controller section of the FMCM to signal an octaword write request. During cycles P3 through P6, LW0-LW3 will flow through the data path section of the FMCM. If the DRAM state machine is not backed up due to a refresh, the M-bus interface will receive WRITE\_DONE from the DRAM controller in cycle P3.

If there was a refresh in progress at the time of the octaword request, the WRITE\_DONE signal would be delayed until a later cycle waiting for completion of the refresh. In this case, depending on how many cycles WRITE\_DONE has been delayed, the MBUSY signal will be asserted to delay the BUS arbitration for the next transaction until the current WRITE transaction can be serviced.

### 3.3.3.4 Memory Space Write (shared)

The shared memory space write is handled in the same way that an unshared memory space write is handled.

### 3.3.3.5 Interlocked Transactions

Interlocked READ and WRITE transactions are treated, by the memory, in exactly the same way as normal READ and WRITE transactions.

### 3.3.3.6 I/O Read Transaction

During P2 of an M-bus transaction, with MDAL bit 31 asserted, the memory will interpret a READ command as an I/O space read. During P3, the memory M-bus interface will decode the address bits to determine, based on the MID backplane slot code, if the address falls within its 32 Mbyte domain. If the address is "owned," the M-bus interface will prepare to drive the M-bus starting in P4 with its MBRQ, MSTATUS, and I/O read data on the MDAL lines.

### 3.3.3.7 I/O Write Transaction

During P2 of an M-bus transaction, with MDAL bit 31 asserted, the memory will interpret a WRITE command as an I/O space write. During P3, the memory M-bus interface will decode the address bits to determine, based on the MID backplane slot code, if the address falls within its 32 Mbyte domain. Also, during P3, the M-bus interface will strobe data from the MDAL and use the MCMD lines (mask) to determine which bytes are to be accessed. If the address is "owned," the M-bus interface will prepare to drive the M-bus starting in P4 with its MBRQ, and the MSTATUS with wait or write status.

### 3.3.3.8 Un-owned Transactions

The Memory M-bus interface will monitor all Un-owned transactions for bus parity, correct protocol, and to keep synchronized with the rest of the system.

### 3.3.3.9 No Slave Response

Lack of slave response during valid transactions will be detected by the memory interface during P4 of I/O space transactions, P4 of interrupt acknowledge transactions, and P7 of memory space read transactions. The lack of an MBRQ on the bus during any of the above cycles will cause the memory M-bus interface to recognize that the transaction has terminated and force it immediately to its idle state.

## 3.3.4 DRAM Controller

READ and WRITE cycles will consist of octaword data transfers which operate DRAMs using page mode timing protocol. Sixty-six bits of data (quadword + 2 Parity Bits) are written to or read from the DRAMs and transferred on the M-bus as four longwords plus parity, one cycle apart. Refresh is performed using RAS only timing protocol.

READ and WRITE cycles operate using addresses that are asserted on the M-bus during the P2 timing state. These addresses are latched for the duration of the current transaction. DRAM READ/WRITE timing states will begin during P3 and all DRAM timing is clocked off of MCLKA.

Timing state C0 is an idle state in which the DRAM state machine determines what task to perform next : READ, WRITE, REFRESH, or remain idle. During C0, the DRAM control will remain idle until read or write request is received from the M-bus state machine or a refresh command is received from the refresh control logic.

The DRAM state machine attaches two levels of priority to cycle requests. The highest priority cycle request is a REFRESH command, followed by a DRAM access request for a READ or WRITE. The refresh state machine issues a REFRESH command when DRAMs have not been refreshed for 14 microseconds. Software provisions have been made to optimize the refresh time for different clock speed (see section 4.3.3 on the FMDCSR register).

During a READ cycle, the DRAM Column Addresses will be selected for QW0 and QW1 by the signals Column\_Addr\_Sel and Sel\_QW1 lines. QW1 will be accessed by asserting the Sel\_QW1 signal.

During a WRITE cycle, column address selects are handled the same as a Read cycle. When QW0 and QW1 get written into the DRAMs. WRITE\_DONE will be asserted and sent to the M-bus state machine.

The DRAMs used in the Firestarter Memory module, require a RAS only Refresh operation. During refresh cycles, internally generated Refresh addresses are presented to the DRAMs and all the RAS lines will be asserted. A REFR\_DONE signal gets asserted during the Refresh cycle which increments the Refresh Address and resets the M-bus clock tick counter.

