Firefox Bus Interface Chip Specification

Revision 3.0

Tom Dewey (DECWSE::DEWEY)
Joe Minacapelli (DECWSE::MINACAPELLI)
Michael Nielsen (DECWSE::NIELSEN)

Workstation Systems Engineering
Digital Equipment Corporation
100 Hamilton Avenue
Palo Alto, CA 94301
415-853-6718

December 30, 1987

RESTRICTED DISTRIBUTION

Blank Page

Table of Contents

5. Firefox Bus Interface Chip

Revision History

Date	Version	Content/Changes
2 Dec 87	3.0	Update to reflect final FBIC design Changed CVAX target cycle time to 70ns Changed ROM maximum access time to 200ns Changed CVAX pin-bus miss to assert CDMR before CRDY and CERR Removed assertion of CDPE during ROM reads Removed reference to M-bus access to tag store in I/O space Reduced number status-indicator bits to 6 bits Added SERR and DBLE error bits to BUSCTL Removed Diagnostic Test Functions 12 and 13 Swapped IPUNIT and DEVUNIT bits in IPDVINT register Added CCLKC pin Added DATWE pin Added XOE pin Added ECL pin Removed TYPNOMEM pin Reduced CRDY and CERR to single pin signals Changed bidirectional MIRQ<3:0> to MIRQI<3:0>/MIRQO<3:0> pins Added performance section Added DC characteristics Added AC characteristics for CVAX pin-bus Added package pinout and pin assignment section
30 Apr 87	2.0	Expanded functional description Reorganized registers Updated pinout and pin descriptions Added CVAX pin-bus timing diagrams Added local I/O space
30 Jan 87	1.1	Integrated with System Specifications Added EPR access to WHAMI, CPUID, SAVGPR registers Removed internal data cache parity Added MRESET to RESET synchronization Added programmable RESET
10 Jan 87	1.0	First external release
20 Dec 86	0.0	Preliminary draft

Blank Page

5. Firefox Bus Interface Chip

This chapter, the design specification of the Firefox Bus Interface Chip (FBIC), describes the functionality of that chip. The FBIC is an LSI Logic, "Sea Of Gates" 1.5-micron gate array, and is packaged in a 223-pin PGA. The chip that is used in the Firefox Workstation to interface to the processor bus (CVAX pin-bus), the system bus (M-bus), and the per-processor, level-2 cache. This specification describes the external behavior and interface of the chip for designers and systems programmers; it does not describe internal operation of the chip. The reader is assumed to be familiar with the *Firefox M-Bus Specification*, the *CVAX CPU Chip Engineering Specification*, and the *Firefox System Specification*.

The FBIC is a triple-ported, multipurpose, bus-interface and cache controller. Figure 5-1 shows the FBIC in a processor configuration with its CVAX pin-bus, M-bus, and level-2 snoopy cache tag-store ports. The FBIC functions as both a CVAX pin-bus master and a CVAX pin-bus slave, depending on the needs of the particular module on which it resides. It also supports the M-bus write-back cache protocol defined in the *Firefox System Specification*. On modules that support external caches, the FBIC provides cache control. On modules that communicate with main memory, but do not support level-2 caches, the FBIC contains an on-chip, single-entry, level-2 snoopy cache.

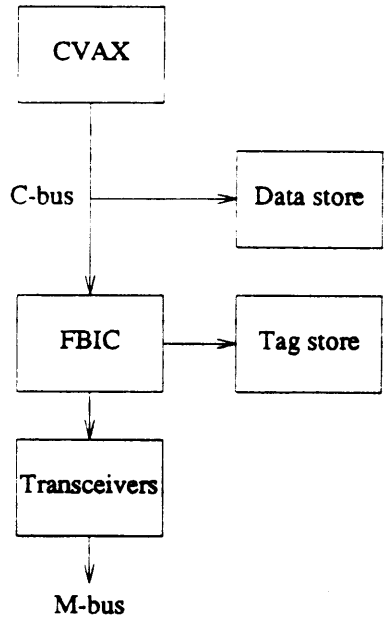


Figure 5-1: Triple-Ported FBIC Diagram

There are two separately clocked, synchronous state machines on the FBIC. The CVAX pin-bus synchronous machine monitors and initiates transactions on the local CVAX pin-bus. The M-bus synchronous state machine monitors and initiates transactions on the system M-bus. The same level-2 cache is used during accesses from either bus port. CVAX pin-bus transactions requiring cache tag compares are time-multiplexed with higher-priority accesses to the M-bus synchronous state machine. This ensures system-wide data consistency. On-chip synchronization circuits allow the two state machines to communicate across time domains.

5.1. Architecture

The FBIC is intended for use on all Firefox modules that require an interface between the CVAX pin-bus and M-bus. An FBIC monitors local CVAX pin-bus accesses and M-bus memory and I/O-space accesses and responds appropriately to requests from either bus. Support functions performed by the FBIC for the module on which it resides include the following:

- Snoopy-cache management
- Interlocked transactions
- SMP processor registers
- I/O-interrupt masking
- Interprocessor/device interrupts
- I/O-space range decoding
- 6-bit status-indicator output
- 2-bit manufacturing-mode input
- 16/32-bit ROM control
- CVAX pin-bus timeout detection
- M-bus module-type identification
- M-bus error detection and error logging
- Diagnostic function testing

The snoopy-cache-controller portion of the FBIC is used on CVAX processor modules that participate in the Firefox SMP architecture. For I/O modules that perform DMA in global memory space, the FBIC also implements a single-entry, on-chip snoopy cache.

5.1.1. FBIC Configurations

The FBIC supports the following configurations when used as a CVAX pin-bus to M-bus interface:

- Single- or dual-processor module
- Synchronous or asynchronous CVAX pin-bus based I/O-module
- Arbiter or auxiliary on CVAX pin-bus based I/O-module
- Retriable or nonretriable CVAX pin-bus based I/O-module

Table 5-1 lists the supported FBIC configurations as determined by the TYP pins.

Table 5-1: Supported FBIC configurations

Typsync	Typdual	Typagnte	Typret	Configuration
0	0	0	0	Bus adapter class (CQBIC)
0	0	0	1	Bus adapter class (CQBIC)
0	0	1	0	Unsupported
0	0	1	1	Unsupported
0	1	0	0	Unsupported
0	1	0	1	Unsupported
0	1	1	0	Unsupported
0	1	1	1	Unsupported
1	0	0	0	I/O class
1	0	0	1	I/O class
1	0	1	0	Graphics class
1	0	1	1	Graphics class
1	1	0	0	Unsupported
1	1	0	1	CPU class (B processor)
1	1	1	0	Unsupported
1	1	1	1	CPU class (A processor)

The FBIC can be used on single- or dual-processor modules. The FBIC TYPDUAL input pin determines whether the FBIC operates in single- or dual-processor mode. On dual-processor modules, the FBIC supports M-bus subarbitration between the two FBICs on the module and jointly manages a single set of M-bus transceivers and buffers. In addition, the two FBICs share the 32-Mbyte, slot-specific, I/O-space region as two 16-Mbyte halves. The FBIC TYPAGNTE input pin determines whether a dual processor operates as the A or B processor.

The FBIC can be used on synchronous or asynchronous CVAX pin-busses on I/O modules. The FBIC TYPSYNC input pin specifies whether the FBIC is on a synchronous or an asynchronous CVAX pin-bus. On a synchronous CVAX pin-bus, the FBIC synchronously samples/asserts CAS, CDAL, and CRDY/CERR but does not monitor CDS. On an asynchronous CVAX pin-bus, the FBIC treats CAS and CDS as asynchronous strobes for CDAL, with full handshake between CDS and CRDY/CERR. On an asynchronous CVAX pin-bus, the FBIC requires external synchronizers for CRDY/CERR, such as the CCLOCK implements. The FBIC supports only memory-space references from asynchronous CVAX pin-bus masters.

The FBIC can be either the arbiter or an auxiliary on an I/O-module CVAX pin-bus. The FBIC TYPAGNTE input pin specifies whether the FBIC is the arbiter or an auxiliary. (The TYPAGNTE specifies A/B processor for dual FBIC modules and grantor/grantee for single FBIC modules.) As arbiter, the FBIC is the default CVAX pin-bus master, monitors CDMR, and drives CDMG. As auxiliary, it asserts CDMR when it requires use of the CVAX pin-bus, and it monitors CDMG.

The FBIC can retry either the CVAX pin-bus or the M-bus to break deadlocks between the two busses. The FBIC TYPRET input pin specifies which bus to retry. If a CVAX pin-bus is nonretriable, I/O-space references from the M-bus that require use of the CVAX pin-bus of the module are retried when there is a pending reference between the CVAX pin-bus and M-bus. If a CVAX pin-bus is retriable, pending references between the CVAX pin-bus and M-bus are retried when an I/O-space reference from the M-bus requires use of the CVAX pin-bus.

5.1.2. Cache Consistency

The FBIC implements the snoopy-cache protocol as defined by the *Firefox System Specification*. That is, FBIC caches are write-back for unshared data and write-through for shared data. The FBIC guarantees workstation-wide cache consistency for shared data written by a single processor, consistency of adjacent

bytes of shared data written simultaneously by different processors, and consistency of bytes of shared data written simultaneously by different processors using interlocked instructions.

The FBIC does not guarantee consistency of bytes of shared data written simultaneously by different processors using noninterlocked instructions. Each cache affected will end up with the value which was written on the M-bus first. For example, if processor A writes 1 to location X, and processor B simultaneously writes 2 to location X, and processor A performs the M-bus write first, both caches will will end up with 1 in X.

No explicit software-synchronization actions are required to maintain cache consistency. There is, of course, a delay between the time a processor issues a write to its cache and the time all other caches in the workstation complete the shared write of themselves. This delay is typically 2 to 3 microseconds.

5.1.3. On-Chip Single-Entry Snoopy Cache

The FBIC implements a single-entry on-chip snoopy-cache for I/O modules that participate in the M-bus global memory space. The FBIC on-chip cache is also used on processor modules after power-up reset until they initialize and enable the external cache.

The on-chip cache consists of a single octaword data store and a single tag-store entry. The on-chip cache is not parity protected. The tag-store portion of the on-chip cache is accessible through CVAX pin-bus I/O space for diagnostic/self-test purposes.

On powerup, the on-chip data store is not initialized, and the on-chip tag store is set to 1s. That is, the on-chip cache initializes to contain a shared, dirty copy of the last octaword in memory space.

5.1.4. External Snoopy Cache

The FBIC supports an external snoopy cache for CVAX processors. This cache is 64 Kbytes in size, is direct-mapped, has octaword line size, and has longword access. The external-cache data store resides directly on the CDAL bus and has byte parity. The external-cache tag store resides on the tag port of the FBIC and has word parity.

The FBICSR<EXCAEN> register bit determines whether or not the external cache is enabled. After powerup, the external cache is off. Software must enable the cache and initialize the external tag store through I/O-space writes. The external-cache tag store is accessible in the FBIC's slot-specific I/O-space region for diagnostic/self-test purposes. The external-cache tag and data stores are inaccessible when the external cache is off.

External cache hits complete in 2 CVAX cycles; there are no wait states. Writes to shared lines are sent through to the M-bus in a pseudo-write-behind fashion. This is so termed because the CVAX receives CRDY and CDMR immediately, and therefore, no additional CVAX pin-bus transactions are completed until the write-through is completed.

The FBIC performs shared reads and write-through updates on behalf of the M-bus as required to maintain cache consistency.

The implementation of the FBIC external cache supports use of the CVAX internal cache for both instruction-stream and data-stream references. The CVAX internal cache is a subset of the external cache, which means that, whenever the FBIC removes a line from the external cache, it also invalidates that line in the CVAX internal cache, and whenever the FBIC does a write-through update of the external data store, it also invalidates that line in the CVAX internal cache. This algorithm ensures consistency between all caches in a Firefox workstation.

To maintain cache consistency, the CVAX internal cache must not be enabled while the FBIC external

cache is off.

Once external caching has been turned on, it must be manually flushed before being turned off, otherwise, modified cache data will be lost. To manually flush the external data store, read all tags through I/O space and force victim writes of any dirty lines by reading memory at a congruent address without generating any memory writes in the process. Turning off the external cache is not recommended.

5.1.4.1. Cache Read

When the CVAX issues a memory-space read, the FBIC probes the tag store to determine whether the requested location is present in the cache. In parallel with the tag probe, the FBIC enables the data store onto the CDAL in anticipation of a cache hit. If the tag probe results in a hit, the FBIC asserts CRDY to the CVAX to complete the transaction.

If the tag probe results in a miss, the FBIC immediately asserts CDMR to the CVAX, and then, on the next cycle, asserts CRDY and CERR. This causes the CVAX to retry the read after relinquishing the CVAX pin-bus. The FBIC then performs victim processing on the data store, fills the data store with the requested line, and releases the CVAX pin-bus.

5.1.4.2. Cache Write

When the CVAX issues a memory-space write, the FBIC probes the tag store to determine whether the requested location is present in the cache. If the tag probe results in a hit, the FBIC enables writes to the data store and asserts CRDY to the CVAX to complete the transaction. In parallel with writing the data store, the FBIC writes the tag store to set the dirty bit for the cache line.

If the tag has the shared bit asserted, the FBIC uses the data and byte masks it latched off the CVAX pin-bus during the data-store write to generate a masked octaword write-through transaction on the M-bus. When the write-through is completed, the FBIC updates the shared bit of the tag store from the value of MSHARED; that is, if the line is no longer in other caches, it reverts to being an unshared line.

If the tag probe results in a miss, the FBIC immediately asserts CDMR to the CVAX and then, on the next cycle, asserts CRDY and CERR. This causes the CVAX to retry the write after relinquishing the CVAX pin-bus. The FBIC then performs victim processing on the data store, fills the data store with the requested line, and releases the CVAX pin-bus.

5.1.4.3. Victim Processing and Cache Fill

When a read or write miss occurs, the FBIC initiates victim processing before filling the data store with the requested line. To terminate the CVAX transaction, the FBIC issues a bus request, which is followed immediately by a retry to ensure that the request will be honored.

If the victim line is clean, the FBIC immediately issues an M-bus memory read to obtain the requested line. In parallel with the M-bus memory read, the FBIC waits for the CVAX to grant it the CVAX pin-bus and then performs an octaword read and CVAX cache invalidate, possibly to remove the victim from the CVAX internal cache. The FBIC then waits for the M-bus memory read to be completed, and when it has the memory-read data, does a CVAX pin-bus octaword write to fill the data store with the requested line. In parallel with the data-store fill, the FBIC updates the tag store with the new address, sets the shared bit from the MSHARED signal, and clears the dirty bit. Finally, it releases the CVAX pin-bus so the CVAX can reissue the missed transaction.

If the victim line is dirty, the FBIC waits for the CVAX to grant it the CVAX pin-bus and then performs an octaword read and CVAX cache invalidate, possibly to remove the victim from the CVAX internal cache. The FBIC then issues an M-bus memory victim write to flush the dirty line back to memory. While the M-bus memory victim write is being completed, the FBIC updates the tag store to clear the dirty bit. It

then issues an M-bus memory read to obtain the requested line, when it has the memory-read data, does a CVAX pin-bus octaword write to fill the data store with the requested line. In parallel with the data-store fill, the FBIC updates the tag store with the new address, sets the shared bit from the MSHARED signal, and clears the dirty bit. Finally, it releases the CVAX pin-bus so the CVAX can reissue the missed transaction.

5.1.4.4. Shared Read

Whenever the FBIC observes the start of a memory read on the M-bus, it probes the tag store to determine whether its data store contains the requested line. If a hit results, it asserts the MSHARED signal so the module issuing the memory read can mark the cache line as being shared. The FBIC also updates its tag store to set the shared bit. If the FBIC has a tag hit to a dirty line, it asserts its MBRQ signal and takes over the role of M-bus slave from the memory module. The FBIC WAITs the M-bus until it can request the CVAX pin-bus, read the requested line out of the external-cache data store, and supply the line to the M-bus. This ensures that modified data resident in caches is used in place of stale memory data.

5.1.4.5. Shared Write

Whenever the FBIC observes the start of a memory write-through (or write unlock) on the M-bus, it probes the tag store to determine whether its data store contains the requested line. If a hit results, the FBIC asserts the MSHARED signal so the module issuing the memory write cannot clear its shared bit. Because the initiator of the write-through has set its dirty bit, the FBIC also clears its tag-store dirty bit. The FBIC BUSYs the M-bus until it can request the CVAX pin-bus, write the specified bytes to the external-cache data store, and invalidate the line in the CVAX internal cache.

5.1.5. Interlocked Transactions

The FBIC implements the M-bus interlocked-transaction mechanism as defined in the *Firefox M-Bus Specification*. That is, the FBIC implements interlocking on hexaword addresses and up to two simultaneous interlocks of different hexaword regions. Interlocks are supported for both memory space and global I/O space.

When a CVAX pin-bus master issues an interlocked read in memory space, the FBIC forces a cache miss and requests an M-bus memory-read interlock. To honor the interlock while the hexaword region is already interlocked, or while two interlocks are already in progress, the FBIC suppresses arbitrating for the M-bus. When a CVAX pin-bus master issues an unlock write in memory space, the FBIC forces a write-through operation to generate an M-bus memory-write unlock.

When a CVAX pin-bus master issues an interlocked read in I/O space, the FBIC requests an M-bus I/O-read interlock. To honor the interlock while the hexaword region is already interlocked, or while two interlocks are already in progress, the FBIC suppresses arbitrating for the M-bus. When a CVAX pin-bus master issues an unlock write in I/O space, the FBIC generates an M-bus I/O-write unlock.

The FBIC does not enforce interlocks for CVAX pin-bus references to its own 32-Mbyte, slot-specific, I/O-space region.

5.1.6. SMP Processor Registers

The FBIC implements the SMP CPUID and WHAMI processor registers. CPUID, a read-only register that specifies the hardware CPU identifier, is accessible as IPR 14 and through I/O space. WHAMI (who am I) is a read/write register that specifies the software CPU identifier, which is typically a pointer to a CPU-specific data structure. It is accessible as IPR 15 and through I/O space.

5.1.7. I/O-Interrupt Masking

The FBIC allows software control of interrupt-request connections between the CVAX pin-bus and M-bus. The interrupt request for each of the four interrupt priority levels (14, 15, 16, and 17) can be independently isolated or connected between the CVAX pin-bus and M-bus. When CVAX pin-bus and M-bus interrupt requests are connected, the signal flow can be either from the M-bus to the CVAX pin-bus (for processor modules) or from the CVAX pin-bus to the M-bus (for I/O modules).

5.1.8. Interprocessor/Device Interrupts

The FBIC implements a vectored interrupt unit that can be used for interprocessor interrupts onto the CVAX pin-bus (for processor modules) or for device interrupts onto the M-bus (for I/O modules). The vector for the interrupts is programmable, and interrupts can be generated under software control for any of the four interrupt priority levels (14, 15, 16, or 17). When the unit is used for device interrupts, interrupts can also be generated via the edge-triggered DEVIRQ input pins. To facilitate distinguishing among interrupts from an FBIC that generates multiple IPL interrupts, the interrupt unit automatically specifies the IPL level in the vector when it acknowledges an interrupt.

5.1.9. M-bus I/O-Space Range Decoder

The FBIC implements an M-bus I/O-space range decoder to allow access to I/O module resources that do not reside within the module's 32-Mbyte, slot-specific, I/O-space region. This is necessary for modules that have more than 32-Mbytes of local resources or local resources with hardwired I/O-space addresses.

The range decoder specifies a match value for the high-order 16 bits of the M-bus address and a mask value to make some of the match bits "don't cares":

```
InRange = ((Match XOR Adr<31:16>) and (not Mask) eql 0) and enable
```

The mask field allows regions larger than 64 Kbytes to be specified and multiple discontiguous regions to be matched.

The CVAX System Support chip (SSC) has registers in the CVAX address range 2014XXXX. Setting the range-decoder match field to 8014 and the mask field to 0000 will cause the FBIC to forward M-bus I/O transactions at addresses 8014XXXX onto the CVAX pin-bus.

The CVAX Q-Bus Interface chip (CQBIC) has registers in both the CVAX address ranges 2000XXXX and 2008XXXX. Setting the range-decoder match field to 8000 and the mask field to 0008 will cause the FBIC to forward M-bus I/O transactions at addresses 8000XXXX and 8008XXXX onto the CVAX pin-bus.

A third example of the use of the M-bus I/O space range decoder is a graphics module that has a 16-Mbyte frame buffer at M-bus addresses 88XXXXXX. In this case, setting the range-decoder match field to 8800, and the mask field to 00FF, will cause the FBIC to forward M-bus I/O transactions at these addresses onto the CVAX pin-bus.

The FBIC does not support local CVAX pin-bus access to the I/O-space addresses specified by the range decoder. Behavior is unpredictable if the hardware generates such references.

5.1.10. Status-Indicator Outputs

The FBIC drives the value of the six FBICSR<LEDS> register bits externally on the LEDS output pins. These outputs (with buffering) can be used to drive light-emitting diodes or hex-digit displays with diagnostic/self-test information.

5.1.11. Manufacturing-Mode Inputs

The FBIC makes the value of the two MNFMOD input pins available in the FBICSR<MFMD> register bits for use by software. This may be used by ROM-based software to operate in special modes, such as continuous self-test or loopback of various forms.

5.1.12. FBIC Support for Diagnostic/Self-test ROM

The FBIC supports CVAX pin-bus access to an external, 16- or 32-bit, 512-Kbyte diagnostic/self-test ROM. This ROM is mapped to both the VAX restart address range of 20040000#16 to 2007FFFF#16 on the CVAX pin-bus, and to the 32-Mbyte, slot-specific, I/O-space region on the CVAX pin-bus/M-bus.

The FBIC ROMWID32 input pin specifies whether the ROM is 16 or 32 bits. For 16-bit ROM, the FBIC stalls the processor and performs two separate reads to the external ROM, using the ROMWADDR signal to select the upper and lower half of a longword. It loads these words into an internal latch and provides the processor with the longword when it has been completely assembled in the latch. The ROM's data lines should be attached to the local CDAL<15:0> lines, its address lines should be connected to ROMWADDR and CDAL<15:2> through an external address latch, and its output enable should be connected to the FBIC ROMOE<0> line. For 32-bit ROM, the FBIC stalls the processor and performs a single read of the ROM. The FBIC still loads the ROM into an internal latch, disables the ROM, and redrives the data onto CDAL, actions necessary because of timing constraints on the synchronous CVAX pin-bus.

To satisfy timing constraints for a 70-ns CVAX pin-bus, the FBIC requires ROM with a maximum access time of 200 ns.

5.1.13. CVAX Pin-Bus Timeout Detection

The FBIC implements a CVAX pin-bus timer that terminates a CVAX pin-bus transaction by asserting CERR if the transaction continues for more than 2048 CVAX pin-bus clock cycles. This timeout function aborts nonexistent I/O references on the local CVAX pin-bus as well as system-level failures, such as hung interlocks and synchronization failures between different bus interfaces.

5.1.14. M-bus Module-Type Identification

The FBIC implements the MODTYPE register as required by the *Firefox M-Bus Specification*. The FBIC MODCL input pins are used to generate the MODTYPE<CLASS> field as a CPU, I/O, graphics, or bus adapter class. The MODTYPE<SUBCLASS> field reflects the value of the FBIC TYPDUAL input pin.

5.1.15. FBIC Error Detection and Logging

To allow precise error logging for diagnostic and Field Service analysis, the FBIC is designed to implement extensive error detection and to save as much nonredundant state as is practical. Every FBIC will detect several types of M-bus, CVAX pin-bus, and external-cache errors and take appropriate action to ensure that the error is recognized by a processor.

The types of M-bus, CVAX pin-bus, and external-cache errors detected by the FBIC are logged in the BUSCSR register by the assertion of a status bit. For errors that require assertion of MABORT, a machine check must always be initiated for every processor in the Firefox system. The FBIC accomplishes this by asserting MEMERR to the local CVAX processor. For modules that do not contain a processor, MEMERR can be used to generate a device interrupt by a loopback to the FBIC DEVIRQ<3> device-interrupt request. If a CVAX I/O cycle is in progress when MABORT is asserted, the FBIC can also terminate a CVAX I/O cycle with error status. Although the FBIC makes a best effort to detect all errors, it is not intended that it catch 100 percent of M-bus, CVAX pin-bus, or external-cache faults.

The FBIC detects the following M-bus errors:

- MBRQ arbitration
- Invalid MCMD encoding
- Invalid data supplied
- Cache tag parity
- MCMD parity
- MSTATUS parity
- MDAL parity
- Interlocked violation
- No slave response
- Slave wait/busy timeout
- Error MSTATUS
- M-bus double errors

The FBIC detects the following CVAX pin-bus errors:

- CDAL parity when sinking data
- Cache tag parity
- Transaction timeout

When the FBIC detects M-bus errors, it freezes the BUSCTL, BUSADR, and BUSDAT log registers in addition to asserting status bits in the BUSCSR register.

Continued system operation after an M-bus error or a cache tag-parity error is not recommended. Because the FBIC is not designed to allow system recovery after M-bus errors and external-cache tag-parity errors, the operating system should log all saved error information and then reboot. The FBIC does not guarantee cache data consistency after an M-bus error or a cache tag-parity error, nor does it guarantee that memory-write transactions in progress at the time of the M-bus error will be completed successfully.

A more detailed description of error conditions and their recommended resolution can be found in Chapter 15, "Firefox Error-Handling Specification."

5.1.16. Diagnostic Function Testing

The FBICSR<TSTFNC> bits allow diagnostic software to exercise various functions of the FBIC involving error generation/checking and bus protocol. In addition, the BUSCTL, BUSADR, and BUSDAT error-logging registers allow observation of M-bus control and data signals.

5.1.17. Firefox I/O-Space Mapping

The M-bus supports a 2-Gbyte memory address space and a separate 2-Gbyte I/O address space. Each module is assigned a 32-Mbyte region of I/O space as a function of its backplane slot. Address-space assignments are listed in Table 5-2.

Table 5-2: Address-Space Assignments for the M-Bus Module

M-Bus Address Range	VAX Address Range	Mbytes	Function
00000000 . . . 1FFFFFFF	00000000 . . . 1FFFFFFF	512	Memory space
20000000 . . . 7FFFFFFF	N/A	1536	Reserved memory space
80000000 . . . 87FFFFFF	20000000 . . . 27FFFFFF	128	Global I/O space
88000000 . . . 8FFFFFFF	28000000 . . . 2FFFFFFF	128	Local I/O space
90000000 . . . 91FFFFFF	30000000 . . . 31FFFFFF	32	Slot 0 I/O space
92000000 . . . 93FFFFFF	32000000 . . . 33FFFFFF	32	Slot 1 I/O space
94000000 . . . 95FFFFFF	34000000 . . . 35FFFFFF	32	Slot 2 I/O space
96000000 . . . 97FFFFFF	36000000 . . . 37FFFFFF	32	Slot 3 I/O space
98000000 . . . 99FFFFFF	38000000 . . . 39FFFFFF	32	Slot 4 I/O space
9A000000 . . . 9BFFFFFF	3A000000 . . . 3BFFFFFF	32	Slot 5 I/O space
9C000000 . . . 9DFFFFFF	3C000000 . . . 3DFFFFFF	32	Slot 6 I/O space
9E000000 . . . 9FFFFFFF	3E000000 . . . 3FFFFFFF	32	Slot 7 I/O space
A0000000 . . . FFFFFFFF	N/A	1536	Reserved I/O space

The FBIC only supports a 30-bit physical address space on the CVAX pin-bus; it does not support M-bus reserved memory space or M-bus reserved I/O space. Table 5-3 lists the connection of the CVAX pin-bus address signals to MDAL signals for cycle P2 of M-bus transactions to convert from 30-bit physical addresses to 32-bit physical addresses. For all other M-bus cycles, the VAX DAL<31:00> is directly connected to MDAL<31:00>.

Table 5-3: VAX 30-Bit Physical Address to M-Bus 32-Bit Address

M-Bus Address	VAX Address
MDAL<31>	DAL<29>
MDAL<30>	0
MDAL<29>	0
MDAL<28:00>	DAL<28:00>

Figure 5-2 shows the FBIC address space from the perspective of a CVAX pin-bus device. CVAX pin-bus references to memory space access the memory through the FBIC cache. The FBIC forwards CVAX pin-bus references to global M-bus I/O space and slot-specific I/O space for other slots onto the M-bus. It handles CVAX pin-bus references to its own registers, tag store, and ROM directly, without using the M-bus. It does not participate in CVAX pin-bus references to the local I/O space and the first 30 Mbytes of the slot-specific I/O space.