### 3.3.5. Error Strategy

The following error strategy will be implemented with regard to M-bus errors or internal memory module errors.

#### 3.3.5.1 M-bus Errors

The M-bus class of errors are either bus parity errors or bus protocol errors. These errors are detected by any of the M-bus interfaces. Any M-bus error detected by the memory will result in the assertion of the MABORT signal for eight cycles (if ISOLATE was not previously set).

When MABORT is asserted, the memory will abort any M-bus generated READ or WRITE cycle that has not yet begun. Any in progress DRAM cycle will be completed. Refresh cycles are unaffected by an MABORT.

The Memory M-bus interface will check parity on the MCMD, MSTATUS, and MDAL lines at appropriate times during a transaction. When the FMCM detects a bus parity error, it asserts MABORT. The following table, taken from the M-bus specification, illustrates the proper times to check each type of parity for non-wait transactions. The letters "C", "S", or "D" are meant to represent the three parity types. A "\*" in the table represents a state during which the MDAL parity should be checked per the M-bus specification but where the Firestarter memory module is unable to do so.

Transaction	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Memory Read		CD					S*	S*	S*	S*
Memory Write		CD	CD	CD	CD	CD				
I/O Read		CD	C	S*						
I/O Write		CD	CD	S						
Interrupt Ack		CD			SD					

For wait transactions, MSTATUS parity and MDAL parity as shown in the table will extend out into later cycles. MDAL parity errors will be ignored during all cycles during which the MSTATUS lines indicate "WAIT".

Multiple MBRQ signals on the bus during non-arbitration cycles will result in an MABORT. The FIRESTARTER memory DOES NOT check for this class of error.

### 3.3.5.2 Internal Memory Errors

There is no error detection of internal memory failures. Detection of this class of error is accomplished via one of the following methods:

- a. A "Slave timeout" for a nonresponding DRAM state machine (this would be caused by too many MBUSYs or MWAITS).
- b. Data parity errors for data bus and DRAM "stuck at" or "open" failures.

If the memory interface does not receive DATA\_RDY (READ cycle) or WRITE\_DONE (WRITE cycle) from the DRAM state machine, the M-bus interfaces asserts "WAIT" (during a READ) or the MBUSY (during a WRITE) on the MSTATUS lines at the appropriate time. This will force a SLAVE\_TIMEOUT error to be detected by one or more of the other M-bus interfaces after the WAIT/MBUSY threshold has been reached.

## 4. Register Description

The following sections describe the I/O registers implemented on the Firestarter Memory Controller Module (FMCM).

### 4.1. FMCM Registers

The FMCM supports a total of 4 internal registers. These registers implement the following:

- o M-bus module type identification
- o M-bus error detection and status
- o Control and status of the FMCM
- o Memory space base address offset

### 4.2. FMCM Register Map

Table 4-1 lists the FMCM registers and their function. The address field of the table shows the low order 24 bits of M-bus address necessary to access a register. The X address bits are related to the M-bus slot position in which the memory module resides and are described in Table 4-2. To determine the M-bus address necessary to access a memory module's I/O register, concatenate the first two address digits from Table 4-2 (slot address) with the last six digits from Table 4-1.

Table 4-1: FMCM Register Map

FMCM Register Offsets			
Name	Address	R/W	Description
MODTYPE	XXFFFFFFC#16	R	Module type register
BUSCSR	XXFFFFFF8#16	R/W	M-bus error status register
FMDCSR	XXFFFFFF4#16	R/W	FMCM control/status register
BASEADDR	XXFFFFFF0#16	R/W	Memory space base address register



4.3.2. BUSCSR Register

The BUSCSR register is a read/write register that maintains M-bus error status, and controls M-bus error logging. Figure 4-2 shows the bit definitions for the BUSCSR register. All bits of the BUSCSR are active low. The FMCM logs errors in the BUSCSR whenever the FROZEN bit is not asserted by asserting the corresponding status bit. Whenever the FMCM asserts a status bit, it also asserts the FROZEN bit. When the FROZEN bit is asserted, the FMCM does not alter the value of the BUSCSR. That is, the BUSCSR saves the first error event. If simultaneous errors occur, and error logging is enabled, the FMCM asserts multiple status bits. To enable error logging, write FFFFFFFF#16 to the BUSCSR. The result of simultaneous I/O space writes to BUSCSR and error events is unpredictable.