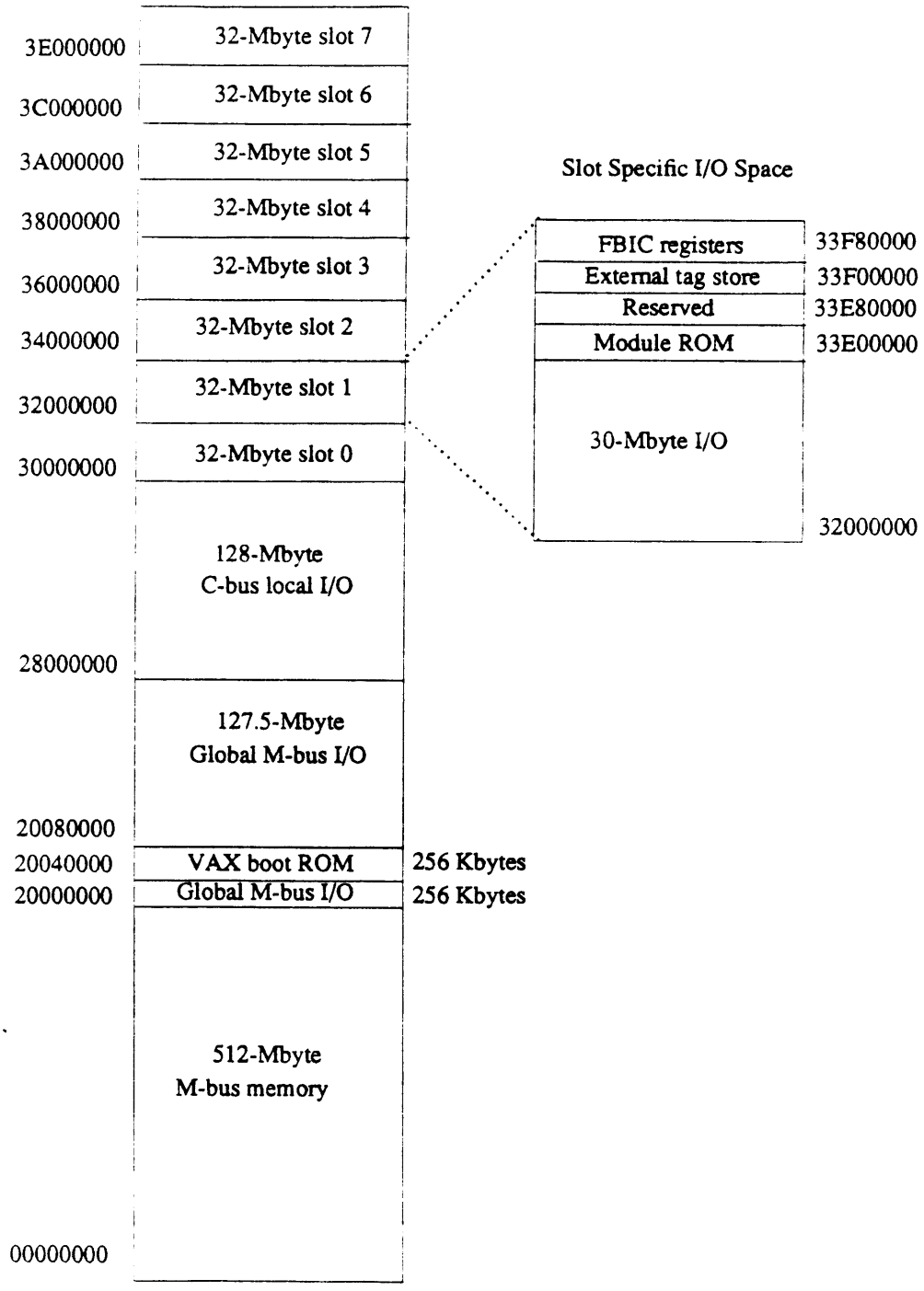


Figure 5-2: FBIC C-Bus Address Space

Figure 5-3 shows the FBIC address space from the perspective of an M-bus device. M-bus references to memory-space access the FBIC cache as appropriate for shared data. The FBIC forwards M-bus references to global I/O space onto its CVAX pin-bus only if its range decoder matches the address. It handles M-bus references to its registers directly, without using the CVAX pin-bus, but forwards M-bus references to its 30 Mbytes of slot-specific I/O space and to its ROM onto the CVAX pin-bus. The FBIC does not participate in other regions of M-bus I/O space.

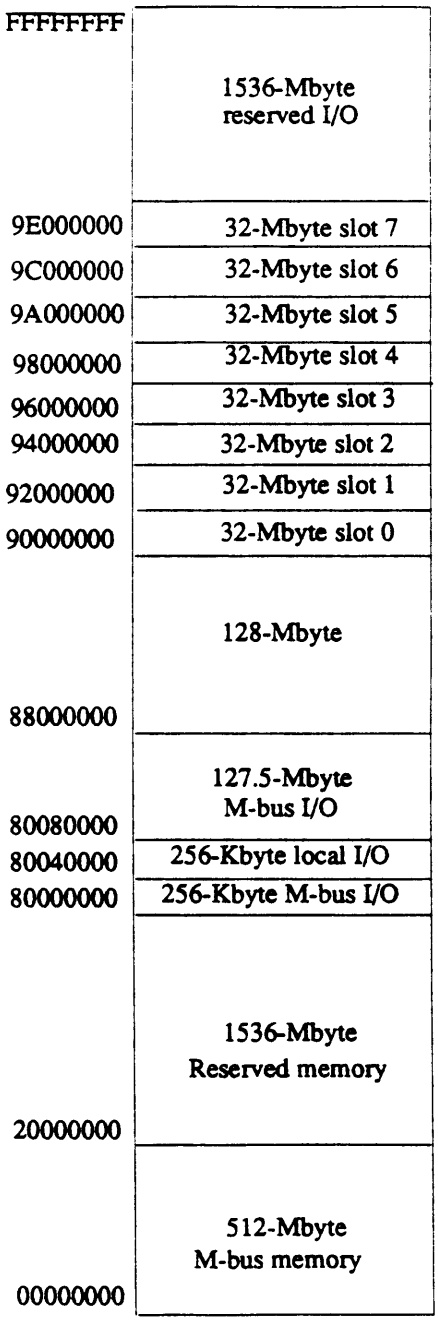


Figure 5-3: FBIC M-Bus Address Space

5.1.18. FBIC I/O-Space Address Mapping

By logically connecting the MDAL<31> and CDAL<29> address lines, the FBIC directly maps into CVAX pin-bus I/O-space accesses those M-bus I/O-space accesses that fall in the 32-Mbyte, slot-specific region of the module on which the FBIC resides. In the same manner, the FBIC also directly maps into M-bus I/O-space accesses the CVAX pin-bus I/O-space accesses to the global I/O space region (except those that reference the bootstrap ROM region of 20040000 . . . 2007FFFF) and to the 32-Mbyte, slot-specific region of other modules. The FBIC does not participate in local I/O-space accesses. CVAX pin-bus I/O-space accesses to the 32-Mbyte, slot-specific region of the module on which the FBIC resides are discussed following Figure 5-4.

The FBIC determines the 32-Mbyte M-bus address range to which it must respond by concatenating the MID (Module ID) input pins, as shown in Figure 5-4.

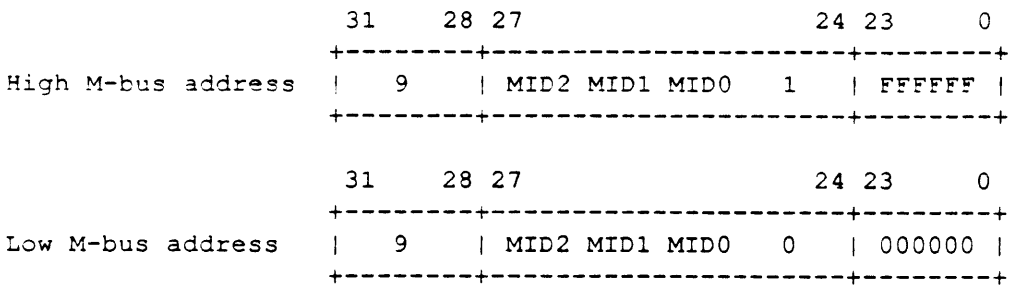


Figure 5-4: Construction of the Module-Specific M-Bus I/O-Space Range

Because there are two separate CVAX pin-busses but only one 32-Mbyte, slot-specific address region on the Firefox dual-CVAX processor module, the 32-Mbyte region is divided into two, 16-Mbyte regions (one per CVAX pin-bus). This means that the FBIC will map into CVAX pin-bus I/O-space accesses only those M-bus I/O-space accesses that fall both in the 32-Mbyte, slot-specific region of a processor module and within the 16-Mbyte range specific to that particular processor. Moreover, the FBIC will not map into M-bus I/O-space accesses the CVAX pin-bus I/O-space accesses to the 16-Mbyte, slot-specific region belonging to the local processor on a Firefox dual-CVAX processor module, although it will map into M-bus I/O-space accesses the CVAX pin-bus I/O-space accesses to the 16-Mbyte, slot-specific region belonging to the other processor on a Firefox dual-CVAX processor module.

The FBIC determines the module type on which it resides from the TYPDUAL, TYPAGNTE, TYPSYNC, and TYPRET input pins defined in Section 1.2, "Interface," later in this chapter. If the FBIC resides on a Firefox dual-CVAX processor module, it determines the per-processor 16-Mbyte M-bus address range to which it must respond by concatenating the MID and the TYPAGNTE input pins, as shown in Figure 5-5. The CPUID register implemented on the FBIC uses these same signals to uniquely identify each processor in the Firefox system.

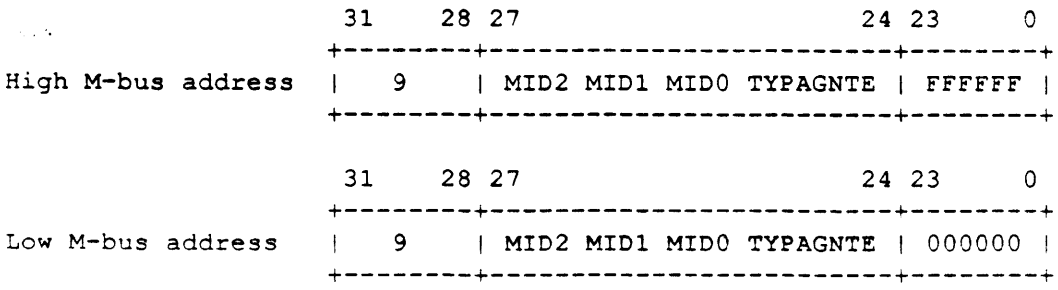


Figure 5-5: Construction of the Processor-Specific M-Bus I/O-Space Range

Within the 16- or 32-Mbyte M-bus I/O-space address range that corresponds to a particular FBIC, the high 2-Mbyte region is reserved for access to FBIC registers, external-cache tag store, and module-specific

ROM. The FBIC will map M-bus I/O-space accesses in this 2-Mbyte range into the appropriate FBIC register and ROM accesses. CVAX pin-bus accesses outside the local 2-Mbyte range but within the 16-Mbyte processor-specific region will not be mapped by the FBIC into corresponding M-bus accesses.

Software running on a particular CVAX processor can determine the 2-Mbyte I/O-space range at which to access local resources by reading the processor's CPUID or WHAMI register with the VAX *move from processor register* instruction.

Because all CVAX processors will expect to find diagnostic/self-test ROM starting at address 20040000#16 on the CVAX pin-bus, the ROM is also mapped by the FBIC to this CVAX pin-bus location. CVAX pin-bus I/O-space accesses to this ROM will never be mapped to M-bus accesses in the corresponding M-bus I/O-space range.

Within the 2-Mbyte region, the top 512 Kbytes are reserved for FBIC registers, the next 512 Kbytes are reserved for the external tag store, the next 512 are reserved for future use, and the bottom 512 Kbytes are reserved for the module's ROM.

The FBIC registers require only a 64-byte region of the 512-Kbyte register region; thus, they appear 8192 times within the region. For compatibility with possible future extensions, software must only access the registers at their designated addresses.

Each external-cache tag appears in I/O space four times at each longword within the naturally aligned octaword; that is, the tag for external-cache line 0 is accessible at offsets XXF00000, XXF00004, XXF00008, and XXF0000C. Software can access the tag for a cache line at any of the four addresses. The entire set of tags reappear eight times within the 512-Kbyte tag space. For compatibility with larger external caches, software must access the tags only in the XXF00000 . . . XXF0FFFF region.

The ROM appears at both the VAX restart addresses of 20040000 and XXE00000. The FBIC only responds to addresses 20040000 . . . 2007FFFF (a total of 256 Kbytes) because of the address-space assignments of the CQBIC. Modules with 512 Kbytes of ROM can access only the upper 256 Kbytes of the ROM at XXE40000 . . . XXE7FFFF. ROMs smaller than their assigned regions reappear multiple times within their regions.

5.1.19. FBIC Registers

The FBIC supports a total of 13 internal registers that implement the following:

- M-bus module-type identification
- M-bus error detection and error status
- M-bus error-control signal log
- M-bus error-address signal log
- M-bus error data signal log
- Control and status of the FBIC
- I/O-interrupt masking
- Diagnostic/self-test LEDs
- I/O-space range decoding
- Interprocessor-interrupt delivery
- Interprocessor-interrupt vector
- Device-interrupt delivery
- Device-interrupt vector
- Software processor identification
- Hardware processor identification
- Interlocked-transaction status
- Scratch-register storage

Table 5-4 lists address offset, access, and function of the FBIC registers, each of which is described in detail in the subsections that follow the table. The FBIC registers are at the top of a 512-Kbyte region; references to unassigned addresses produces unpredictable results.

Table 5-4: FBIC Register Map

Register	Address	R/W*	Register Description
MODTYPE	XXFFFFFC#16	R	Module type
BUSCSR	XXFFFFF8#16	R/W	M-bus error status
BUSCTL	XXFFFFF4#16	R/W	M-bus error-control signal log
BUSADR	XXFFFFF0#16	R/W	M-bus error-address signal log
BUSDAT	XXFFFFEC#16	R/W	M-bus error-data signal log
FBICSR	XXFFFFE8#16	R/W	FBIC control status
RANGE	XXFFFFE4#16	R/W	I/O-space-range decode
IPDVINT	XXFFFFE0#16	R/W	Interprocessor/device interrupt
WHAMI	XXFFFFDC#16	R/W	Unique software ID
CPUID	XXFFFFD8#16	R	Unique hardware ID
IADR1	XXFFFFD4#16	R/W	Interlock 1 address
IADR2	XXFFFFD0#16	R/W	Interlock 2 address
SAVGPR	XXFFFFC4#16	R/W	Halt-code scratch

* R/W = Read/Write

In the FBIC register descriptions that follow, all fields labeled MBZ (must be zero) indicate reserved bits. These bits should always be written with 0s to provide for future enhancements; reading these bits will return all 0s. All register bits with write access are cleared by MRESET. All registers with write access support byte writes. Unless otherwise noted, all register bits are active-high.

5.1.19.1. MODTYPE Register

The MODTYPE register is a read-only register that indicates the class of the module, information specific to the class, the interface chip, and the revision of the interface chip. Figure 5-6 shows the bit definitions for the MODTYPE register.

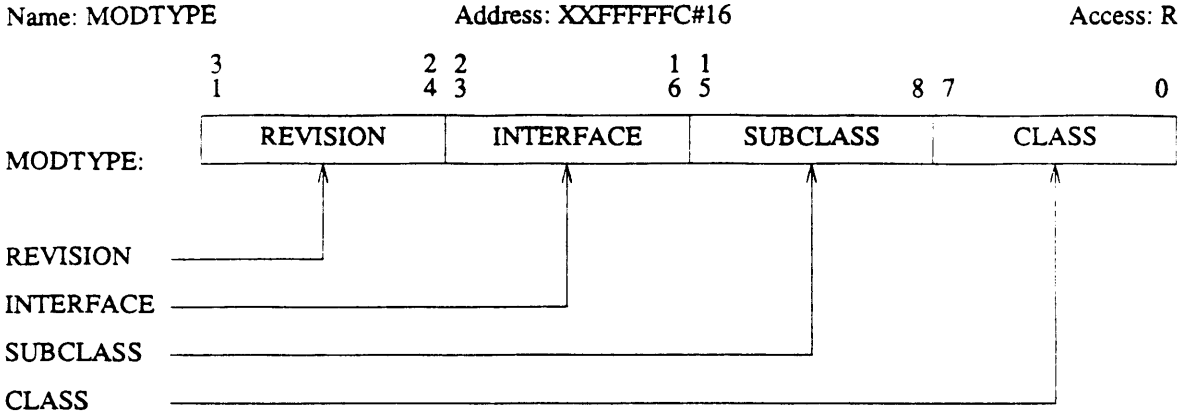


Figure 5-6: MODTYPE Register

MODTYPE<7:0> CLASS (R) Module Class

The CLASS field indicates the class of module on which the FBIC is physically situated. These bits reflect the value of the FBIC MODCL<1:0> input pins. Table 5-5 shows the value of the MODTYPE<7:0> as a function of the MODCL<1:0> pins.

Table 5-5: MODCL Encoding for MODTYPE<CLASS>

MODCL<1:0>	MODTYPE<7:0>	Class
0	01	Bus adapter
1	02	Graphics
2	04	I/O
3	08	CPU

MODTYPE<15:8> SUBCLASS (R) Module Subclass

The FBIC echos the state of the TYPDUAL input pin on MODTYPE<8>. MODTYPE<15:9> always reads as 0s.

MODTYPE<23:16> INTERFACE (R) Interface Chip

The FBIC specifies the FBIC code of 1 in this field.

MODTYPE<31:24> REVISION (R) Interface Revision

The FBIC specifies revision 1 in this field.

As an example of the contents of the MODTYPE register, an FBIC on a Firefox dual-CVAX processor module has a MODTYPE value of 01010108; the FBIC on an I/O module has MODTYPE of 01010004.

5.1.19.2. BUSCSR Register

The BUSCSR register is a read/write register that maintains M-bus error status and controls M-bus error logging. Figure 5-7 shows the bit definitions for the BUSCSR register. All bits of BUSCSR are active-low. Whenever the FRZN bit is set, the FBIC logs errors in BUSCSR by clearing the corresponding status bit. When the FBIC clears a status bit, it also clears the FRZN bit. When the FRZN bit is clear, the FBIC does not alter the value of BUSCSR. This means that the BUSCSR saves the first error event. If errors occur simultaneously and error logging is enabled, the FBIC clears multiple-status bits. To enable error logging, write FFFFFFFF#16 to BUSCSR. The result of simultaneous error events and I/O-space writes to BUSCSR is unpredictable.

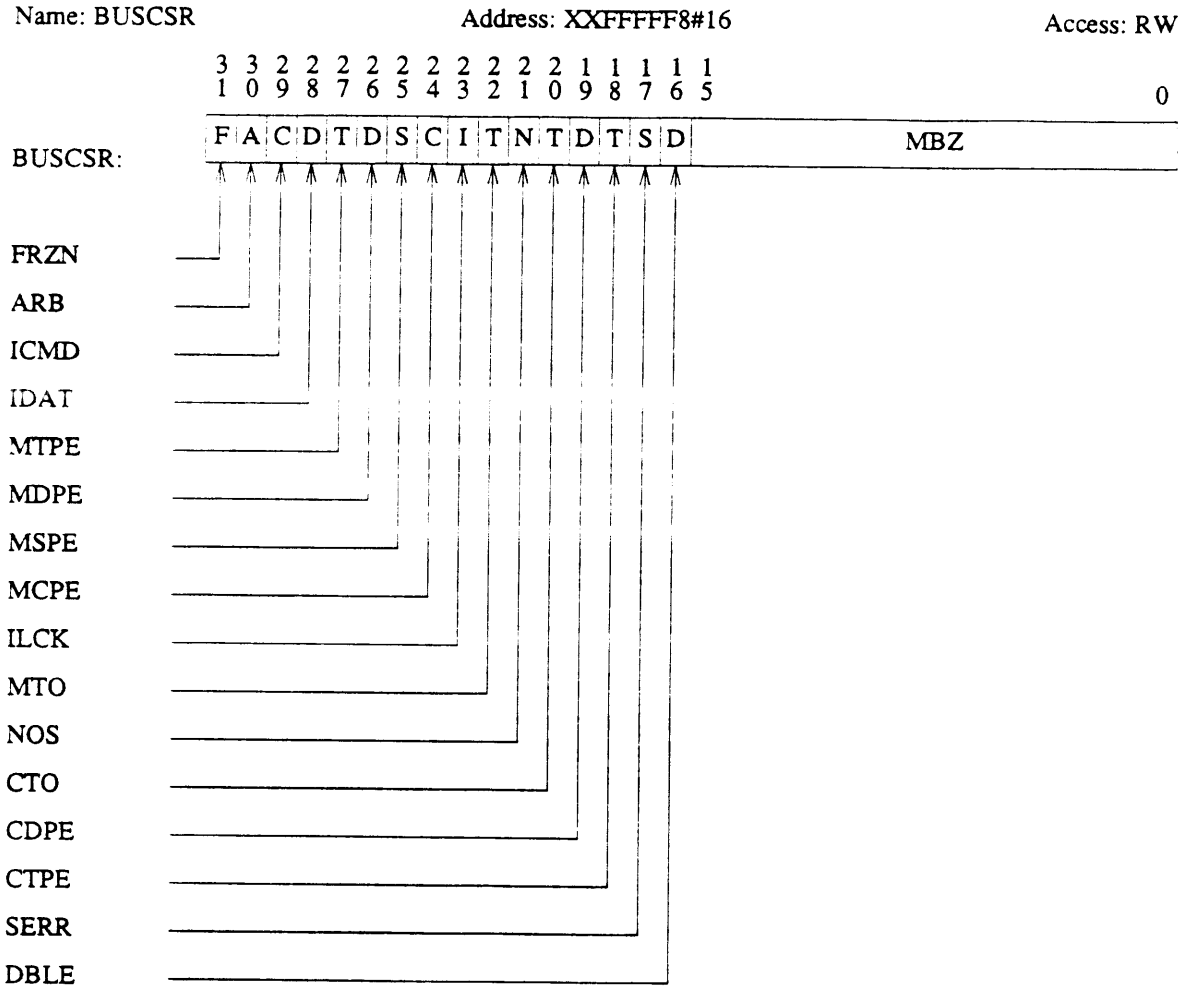


Figure 5-7: BUSCSR Register

BUSCSR<16> DBLE (RW) Double M-Bus Error

The FBIC clears DBLE when an M-bus error is detected and the FRZN (BUSCSR<31>) bit is already cleared.

BUSCSR<17> SERR (RW) Error on MSTATUS

The FBIC clears SERR when an M-bus slave supplies an MSTATUS value of error (11#2).

BUSCSR<18> CTPE (RW) CDAL Tag-Store Parity Error

The FBIC clears CTPE when a CVAX pin-bus memory-space reference detects a tag-store parity error. It also asserts the CERR signal to terminate the CVAX pin-bus transaction when it detects a tag-store parity error.

BUSCSR<19> CDPE (RW) CDAL Parity Error

The FBIC clears CDPE when it reads data from the CDAL bus. It also clears CDPE when it accepts write data from the CDAL bus, if the CDAL and CSDP signals are inconsistent and the CDPE signal is asserted.

BUSCSR<20> CTO (RW) CDAL Timeout

The FBIC clears CTO when CAS is asserted for 2048 continuous CVAX pin-bus cycles. It also asserts the CERR signal to terminate the transaction.

BUSCSR<21> NOS (RW) M-Bus No-Slave Response

The FBIC clears NOS when no M-bus slave responds to an M-bus transaction initiated by this FBIC.

BUSCSR<22> MTO (RW) M-Bus Slave Timeout

The FBIC clears MTO when an M-bus slave either specifies WAIT status or asserts MBUSY for 256 continuous M-bus cycles. The FBIC also generates an MABORT sequence when it clears MTO.

BUSCSR<23> ILCK (RW) M-Bus Interlock Violation

The FBIC clears ILCK when an unexpected interlocked operation is issued on the M-bus. An unexpected interlocked operation would be an interlocked read to a hexaword address that is already interlocked, a third interlocked read when two interlocks are already in progress, or an unlock write to a hexaword address that is not interlocked. The FBIC also generates an MABORT sequence when it clears ILCK.

BUSCSR<24> MCPE (RW) M-Bus MCMD Parity Error

The FBIC clears MCPE when it detects a parity error on the MCMD signals. It checks MCMD parity the cycle after the value is 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 FBIC also generates an MABORT sequence when it clears MCPE.

BUSCSR<25> MSPE (RW) M-Bus MSTATUS Parity Error

The FBIC clears MSPE when it detects a parity error on the MSTATUS signals. It checks MSTATUS parity the cycle after the value is 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 FBIC also generates an MABORT sequence when it clears MSPE.

BUSCSR<26> MDPE (RW) M-Bus MDAL Parity Error

The FBIC clears MDPE when it detects a parity error on the MDAL signals. It checks MDAL parity the cycle after the value is on the M-bus--that is, P3; memory read P8, P9, P10, P11; memory write P4, P5, P6, first P7; I/O read P5; I/O write first P4; and interrupt acknowledge P6. The FBIC also generates an MABORT sequence when it clears MDPE.

BUSCSR<27> MTPE (RW) M-Bus Tag Parity Error

The FBIC clears MTPE when it does an M-bus memory-space tag probe and detects a tag-store parity error. It also generates an MABORT sequence when it clears MTPE.

BUSCSR<28> IDAT (RW) M-Bus Invalid Data Supplied

The FBIC clears IDAT when it supplies data onto MDAL indicating that it detected a parity error on obtaining read data from the CDAL.

BUSCSR<29> ICMD (RW) M-Bus Invalid MCMD Encoding

The FBIC clears ICMD when it detects that an M-bus transaction has an undefined MCMD encoding during P2. It also generates a MABORT sequence when it clears ICMD.

BUSCSR<30> ARB (RW) M-Bus Arbitration Error

The FBIC clears ARB when it detects an M-bus arbitration error. Arbitration errors represent either premature deassertion of an MBRQ signal when the FBIC is monitoring a transaction, or assertion of another MBRQ signal when the FBIC has its MBRQ signal asserted as a master or slave of a transaction. The FBIC also generates an MABORT sequence when it clears ARB.

BUSCSR<31> FRZN (RW) M-Bus Error Logging Frozen

The FBIC clears FRZN when it clears one or more of the BUSCSR<30:0> bits or when it detects MABORT asserted.

5.1.19.3. BUSCTL Register

The BUSCTL register is a read/write register that logs the value of M-bus control signals at the time the FBIC detects an error. Write access to the BUSCTL register is only for diagnostic testing; the result of I/O-space writes to the BUSCTL register when error logging is enabled is unpredictable. Figure 5-8 shows the bit definitions for the BUSCTL register.

Name: BUSCTL Address: XXFFFFF4#16 Access: RW

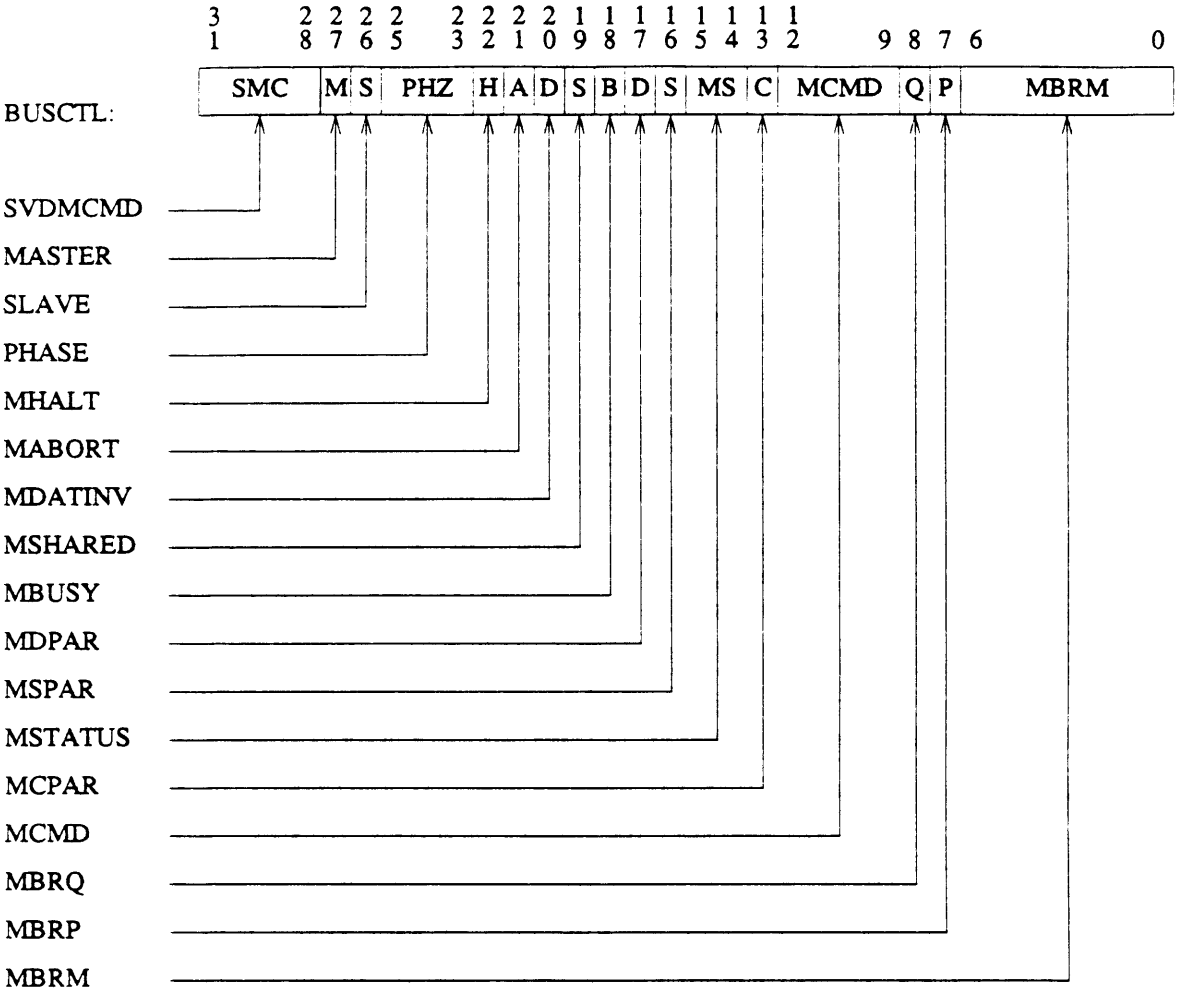


Figure 5-8: BUSCTL Register

BUSCTL<6:0> **MBRM** (RW) **M-Bus MBRM Signals**

The FBIC continuously updates MBRM from the M-bus MBRM signals when BUSCSR<FRZN> is set or MRESET is asserted. This provides module-present information on reset and M-bus MBRQ status information during M-bus errors.