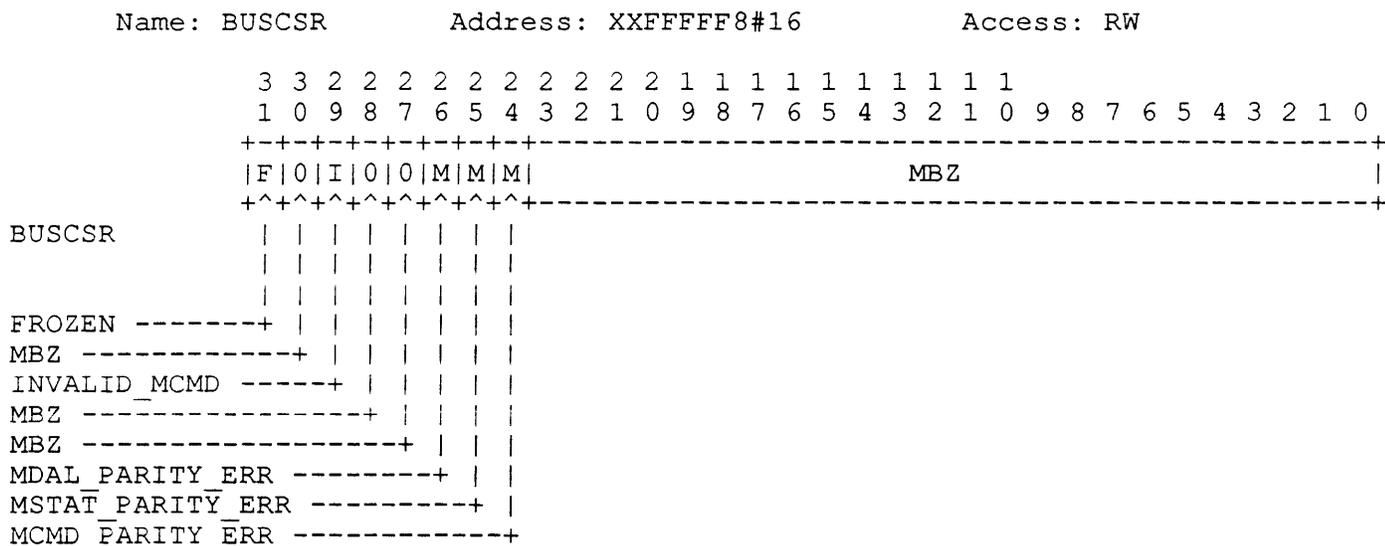


Figure 4-2: BUSCSR Register

\*\*\*\*\*  
NOTE: Asserting a bit in this register means that it will be read as a 0.  
Clearing a bit in this register means that it will be read as a 1.  
\*\*\*\*\*

BUSCSR<23:0> MBZ - always read as 0.

BUSCSR<24>MCMD\_PARITY\_ERR(RW) M-bus MCMD Parity Error

MCMD\_PARITY\_ERR is asserted when a parity error occurs on the MCMD signals. The FMCM checks MCMD parity the cycle after the value appears on the M-bus, that is: P3; memory write P4, P5, P6, first P7; I/O read first P4; and I/O write first P4. The FROZEN bit is asserted along with the MCMD\_PARITY\_ERR bit. The FMCM generates an MABORT sequence when it asserts MCMD\_PARITY\_ERR if the FMDCSR\_ISOLATE bit is not set.

BUSCSR<25>MSTAT\_PARITY\_ERR(RW) M-bus MSTATUS Parity Error

MSTAT\_PARITY\_ERR is asserted when a parity error occurs on the MSTATUS signals. The FMCM checks MSTATUS parity the cycle after the value appears on the M-bus, that is: memory read P8, P9, P10, P11; I/O read P5; I/O write P5; and interrupt acknowledge P6. The FROZEN bit is asserted along with the MSTAT\_PARITY\_ERR bit. The FMCM generates an MABORT sequence when it asserts MSTAT\_PARITY\_ERR if the FMDCSR\_ISOLATE bit is not set.