BUSCTL<7> MBRP (RW) M-Bus MBRP Signal

The FBIC continuously updates MBRP from the FBIC MBRP input pin when BUSCSR<FRZN> is set or MRESET is asserted. This provides module-present information on reset and M-bus MBRQ status information during M-bus errors.

BUSCTL<8> MBRQ (RW) M-Bus MBRQ Signal

The FBIC continuously updates MBRQ from the FBIC MYMBRQ output pin when BUSCSR<FRZN> is set.

BUSCTL<12:9> MCMD (RW) M-Bus MCMD Signals

The FBIC updates MCMD from the M-bus MCMD signals when it checks parity on the MCMD signals and BUSCSR<FRZN> is set.

BUSCTL<13> MCPAR (RW) M-Bus MCPAR Signal

The FBIC updates MCPAR from the M-bus MCPAR signal when it checks parity on the MCMD signals and BUSCSR<FRZN> is set.

BUSCTL<15:14> MSTATUS (RW) M-Bus MSTATUS Signals

The FBIC updates MSTATUS from the M-bus MSTATUS signals when it checks parity on the MSTATUS signals and BUSCSR<FRZN> is set.

BUSCTL<16> MSPAR (RW) M-Bus MSPAR Signal

The FBIC updates MSPAR from the M-bus MSPAR signal when it checks parity on the MSTATUS signals and BUSCSR<FRZN> is set.

BUSCTL<17> MDPAR (RW) M-Bus MDPAR Signal

The FBIC updates MDPAR from the M-bus MDPAR signal when it checks parity on the MDAL signals and BUSCSR<FRZN> is set.

BUSCTL<18> MBUSY (RW) M-Bus MBUSY Signal

The FIBC continuously updates MBUSY from the M-bus MBUSY signal when BUSCSR<FRZN> is set.

BUSCTL<19> MSHARED (RW) M-Bus MSHARED Signal

The FBIC continuously updates MSHARED from the M-bus MSHARED signal when BUSCSR<FRZN> is set.

BUSCTL<20> MDATINV (RW) M-Bus MDATINV Signal

The FBIC continuously updates MDATINV from the M-bus MDATINV signal when BUSCSR<FRZN> is set.

BUSCTL<21> MABORT (RW) M-Bus MABORT Signal

The FBIC updates MABORT from the M-bus MABORT signal when BUSCSR<FRZN> is set.

BUSCTL<22> MHALT (RW) M-Bus MHALT Signal

The FBIC updates MHALT from the M-bus MHALT signal when BUSCSR<FRZN> is set.

BUSCTL<25:23> PHASE (RW) M-Bus Transaction Phase

The FBIC continuously updates PHASE from the M-bus state-machine transaction phase when BUSCSR<FRZN> is set. Table 5-6 shows the encoding of PHASE as a function of the M-bus transaction phase.

Table 5-6: Encoding of the BUSCTL M-Bus Transaction Phase

M-Bus	PHASE
P1	0
P2	1
P3	2
P4	3
P5	4
P6	5
P7	6
P8	7
P9	7
P10	7

BUSCTL<26> SLAVE (RW) M-Bus Slave

The FBIC continuously updates SLAVE from the M-bus state-machine slave mode when BUSCSR<FRZN> is set.

BUSCTL<27> MASTER (RW) M-Bus Master

The FBIC continuously updates MASTER from the M-bus state-machine master mode when BUSCSR<FRZN> is set.

BUSCTL<31:28> SVDMCMD (RW) M-Bus Saved MCMD Signals

The FBIC updates SVDMCMD from the P2 value of the M-bus MCMD signals during P3 of every transaction when BUSCSR<FRZN> is set.

5.1.19.4. BUSADR Register

The BUSADR register is a read/write register that maintains the M-bus address of the last M-bus transaction. Write access to the BUSADR register is only for diagnostic testing; the result of I/O-space writes to the BUSADR register when error logging is enabled is unpredictable. Figure 5-9 shows the bit definitions for the BUSADR register.

Name: BUSADR Address: XXFFFFF0#16 Access: RW

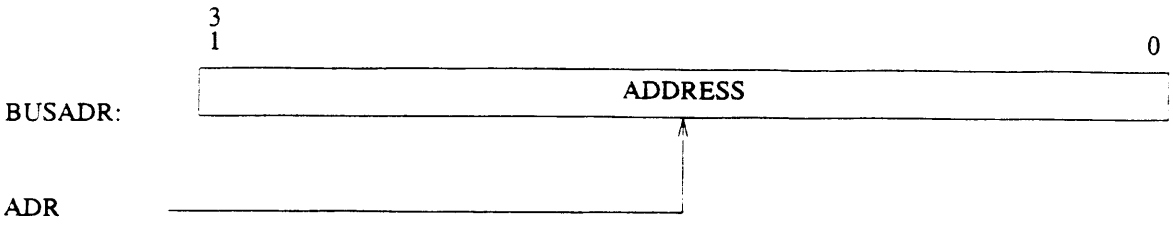


Figure 5-9: BUSADR Register

BUSADR<31:0> ADR (RW) M-Bus Error Address

The FBIC updates ADR from the P2 value of the M-bus MDAL signals during P3 of every transaction when BUSCSR<FRZN> is set.

5.1.19.5. BUSDAT Register

The BUSDAT register is a read/write register that maintains the last M-bus MDAL value. Write access to the BUSDAT register is only for diagnostic testing; the result of I/O-space writes to the BUSCTL register when error logging is enabled is unpredictable. Figure 5-10 shows the bit definitions for the BUSDAT register.

Name: BUSDAT Address: XXFFFFEC#16 Access: RW

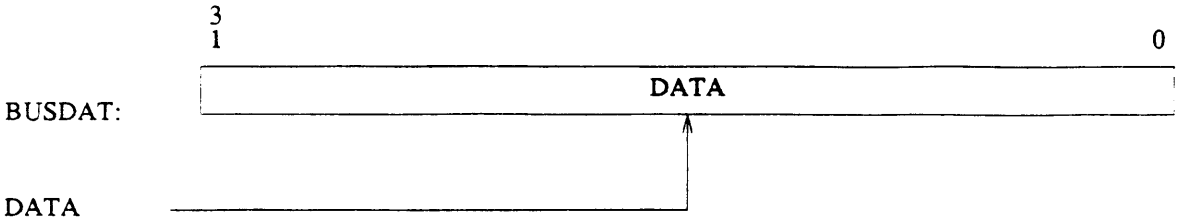


Figure 5-10: BUSDAT Register

BUSDAT<31:0> DATA (RW) M-Bus Error Data

The FBIC updates DATA from the M-bus MDAL signals when the FBIC checks parity on the MDAL signals and BUSCSR<FRZN> is set.

5.1.19.6. FBICSR Register

The FBICSR register is a read/write control and status register for the FBIC itself. Figure 5-11 shows the bit definitions for the FBICSR register.

Name: FBICSR Address: XXFFFFE8#16 Access: RW

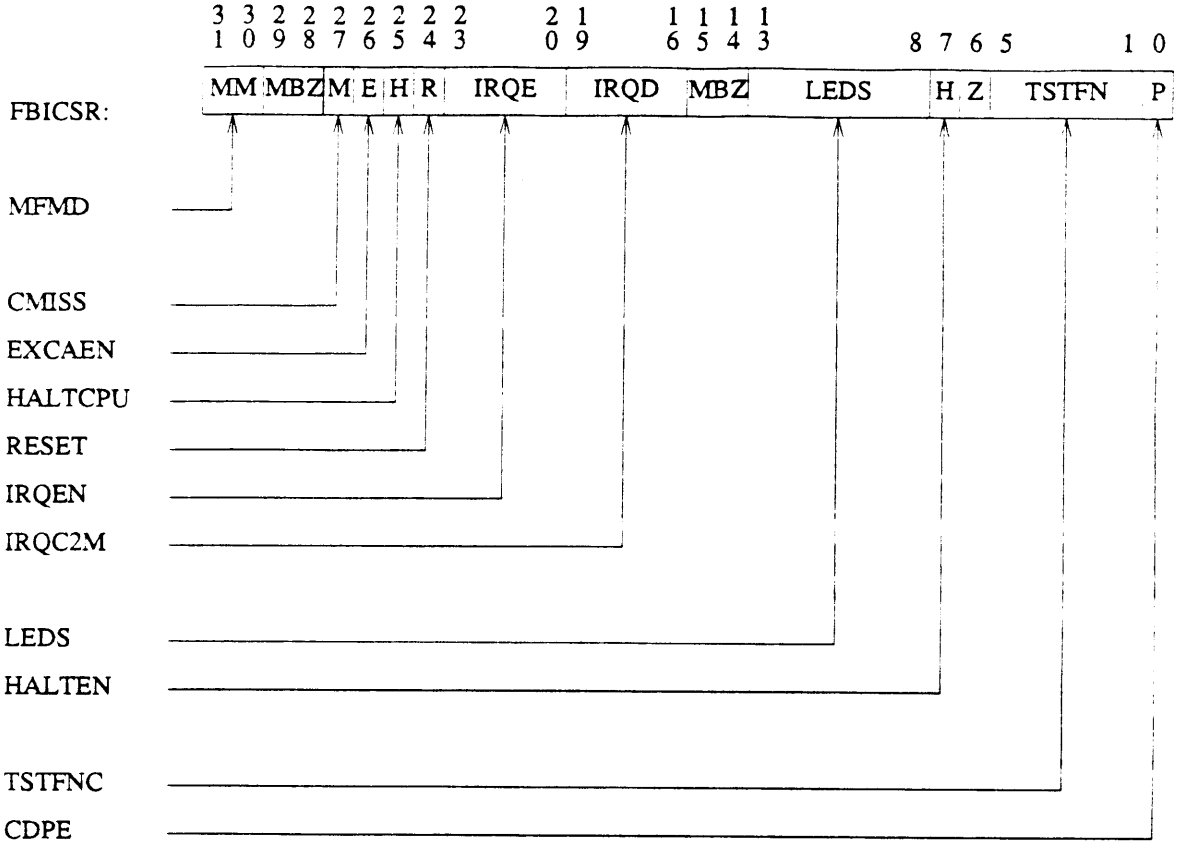


Figure 5-11: FBIC Control and Status Register

FBICSR<0> CDPE (RW) CVAX Pin-Bus Parity-Check Enable

The CDPE bit enables checking on the CVAX pin-bus CDAL/CCSDP signals. The FBIC asserts the CVAX pin-bus CDPE signal when it or the external-cache data store drives data onto the CVAX pin-bus and FBICSR<CDPE> is asserted.

FBICSR<5:1> TSTFNC (RW) FBIC Diagnostic Test Function

The TSTFNC bits select a test function for use by diagnostics to facilitate testing of various FBIC functions. The TSTFNC bits are not cleared by MABORT. Table 5-7 lists encodings for the test function.

Table 5-7: Encodings for the FBICSR Diagnostic Test Function

TSTFNC	Diagnostic Test Function
0	Logically isolate FBIC from M-bus
1	Complement output of MCMD parity generator
2	Complement output of MSTATUS parity generator
3	Complement output of MDAL parity generator
4	Complement output of MCMD valid decoder
5	Force all reads/writes interlocked, honor ISIP
6	Force all reads/write interlocked, ignore ISIP
7	Assert MBUSY
8	Assert MDATINV
9	Assert MSHARED
10	Assert MABORT
11	Complement output of CDAL parity generator
12	Unpredictable
13	Unpredictable
14	Force cache hit
15..30	Unpredictable
31	Normal mode

FBICSR<7> HALTEN (RW) Enable CPU Halts

The FBIC asserts the CHALT signal when (FBICSR<HALTCPU> OR MHALT) AND HALTEN is true. Clearing the HALTEN bit allows console software to suppress (additional) halts. The HALTEN bit is automatically cleared by the FBIC when MHALT and HALTEN are asserted to provide a debouncing mechanism for the MHALT switch. However, the HALTEN bit is not cleared when FBICSR<HALTCPU> and HALTEN are asserted.

FBICSR<13:8> LEDS (RW) FBIC LED Outputs

The FBIC drives the module LEDS with the value of LEDS. Clearing a bit of LEDS illuminates the corresponding module LED. Setting a bit of LEDS turns off the corresponding module LED. Because MRESET clears the FBICSR register, the FBIC illuminates all module LEDS during MRESET.

FBICSR<19:16> IRQC2M (RW) Interrupt-Request Direction

IRQC2M specifies the flow direction of the MIRQ/CIRQ signals. If a bit of IRQC2M is 1, assertion of the corresponding CIRQ signal causes assertion of the corresponding MIRQ signal, provided that the corresponding FBICSR<IRQEN> bit is 1. This is used by modules that generate interrupts *onto* the M-bus. If a bit of IRQC2M is 0, assertion of the corresponding MIRQ signal causes assertion of the corresponding CIRQ signal, provided that the corresponding FBICSR<IRQEN> bit is 1. This is used by modules that service interrupts *from* the M-bus.

FBICSR<23:20> IRQEN (RW) Interrupt-Request Enable

The IRQEN bits allow connection of the MIRQ/CIRQ signals. If a bit of IRQEN is 1, the FBIC electrically connects the corresponding CIRQ/MIRQ signals. If a bit of IRQEN is 0, the FBIC electrically isolates the corresponding CIRQ/MIRQ signals. Table 5-8 lists the correspondence between the IRQEN/IRQC2M bits and the CIRQ/MIRQ signals.

Table 5-8: FBICSR IRQEN/IRQC2M and CIRQ/MIRQ Correspondence

IRQEN	IRQC2M	CIRQ	MIRQ
FBICSR<20>	FBICSR<16>	CIRQ0	MIRQ0
FBICSR<21>	FBICSR<17>	CIRQ1	MIRQ1
FBICSR<22>	FBICSR<18>	CIRQ2	MIRQ2
FBICSR<23>	FBICSR<19>	CIRQ3	MIRQ3

FBICSR<24> RESET (RW) CVAX Pin-Bus RESET Control

The FBIC asserts the CRESET signal when RESET is 1. The M-bus portion of the FBIC is not affected by the state of the CRESET signal. The CVAX pin-bus portion of the FBIC returns to the idle state. The RESET bit is not automatically cleared; a 0 must be explicitly written to clear it.

Software must guarantee a 500-ns minimum pulse width on CRESET. Software must also guarantee that the M-bus is idle while asserting the RESET bit in a module, or unpredictable behavior will result. Because the RESET bit is not automatically cleared, a module may not reset itself.

FBICSR<25> HALTCPU (RW) CVAX Pin-Bus HALT Control

The FBIC asserts the CHALT signal when HALTCPU is 1 and the FBICSR<HALTEN> bit is 1.

FBICSR<26> EXCAEN (RW) External-Cache Enable

The FBIC enables its external cache when EXCAEN is 1.

FBICSR<27> CMISS (RW) CVAX Pin-Bus Cache Miss Occurred

The FBIC clears CMISS independent of any diagnostic modes when a CVAX pin-bus memory-space reference misses in the external cache. The CMISS bit is used in conjunction with the force-cache-hit diagnostic mode to test operation of the tag comparators.

FBICSR<31:30> MFMD (R) Manufacturing Mode

The MFMD bits reflect the value of the FBIC MANFMODE<1:0> input pins.

5.1.19.7. RANGE Register

The RANGE register is a read/write register that specifies an I/O-space address range that should be decoded for a given module. Note that it is possible to program the same address range to FBICs in different slots. This will result in MBRQ errors whenever M-bus cycles in this address range are attempted. Figure 5-12 shows the bit definitions for the RANGE register.

Name: RANGE Address: XXFFFFE4#16 Access: RW

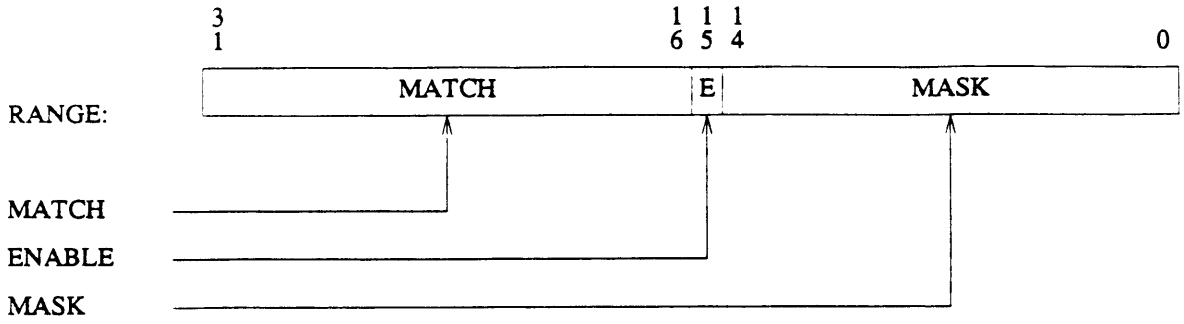


Figure 5-12: RANGE Register

RANGE<14:0> MASK (RW) M-Bus I/O-Space Address Range Mask

If a bit of the MASK field is 1, the corresponding bit of RANGE<MATCH> is a "don't care" bit. If a bit of the MASK field is 0, the corresponding bit of RANGE<MATCH> must be the same as the corresponding MDAL bit to generate a range match.

RANGE<15> ENABLE (RW) M-Bus I/O-Space Address-Range Enable

If the ENABLE bit is 1, the matches of the range decoder cause the FBIC to respond as an M-bus slave and forward the transaction into the CVAX pin-bus. If the ENABLE bit is 0, the range decoder never matches.

RANGE<31:16> MATCH (RW) M-Bus I/O-Space Address-Range Match

The MATCH bits specify the value of MDAL<31:16> to generate a range decoder *hit* for M-bus transaction addresses. The range-decoder function is (((MDAL<31:16> XOR MATCH) AND (NOT MASK)) EQL 0) AND ENABLE.

5.1.19.8. IPDVINT Register

The IPDVINT register is a read/write register that implements interprocessor and device interrupts. Interprocessor interrupts are into the local module, whereas device interrupts are onto the M-bus. The IPDVINT register can be used either as an interprocessor-interrupt unit or a device-interrupt unit, but not as both simultaneously. Figure 5-13 shows the bit definitions for the IPDVINT register.

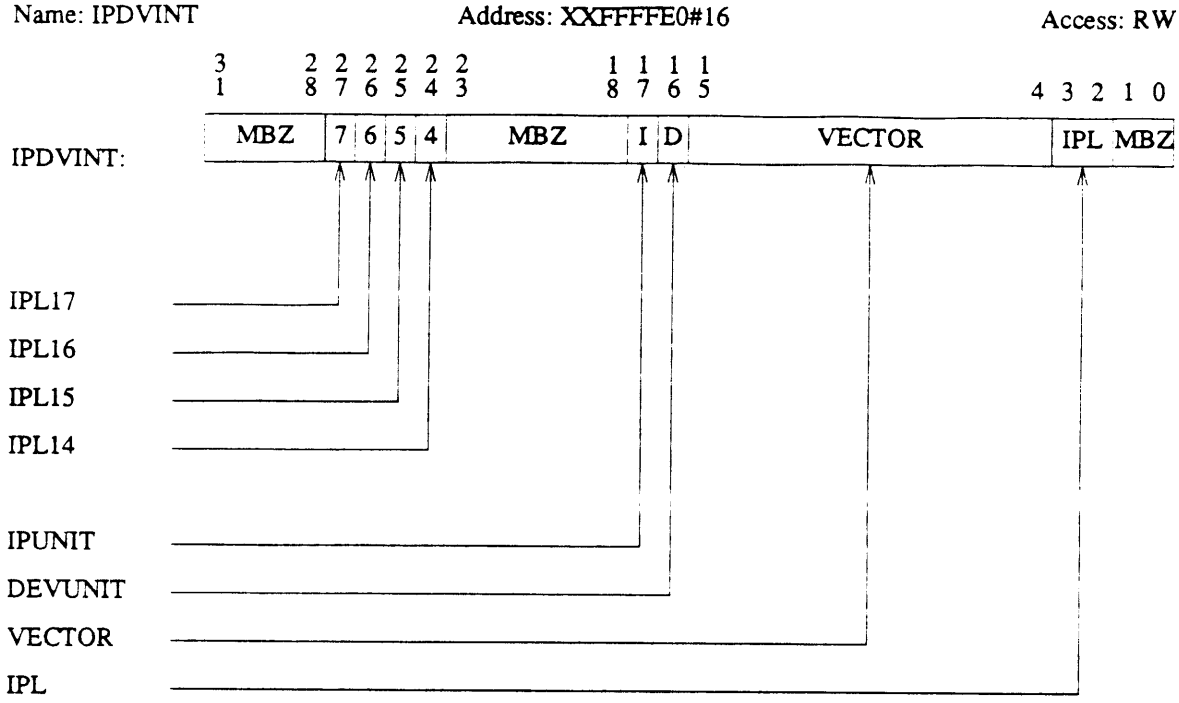


Figure 5-13: IPDVINT Register

IPDVINT<3:2> IPL (RW) Interrupt Level

The IPL field specifies the low two bits of the interrupt vector that the FBIC supplies when it responds to an interrupt acknowledge as an interprocessor or device-interrupt unit. When the FBIC returns a vector, the IPL field is set to the IPL level bits <3:2>. Table 5-9 lists the encoding for each IPL level.

Table 5-9: Encoding for the IPDVINT IPL Field

IPL	VECTOR<3:2>
14	0
15	1
16	2
17	3

When the FBIC returns the contents of the IPDVINT register during an I/O-space read, the IPL field is set to bits <3:2> of the register address (namely 00#2).

IPDVINT<15:4> VECTOR (RW) Interrupt Vector

The VECTOR field specifies the interrupt vector that the FBIC supplies when it responds to an interrupt acknowledge as an interprocessor or device-interrupt unit.

IPDVINT<17> DEVUNIT (RW) Device-Interrupt Unit

When the DEVUNIT bit is 1, the IPDVINT register functions as a device-interrupt unit. This means that interrupt requests are asserted on the MIRQ signals. Operation of the IPDVINT register is unpredictable if both IPUNIT and DEVUNIT are 1.

IPDVINT<16> IPUNIT (RW) Interprocessor-Interrupt Unit

When the IPUNIT bit is 1, the IPDVINT register functions as an interprocessor-interrupt unit. This means that interrupt requests are asserted on the CIRQ signals. Operation of the IPDVINT register is unpredictable if both IPUNIT and DEVUNIT are 1.

IPDVINT<24> IPL14 (RW) Generate IPL 14 Interrupt

When the IPL14 bit is 1, the IPDVINT register generates an IPL 14 interrupt. As a result, it asserts CIRQ0/MIRQ0. When the IPDVINT responds to an IPL 14 interrupt acknowledge, it clears the IPL14 bit. When the FBIC is in diagnostic-isolate test mode, the IPL14 bit is read/write. Otherwise, writing a 1 to IPL14 sets it and writing a 0 has no effect.

IPDVINT<25> IPL15 (RW) Generate IPL 15 Interrupt

When the IPL15 bit is 1, the IPDVINT register generates an IPL 15 interrupt. As a result, it asserts CIRQ1/MIRQ1. When the IPDVINT responds to an IPL 15 interrupt acknowledge, it clears the IPL15 bit. When the FBIC is in diagnostic-isolate-test mode, the IPL15 bit is read/write. Otherwise, writing a 1 to IPL15 sets it and writing a 0 has no effect.

IPDVINT<26> IPL16 (RW) Generate IPL 16 Interrupt

When the IPL16 bit is 1, the IPDVINT register generates an IPL 16 interrupt. As a result, it asserts CIRQ2/MIRQ2. When the IPDVINT responds to an IPL 16 interrupt acknowledge, it clears the IPL16 bit. When the FBIC is in diagnostic-isolate-test mode, the IPL16 bit is read/write. Otherwise, writing a 1 to IPL16 sets it and writing a 0 has no effect.

IPDVINT<27> IPL17 (RW) Generate IPL 17 Interrupt

When the IPL17 bit is 1, the IPDVINT register generates an IPL 17 interrupt. As a result, it asserts CIRQ3/MIRQ3. When the IPDVINT responds to an IPL 17 interrupt acknowledge, it clears the IPL17 bit. When the FBIC is in diagnostic-isolate-test-mode, the IPL17 bit is read/write. Otherwise, writing a 1 to IPL17 sets it and writing a 0 has no effect.

5.1.19.9. WHAMI Register

The WHAMI register is a read/write register for use by operating system software as a unique who-am-i identifier. The WHAMI register is also accessible to the local CPU as EPR 15. Figure 5-14 shows the bit definitions for the WHAMI register.

Name: WHAMI Address: XXFFFFDC#16 Access: RW

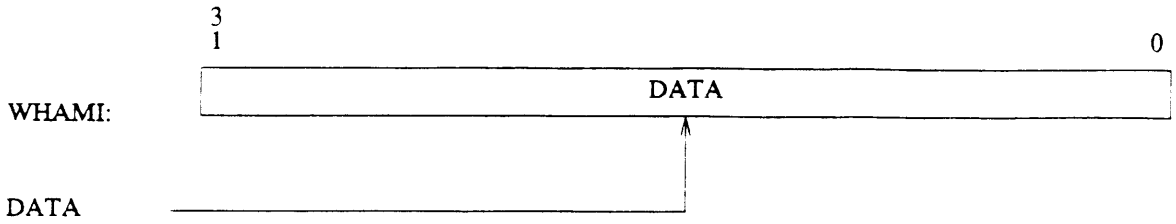


Figure 5-14: WHAMI Register

WHAMI<31:0> DATA (RW) Pointer to Processor-Specific Data

DATA interpretation is specific to the operating system.

5.1.19.10. CPUID Register

The CPUID register is a read-only register that uniquely identifies each processor in a Firefox workstation. It is also accessible to the local CPU as EPR 14. Figure 5-15 shows the bit definitions for the CPUID register.

Name: CPUID Address: XXFFFFD8#16 Access: R

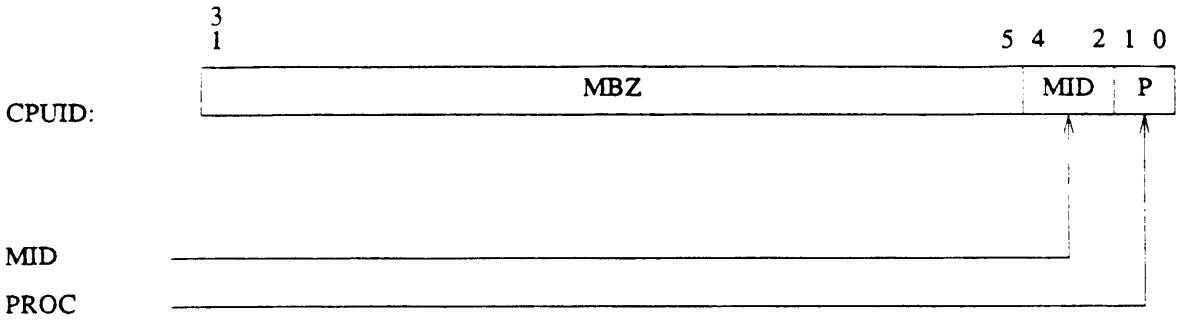


Figure 5-15: CPUID Register

CPUID<1:0> PROC (R) Processor Identifier

The PROC field uniquely identifies each processor for multiple-processor modules. It reflects the value of the FBIC TYPAGNTE input pin in both bits. If the TYPAGNTE pin is 0, the PROC field is 0; if the TYPAGNTE pin is 1, the PROC field is 3.

CPUID<4:2> MID (R) Module Slot Identifier

The MID field reflects the values of the FBIC MID input pins.

5.1.19.11. IADR1 Register

The IADR1 register is the first of the two units used for an interlocked sequence in progress. When one interlock is in progress, IADR1 is used to hold the hexaword interlock address. It is accessible as a register only for diagnostic purposes. IADR1 must not be written during normal system operation; if it is, system behavior will be unpredictable. Figure 5-16 shows the format of the IADR1 register.