BUSCSR<26>MDAL\_PARITY\_ERR(RW) M-bus MDAL Parity Error

MDAL\_PARITY\_ERR is asserted when a parity error occurs on the MDAL. The FMCM checks MDAL parity the cycle after the value appears on the M-bus, that is: P3; memory write P4, P5, P6, first P7; I/O write first P4; and interrupt acknowledge P6. The FROZEN bit is asserted along with the MDAL\_PARITY\_ERR bit. The FMCM generates an MABORT sequence when it asserts MDAL\_PARITY\_ERR if the FMDCSR\_ISOLATE bit is not set.

BUSCSR<28:27> MBZ - always read as a 0.

BUSCSR<29>INVALID\_MCMD(RW) M-bus Invalid MCMD Error

INVALID\_MCMD is asserted when the memory detects an invalid MCMD encoding during P2. The FROZEN bit is asserted along with the INVALID\_MCMD bit. The FMCM generates an MABORT sequence when it asserts INVALID\_MCMD if the FMDCSR\_ISOLATE bit is not set.

BUSCSR<30> MBZ - always read as 0.

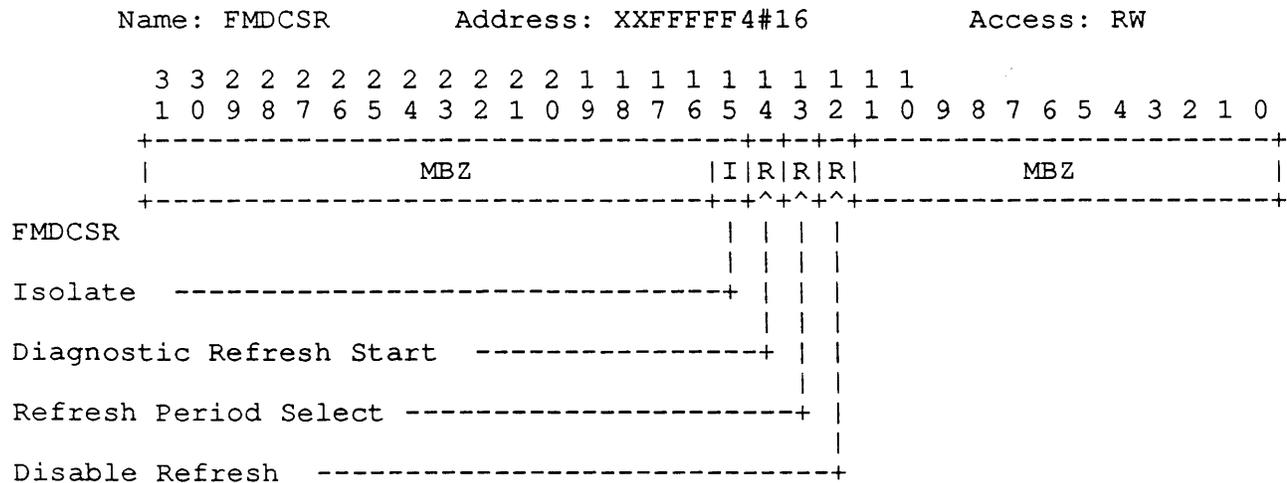
BUSCSR<31>FROZEN(RW) BUSCSR Latch Is Frozen

When the memory detects an M-bus error, or an MABORT on the bus, the FROZEN bit is asserted and the BUSCSR register is frozen. FROZEN bit may be used as an error summary bit for M-Bus errors. System reset asserts FROZEN.

#### 4.3.3. FMDCSR Register

The FMDCSR Register is an 32-bit read/write register that holds control and status information for the memory module. Only 4 of the 32 bits are used on the the Firestarter Module. The bits and sub-fields of this register are defined in figure 4-6 below :

Figure 4-6: FMDCSR Register



FMDCSR<12>DISABLE\_REFRESH (RW)

When set, the disable refresh bit disables the automatic refresh features of the memory module. There is no guarantee that data stored in the array will be preserved while bit 12 is asserted unless all rows of the DRAMs are refreshed (ie: read or rewritten) every 4 milliseconds. To avoid this, each DRAM row should be refreshed, on average, every 15 microseconds because there are 256 rows in each of the 256K X 1 chips.