Name: IADR1 Address: XXFFFFD4#16 Access: RW

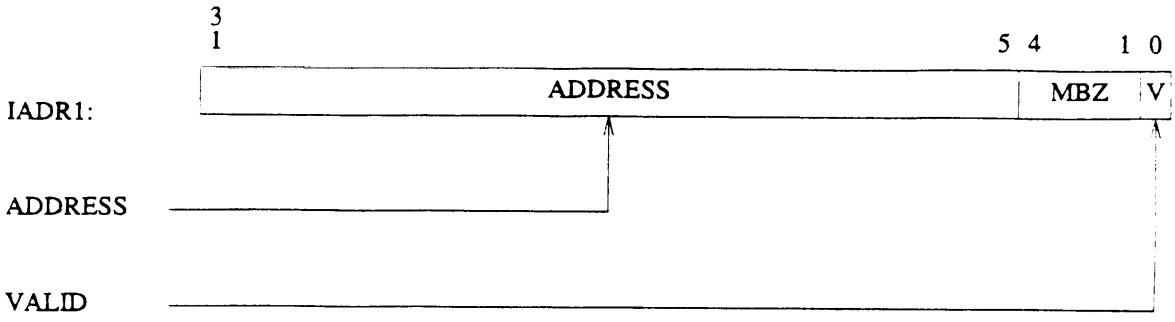


Figure 5-16: IADR1 Register

IADR1<0> VALID (RW) Interlock Register Valid

When VALID is 1, IADR1 contains the hexaword address of a locked region. The FBIC sets VALID when an interlocked read transaction is issued on the M-bus and VALID is not already set. The FBIC clears VALID when an unlock-write transaction for the hexaword specified by IADR1<ADDRESS> is issued on the M-bus. The FBIC also clears VALID when MABORT is asserted.

IADR1<31:5> ADDRESS (RW) Interlock Register Address

The ADDRESS field records the hexaword address of the currently interlocked region specified by IADR1.

5.1.19.12. IADR2 Register

The IADR2 register is the second of the two units used for an interlocked sequence in progress. When one interlock is already in progress, IADR2 is used to hold the hexaword interlock address of the second interlock. It is accessible as a register only for diagnostic purposes. IADR2 must not be written during normal system operation; if it is, system behavior will be unpredictable. Figure 5-17 shows the format of the IADR2 register.

Name: IADR2 Address: XXFFFFD0#16 Access: RW

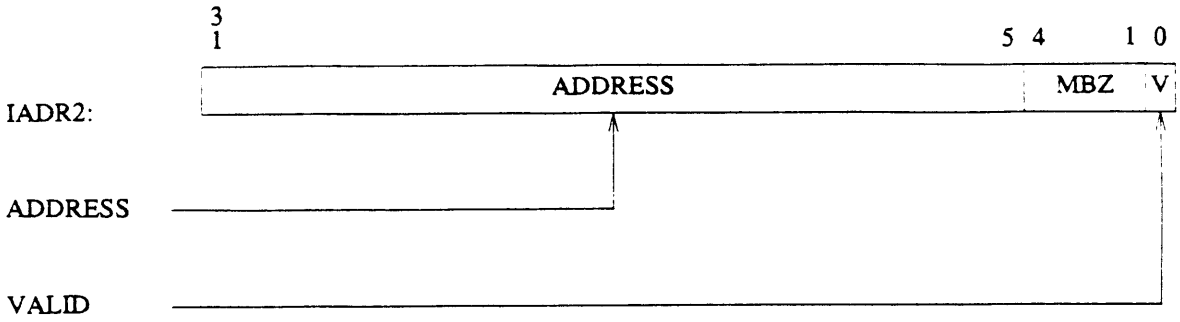


Figure 5-17: IADR2 Register

IADR2<0> VALID (RW) Interlock Register Valid

When VALID is 1, IADR2 contains the hexaword address of a locked region. The FBIC sets VALID when an interlocked read transaction is issued on the M-bus and IADR1<VALID> is set. The FBIC clears VALID when an unlock-write transaction for the hexaword specified by IADR2<ADDRESS> is issued on the M-bus. The FBIC also clears VALID when MABORT is asserted.

IADR2<31:5> ADDRESS (RW) Interlock Register Address

The ADDRESS field records the hexaword address of the currently interlocked region specified by IADR2.

5.1.19.13. SAVGPR Register

The SAVGPR register is a read/write, temporary-storage register for use by console software during processor HALT handling. The SAVGPR register is also accessible to the local CPU as EPR 41. Figure 5-18 shows the bit definitions for the SAVGPR register.

Name: SAVGPR Address: XXFFFFC4#16 Access: RW

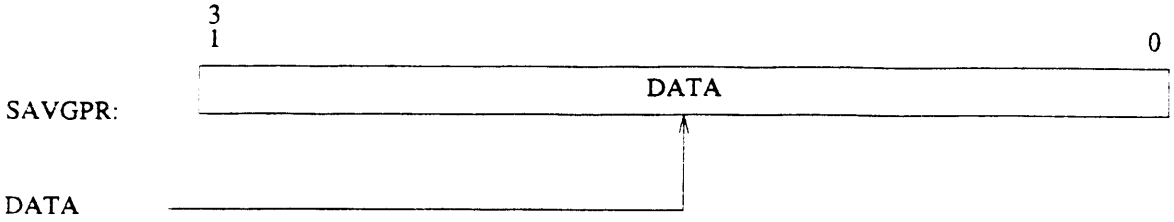


Figure 5-18: SAVGPR Register

SAVGPR<31:0> DATA (RW) Save General-Purpose Register

DATA use is specific to the console software.

5.2. Interface

The FBIC has four groups of signals that implement the CVAX pin-bus interface, the M-bus interface, the external-cache interface, and miscellaneous external logic and datapath control. Table 5-10 lists the FBIC pin signals and their functions. The signal type "I" refers to an input with respect to the FBIC. The signal type "O" refers to an output with respect to the FBIC. The signal type "I/O" refers to both.

Table 5-10: FBIC Pinout

Group	Signal	Assert	Count	Type	Function
CBUS	CDAL	H	32	I/O	Data and address
	CCSDP	L	4	I/O	Cycle status/data parity
	CDPE	L	1	I/O	Data-parity enable
	CAS	L	1	I/O	Address strobe
	CDS	L	1	I/O	Data strobe
	CBM	L	4	I/O	Byte mask
	CWR	L	1	I/O	Write/read
	CRDY	L	1	I/O	Ready
	CERR	L	1	I/O	Error
	CCCTL	L	1	O	Cache control
	CDMR	L	1	I/O	DMA request
	CDMG	L	1	I/O	DMA grant
	CRESET	L	1	I	Synchronous RESET
	SYSRESET	L	1	O	Asynchronous RESET
	CHALT	L	1	O	HALT
	CIRQ	L	4	I/O	Interrupt request
	CRD	L	1	O	Corrected read data
	MEMERR	L	1	O	Memory error
	CCLKA	H	1	I	Clock A
	CCLKB	H	1	I	Clock B
	CCLKC	H	1	I	Clock C
			61	**Subtotal**	
MBUS	MBRM	L	7	I	Module M-bus requests
	MBRP	L	1	I	Partner M-bus request
	MYMBRQ	L	1	O	Local M-bus request
	MBUSYI	L	1	I	Slave-busy input
	MBUSYO	L	1	0	Slave-busy output
	MCMD	H	4	I/O	M-bus cycle command
	MSTATUS	H	2	I/O	M-bus cycle status
	MDAL	H	32	I/O	Data and address
	MCPAR	H	1	I/O	MCMD parity
	MSPAR	H	1	I/O	MSTATUS parity
	MDPAR	H	1	I/O	MDAL parity
	MCDRV	L	1	O	MCMD xcvr direction
	MSDRV	L	1	O	MSTATUS xcvr direction
	MDDRV	L	1	O	MDAT xcvr direction
	MSHAREDI	L	1	I	Shared line input
	MSHAREDO	L	1	0	Shared line output
	MDATINVI	L	1	I	Data-invalid input
	MDATINVO	L	1	0	Data-invalid output
	MID	H	3	I	Module ID
	MRESET	L	1	I	System reset
	MABORTI	L	1	I	Transaction-abort input
	MABORTO	L	1	0	Transaction-abort output
	MIRQI	L	4	I	Interrupt requests input
	MIRQO	L	4	O	Interrupt requests output
	MHALT	L	1	I	Halt CPU input
	MCLKA	H	1	I	M-bus clock-A phase
	MCLKB	H	1	I	M-bus clock-B phase
			76	**Subtotal**	

Group	Signal	Assert	Count	Type	Function
CACHECTRL	TCACHE	H	13	I/O	Tag cache
	TAGSH	H	1	I/O	Tag shared signal
	TAGDR	H	1	I/O	Tag dirty signal
	TAGPAR	H	1	I/O	Tag parity
	TAGWE	L	1	O	Tag-cache write enable
	TAGCE	L	1	O	Tag-cache chip enable
	DATCE	L	4	O	Data-cache chip enable
	DATWE	L	1	O	Data-cache write enable
	XOE		L	1	O
	ECL		H	1	O
	CTINDX_OE	L	1	O	CVAX pin-bus tag-index output enable
	MTINDX_LE	H	1	O	M-bus tag-index latch enable
	MTINDX_OE	L	1	O	M-bus tag-index output enable
			28	**Subtotal**	
MISC	MODCL	H	2	I	Module class
	TYPDUAL	H	1	I	Dual module
	TYPAGNTE	H	1	I	A processor/grantee
	TYPRET	H	1	I	Retriable
	TYPSYNC	H	1	I	Synchronous bus
	DEVIRQ	L	4	I	Device-interrupt requests
	ROMOE	L	1	O	External-ROM output enable
	ROMWID32	H	1	I	External-ROM width 16/32
	ROMWADDR	H	1	O	External-ROM word address
	MNFMOD	L	2	I	Manufacturing mode
	LEDS	L	6	O	LED value
	TESTOUT	H	1	O	Test output
	TRISTATE	H	1	I	Tristate all FBIC pins
			23	**Subtotal**	
SIGNALS			188	**Signal pins**	
RESERVED			7	**Spare pins**	
VDD			13	**Supply pins**	
VSS			15	Ground pins	
			223	**Total**	

In the following pages, the four groups of FBIC pins are described in detail.

5.2.0.1. CVAX Pin-Bus Pinout Group

The CVAX pin-bus group consists of those FBIC pins that are connected to the local CVAX pin-bus. The signal descriptions are from the perspective of the FBIC. Unless otherwise noted, signal descriptions apply to FBIC operation as both a CVAX pin-bus master and a CVAX pin-bus slave. Unless otherwise noted, signal descriptions apply to both synchronous and asynchronous CVAX pin-busses. The pins that make up this group are described here.

5.2.0.1.1. CDAL<31:0>–Data/Address Lines

The CDAL is a 32-bit, time-multiplexed bus used to transfer all data and address information on the CVAX pin-bus. The CVAX pin-bus master drives CDAL with an address around the falling edge of CAS. The CVAX pin-bus slave drives CDAL with data for memory-space reads, I/O-space reads, EPR reads, and interrupt acknowledges until the rising edge of CDS. The CVAX pin-bus master drives CDAL with data

for memory-space writes, I/O-space writes, and EPR writes until the rising edge of CDS.

For memory-space addresses, CDAL<31:30> indicates the length of the transfer as shown in Table 5-11. CDAL<29> is 0, CDAL<28:2> specifies the longword address of the operand, and CDAL<1:0> is undefined. Quadword, hexword, and octaword transfers are all within the same naturally aligned region for the transfer size. Table 5-12 shows the implied sequencing of CDAL<3:2> for such references.

Table 5-11: Transfer-Length Encoding

CDAL<31:30>	Length
00	3 longwords
01	1 longword
10	2 longwords
11	4 longwords

Table 5-12: Quad/Hex/Octaword Implied Address Sequencing

CDAL<3:2>	Second Address	Third Address	Fourth Address
00	01	10	11
01	00	11	10
10	11	00	01
11	10	01	00

For I/O-space addresses, CDAL<31:30> indicates the length of the transfer, which must be one longword or behavior will be unpredictable. CDAL<29> is 1, CDAL<28:2> specifies the longword address of the operand, and CDAL<1:0> is undefined.

For EPR addresses, CDAL<31:8> is undefined, CDAL<7:2> specifies the register, and CDAL<1:0> is undefined.

For interrupt-acknowledge addresses, CDAL<31:7> is undefined, CDAL<6:2> specifies the IPL of the interrupt being acknowledged, and CDAL<1:0> is undefined.

For memory-space, I/O-space, and EPR reads, CDAL<31:0> specifies the data. For interrupt acknowledges, CDAL<31:16> is undefined and CDAL<15:0> specifies the vector. For memory-space, I/O-space, and EPR writes, CDAL<31:0> specifies the data.

5.2.0.1.2. CCSDP<3:0>—Cycle Status/Data Parity

During the address portion of a CVAX pin-bus transaction, the cycle-status lines, in conjunction with the CWR signal, characterize the type of cycle occurring on the CVAX pin-bus. Table 5-13 shows the specific encoding of the pins around the falling edge of CAS.

Table 5-13: Encoding for the CCSDP Cycle Type

CWR	CCSDP<2:0>	CVAX Pin-Bus Cycle Type	FBIC Cycle Type
1	000	Request D-stream read	Read
1	001	Reserved	UPREDICTABLE
1	010	EPR read	EPR read
1	011	Interrupt acknowledge	Interrupt acknowledge
1	100	Request I-stream read	Read
1	101	Demand D-stream read (lock)	Read interlocked
1	110	Demand D-stream read with modify intent	Read
1	111	Demand D-stream read (no lock or modify intent)	Read
0	000	Reserved	UNPREDICTABLE
0	001	Reserved	UNPREDICTABLE
0	010	EPR write	EPR write
0	011	Reserved for DMA-device use	UNPREDICTABLE
0	100	Reserved	UNPREDICTABLE
0	101	Write unlock	Write unlock
0	110	Reserved	UNPREDICTABLE
0	111	Write no unlock	Write

As a CVAX pin-bus master, the FBIC generates only transactions involving demand D-stream read, demand D-stream read interlocked, write, write unlock, and interrupt acknowledge.

During the data portion of CVAX pin-bus transactions, CCSDP<3:0> specifies byte parity for CDAL. Even parity is checked/generated on even bytes; odd parity, on odd bytes. Even parity will drive a 0 when there are an even number of 1s in the byte's data; odd parity will drive a 0 for an odd number of 1s. Table 5-14 lists the correspondence of CCSDP signals and CDAL bytes. If CDPE is asserted, the device that sources data onto CDAL is generating parity for all bytes and the device that sinks data from CDAL can check parity for the bytes specified by CBM<3:0>.

The FBIC always generates parity when it drives the CDAL with data. If it is supplying data it obtained from the M-bus with MDATINV asserted, the FBIC intentionally generates invalid parity on the CVAX pin-bus. Otherwise, it generates valid parity on the CVAX pin-bus. It checks parity when it receives data from CDAL and CDPE is asserted.

Table 5-14: Byte Correspondence Between CCSDP and CDAL

CCSDP	CDAL
<3>	<31:24>
<2>	<23:16>
<1>	<15:08>
<0>	<07:00>

5.2.0.1.3. CDPE<0>--Data-Parity Enable

When CDPE is asserted, the device that sinks data from CDAL/CCSDP should check parity. When CDPE is deasserted, the device that sinks data from CDAL/CCSDP must not check parity.