FMDCSR<13>REFRESH\_PERIOD\_SELECT (RW)

The REFRESH\_PERIOD\_SELECT bit allows the the refresh control logic to optimize its refresh counter to different clock periods planned for the MBUS. If a clock period of less than 120 ns is used, the refresh requirements of the DRAMs will still be met; however, there is a slightly higher chance of a refresh slowing down a memory transaction. The CPU may select a slower refresh rate by writing a one to this bit. When the RPS bit is set to one, the memory will assume that a 80 ns clock is being used.

FMDCSR<14>DIAGNOSTIC\_REFRESH\_START (RW)

The DRS bit allows memory manufacturing to externally issue a refresh command. This bit is used in a test environment in conjunction with assertion of the FMDCSR<12>DISABLE\_REFRESH bit) to force a refresh cycle to occur. Its intention is to allow the number of clock cycles that a memory operation takes to be deterministic by removing REFRESH which is a randomly occurring event. This command will have no effect unless the DISABLE\_REFRESH bit (FMDCSR<12>) is set.

FMDCSR<15>ISOLATE (RW)

The ISOLATE bit is used to inhibit the memory module from generating an MABORT when it encounters any of the error conditions specified in the BUSCSR. This bit is set by detecting a BUSCSR error condition, an incoming MABORT or via an I/O WRITE to this bit with a data value of "1". The ISOLATE bit can only be cleared via an I/O WRITE to this bit with a data value of "0". On system reset, this bit will be cleared.

4.3.4. BASEADDR Register

The BASEADDR Register is a 12-bit read/write register that holds the starting address of the memory address space supported by this memory module and an enable bit which allows it to respond to memory space transactions. The starting address is programmable on 1 megabyte boundaries. Figure 4-7 shows the bit definitions for the BASEADDR Register.

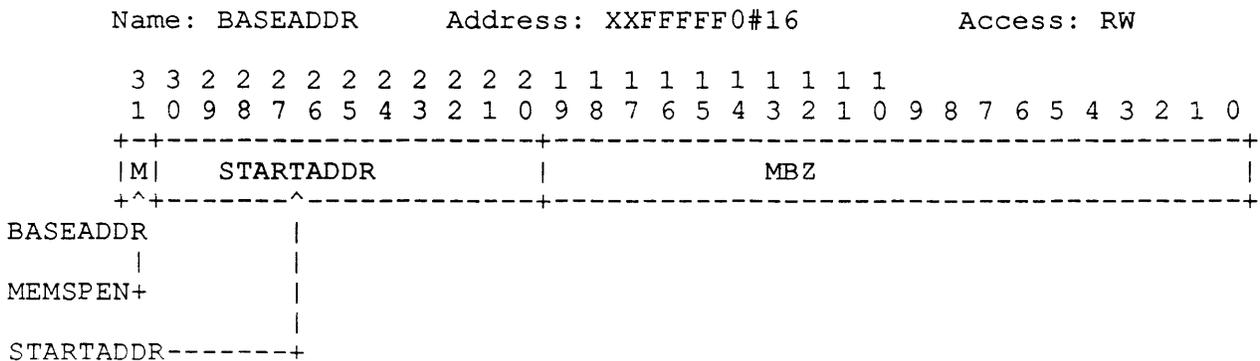


Figure 4-7: BASEADDR Register

BASEADDR<19:0> MBZ - always read as a 0.

BASEADDR<30:20>STARTADDR(RW) Base Address of memory space

The STARTADDR field is an 11-bit read/write bit field that holds the starting address, in the M-bus address map, of the memory located on this module. A read to this register returns the current STARTADDR contents. A write to this bit-field will overwrite the current contents of the STARTADDR field. On system reset, this bit field is cleared to 0's.

BASEADDR<31>MEMSPEN(RW) Memory Space Enable

This bit, when set, enables the memory module to respond to memory space transactions. Prior to setting this bit, the memory module will only respond to I/O space transactions. On system reset, this bit is cleared.

5. Power Requirements

5.1 Module Power Requirements

The memory module will require one power source, +5V. The maximum memory power will be required during a read cycle.

The following assumptions have been made to calculate memory module power requirements : read cycle time is 650 ns, CAS pulse width is 80 ns, refresh cycle time is 260 ns, and refresh period is 14 microseconds.

Module Size	Operating Mode	Current (Amps)/ Power (Watts)	
		Typical	Maximum
8 MB Firestarter	Active	8.6A/45W	11.4A/60W
8 MB Firestarter	Standby	7.0A/35W	8.6A/43W