The FBIC asserts the CDPE signal whenever the FBICSR<CDPE> bit is asserted and either it or the external data store is driving data onto the CVAX pin-bus, except during CVAX ROM reads.

The CDPE signal requires an external pull-up resistor (nominally 600 ohms).

5.2.0.1.4. CAS<0>--Address Strobe

A CVAX pin-bus master asserts CAS when the CDAL, CCSDP, and CWR signals contain valid information during the address portion of a CVAX pin-bus transaction. A CVAX pin-bus slave latches these signals at that time, and it interprets them as an address and transaction specifier.

The CVAX pin-bus master continues to assert CAS until the transaction ends. The rising edge of CAS always terminates a CVAX pin-bus transaction, whether CRDY/CERR has been asserted or not.

When the FBIC generates octaword reads and writes on the CVAX pin-bus to maintain the external-cache data store, it does not assert the CRDY/CERR signals. Nor does it assert the CRDY/CERR signals when it generates I/O reads on the CVAX pin-bus to access the ROM.

The CAS signal requires an external pull-up resistor (nominally 2K ohms).

5.2.0.1.5. CDS<0>--Data Strobe

The CDS signal provides timing information for asynchronous data transfers on the CVAX pin-bus. During a memory read (single or multiple transfer), I/O read, EPR-read, or interrupt acknowledge, the CVAX pin-bus master asserts CDS to indicate that CDAL<31:0> and CCSDP<3:0> are free to receive incoming data and deasserts CDS to indicate that it has received and latched the incoming data. During a memory write, I/O write, or EPR write, the CVAX pin-bus master asserts CDS to indicate that CDAL<31:0> and CS/DP<3:0> contain valid outgoing data and deasserts CDS to indicate that the data is about to be removed.

When operating as a synchronous CVAX pin-bus slave, the FBIC does not monitor CDS; data transfers always occur during successive CVAX pin-bus cycles.

The CDS signal requires an external pull-up resistor (nominally 2K ohms).

5.2.0.1.6. CBM<3:0>--Byte Mask

The CBM signals specify which bytes of the CDAL bus contain valid information during the data portion of a CVAX pin-bus cycle. The CVAX pin-bus master supplies byte masks for each longword of quad/hex/octaword transfers. Table 5-15 lists the correspondence between CBM signals and CDAL bytes.

Table 5-15: Byte Correspondence Between CBM and CDAL

CBM	CDAL
<3>	<31:24>
<2>	<23:16>
<1>	<15:08>
<0>	<07:00>

When the FBIC is a CVAX pin-bus slave for memory writes, I/O writes, and EPR writes, it supports all possible byte masks. As a CVAX pin-bus slave for memory reads, I/O reads, EPR reads, and interrupt acknowledges, it ignores the byte masks.

To control which bytes are written with M-bus shared write-through data, the FBIC uses the CBM signals during octaword writes to the external-cache data store.

5.2.0.1.7. CWR<0>--Write

The CWR signal specifies the direction of data transfer on the CDAL bus. If CWR is asserted, the CVAX pin-bus master is driving data onto the CDAL. If it is deasserted, the CVAX pin-bus master is receiving data from the CDAL. CWR is valid around the falling edge of CAS.

CWR must be stable throughout the assertion of CAS or behavior is unpredictable.

5.2.0.1.8. CRDY<0>--Ready

The CVAX pin-bus slave asserts the CRDY signal to indicate normal termination of the current CVAX pin-bus transaction. During a CVAX pin-bus memory-read, I/O-read, EPR read, or interrupt-acknowledge transaction, CRDY indicates that the CVAX pin-bus slave has placed the requested data on the CDAL bus in time for the next sampling point. During a CVAX pin-bus memory-write, I/O-write, or EPR-write transaction, CRDY indicates that the information on the CDAL bus has been received and can be removed following the next sampling point. Upon assertion of CRDY, the CVAX pin-bus master terminates the current CVAX pin-bus transaction, and the CVAX pin-bus slave deasserts CRDY.

The CRDY signal requires an external pull-up resistor (nominally 600 ohms).

5.2.0.1.9. CERR<0>--Error

The CVAX pin-bus slave asserts the CERR signal to indicate abnormal termination of the current CVAX pin-bus transaction. The interpretation of CERR depends on whether CRDY is also asserted. When CERR and CRDY are asserted simultaneously, the CVAX pin-bus master will *retry* the current bus cycle. When CERR is asserted and CRDY deasserted, the CVAX pin-bus master will *abort* the current bus cycle. To eliminate timing hazards associated with CERR and CRDY synchronizers, the CVAX pin-bus master interprets as a retry, assertion of CERR followed by assertion of CERR and CRDY.

If retries are specified for the second, third, or fourth longword of a quad/hex/octaword memory transaction, behavior is unpredictable.

The CERR signal requires an external pull-up resistor (nominally 600 ohms).

5.2.0.1.10. CCCTL<0>--Cache Control

The CCCTL signal can be used to generate CVAX cache invalidates and to suppress CVAX data caching. The FBIC always drives the CCCTL signal, even if it is not the current CVAX pin-bus master. It uses CCCTL only to generate the CVAX octaword cache invalidates as required to maintain the external-cache data store.

The CCCTL signal requires an external pull-up resistor (nominally 600 ohms).

5.2.0.1.11. CDMR<0>--DMA Request

The CDMR signal is asserted by a CVAX pin-bus slave when it wishes to take control of the CDAL bus and related control signals for DMA or other purposes. When the CVAX pin-bus master observes CDMR asserted, it completes the current CVAX pin-bus transaction, tristates the CVAX pin-bus control and data signals, and asserts CDMG. When the CVAX pin-bus master observes CDMR deasserted, it deasserts CDMG and resumes driving the CVAX pin-bus control and data signals.

If the FBIC is the CVAX pin-bus grantor, it monitors CDMR and asserts CDMG when it relinquishes the CVAX pin-bus. As CVAX pin-bus grantee, it asserts CDMR when it requires the CVAX pin-bus and monitors CDMG. It determines its grantor/grantee role from the TYPDUAL/TYPAGNTE input pins.

The CDMR signal requires an external pull-up resistor (nominally 600 ohms).

5.2.0.1.12. CDMG<0>--DMA Grant

The CDMG signal is asserted by the CVAX pin-bus master to grant to a CVAX pin-bus slave control of both the CDAL bus and related control signals. The CVAX pin-bus master tristates the CDAL, CAS, CDS, CBM, CDPE, CCSDP, and CWR signals. When the CVAX pin-bus slave deasserts CDMR, the CVAX pin-bus master responds by deasserting CDMG and starting the next bus cycle.

If the FBIC is the CVAX pin-bus grantor, it monitors CDMR and asserts CDMG when it relinquishes the CVAX pin-bus. As CVAX pin-bus grantee, it asserts CDMR when it requires the CVAX pin-bus and

monitors CDMG. The FBIC determines its grantor/grantee role from the TYPDUAL/TYPAGNTE input pins.

5.2.0.1.13. CRESET<0>--Synchronous RESET

When CRESET is asserted, all CVAX pin-bus devices initialize their internal logic and return to an idle state. While CRESET is asserted, the CVAX pin-bus default master tristates the CDAL, CDPE, CBM, CWR, CCSDP, CAS, and CDS signals.

5.2.0.1.14. SYSRESET<0>--Asynchronous RESET

When SYSRESET is asserted, module logic synchronizes it to generate the CRESET signal. The FBIC asserts SYSRESET when FBICSR<RESET> or MRESET is asserted.

The SYSRESET signal requires an external pull-up resistor (nominally 600 ohms).

5.2.0.1.15. CHALT<0>--HALT

When the CHALT signal makes a deasserted-to-asserted transaction, CVAX processors initiate a HALT. The FBIC asserts CHALT when FBICSR<HALTEN> AND (FBICSR<HALTCPU> OR MHALT) is true.

The CHALT signal requires an external pull-up resistor (nominally 600 ohms).

5.2.0.1.16. CIRQ<3:0>--Interrupt Requests

The CIRQ signals specify an interrupt request. Table 5-16 lists the correspondence of CIRQ signals to interrupt priority levels. The CIRQ signals are level-sensitive. The FBIC can be programmed to forward assertions of CIRQ signals onto MIRQ signals (for non-CPU modules) or assertions of MIRQ signals onto CIRQ signals (for CPU modules).

Table 5-16: CIRQ Interrupt Priority Levels

CIRQ	IPL
<3>	17
<2>	16
<1>	15
<0>	14

The CIRQ signals requires external pull-up resistors (nominally 600 ohms).

5.2.0.1.17. CRD<0>--Corrected Read Data

The CRD signal allows a CVAX pin-bus slave to indicate that data supplied to the CVAX pin-bus had a corrected single-bit ECC error. In the CVAX, CRD interrupts at IPL 1A (SCB vector 54#16). A corrected read-data interrupt is not acknowledged by the CVAX.

The FBIC asserts CRD when it obtains corrected data from an M-bus memory read.

The CRD signal requires an external pull-up resistor (nominally 600 ohms).

5.2.0.1.18. MEMERR<0>--Memory Error

The MEMERR signal allows the CVAX pin-bus slave to signal a memory/M-bus error to the CVAX pin-bus. In the CVAX, MEMERR interrupts at IPL1D (SCB vector 60#16). A MEMERR interrupt is not acknowledged by the CVAX.

The FBIC asserts MEMERR when it supplies data with a parity error onto the M-bus or when a M-bus abort occurs.

The MEMERR signal requires an external pull-up resistor (nominally 600 ohms).

5.2.0.1.19. CCLKA<0>, CCLKB<0>, CCLKC<0>--Clocks A, B, and C

CCLKA, CCLKB, and CCLKC are the clocks for the CVAX pin-bus. CCLKA and CCLKB are square-wave signals that are phase-shifted 180 degrees. CCLKC is a square-wave signal that is inphase but half the frequency of CCLKA. It is used to distinguish CVAX pin-bus phase P1 from CVAX pin-bus phase P3. The FBIC supports a CVAX pin-bus cycle time of 70 to 100 ns for modules that do not have an external cache and a CVAX pin-bus cycle time of 70 to 80 ns for modules that do have an external cache.

5.2.0.2. M-Bus Pinout Group

The M-bus pinout group consists of those FBIC pins that are connected to the M-bus, either directly or through external transceivers/buffers. Operation of the M-bus interface is as defined by the *Firefox M-Bus Specification*. A detailed description of the M-bus signals can be found in that document. The pins themselves are described here.

5.2.0.2.1. MBRM<6:0>--Request Monitor

The M-bus MBRM signals are the M-bus requests from the other backplane slots. The FBIC MBRM signals connect directly to the M-bus MBRM signals for the backplane slot.

5.2.0.2.2. MBRP<0>--Partner Request Monitor

The FBIC MBRP signal connects to the MYMBRQ output of the other FBIC on a dual-FBIC module. It is used to resolve intramodule M-bus arbitration in the same fashion that the M-bus MBRQ signals resolve intermodule M-bus arbitration.

5.2.0.2.3. MYMBRQ<0>--Request Output

The M-bus MBRQ signal indicates that a module is in arbitration for the M-bus, or that it is driving the M-bus as master/slave. The FBIC MYMBRQ signal drives the M-bus MBRQ signal for the backplane slot through a 74F244-class buffer. To form the module MBRQ signal on dual-FBIC modules, the two FBIC MYMBRQ signals must be logically ORed together with external logic.

5.2.0.2.4. MBUSYI<0>/MBUSYO<0>--MBUSY Input/Output

When asserted, the M-bus MBUSY signal stalls commencement of new M-bus transactions. The FBIC MBUSYI signal connects directly to the M-bus MBUSY signal. The FBIC MBUSYO signal drives the M-bus MBUSY signal through a 74AS760-class open-collector buffer.

5.2.0.2.5. MCMD<3:0>--Transaction Command

The M-bus MCMD signals specify transaction type, memory-write byte masks, and I/O byte masks from M-bus masters. The FBIC MCMD signals connect to the M-bus MCMD signals through an external 74F245 transceiver. By default, the FBIC MCMD signals are inputs so two FBICs can share an M-bus transceiver.

5.2.0.2.6. MSTATUS<1:0>--Transaction Status

The M-bus MSTATUS signals specify transaction status from M-bus slaves. The FBIC MSTATUS signals connect to the M-bus MSTATUS signals through an external 74F245 transceiver. By default, the FBIC MSTATUS signals are inputs so two FBICs can share an M-bus transceiver.

5.2.0.2.7. MDAL<31:0>--Transaction Data/Address

The M-bus MDAL signals specify M-bus transaction addresses and data. The FBIC MDAL signals connect to the M-bus MDAL signals through external 74F245 transceivers. By default, the FBIC MDAL signals are inputs so two FBICs can share a set of M-bus transceivers.

5.2.0.2.8. MCPAR<0>--Transaction Command Parity

The M-bus MCPAR signal specifies even parity for the M-bus MCMD signals. The FBIC MCPAR signal connects to the M-bus MCPAR signal through an external 74F245 transceiver. By default, the FBIC MCPAR signal is an input so two FBICs can share an M-bus transceiver.

5.2.0.2.9. MSPAR<0>--Transaction Status Parity

The M-bus MSPAR signal specifies even parity for the M-bus MSTATUS signals. The FBIC MSPAR signal connects to the M-bus MSPAR signal through an external 74F245 transceiver. By default, the FBIC MSPAR signal is an input so two FBICs can share an M-bus transceiver.

5.2.0.2.10. MDPAR<0>--Transaction Data/Address Parity

The M-bus MDPAR signal specifies even parity for the M-bus MDAL signals. The FBIC MDPAR signal connects to the M-bus MDPAR signal through an external 74F245 transceiver. By default, the FBIC MDPAR signal is an input so two FBICs can share an M-bus transceiver.

5.2.0.2.11. MCDRV<0>--Transaction Command Drive

The FBIC MCDRV signal controls the direction of the external 74F245 transceivers for the FBIC MCMD/MCPAR signals. To form the module MCDRV signal on dual-FBIC modules, the two FBIC MCDRV signals must be logically ORed together with external logic.

5.2.0.2.12. MSDRV<0>--Transaction Status Drive

The FBIC MSDRV signal controls the direction of the external 74F245 transceivers for the FBIC MSTATUS/MSPAR signals. To form the module MSDRV signal on dual-FBIC modules, the two FBIC MSDRV signals must be logically ORed together with external logic.

5.2.0.2.13. MDDRV<0>--Transaction Data/Address Drive

The FBIC MDDRV signal controls the direction of the external 74F245 transceivers for the FBIC MDAL/MDPAR signals. To form the module MDDRV signal on dual-FBIC modules, the two FBIC MDDRV signals must be logically ORed together with external logic.

5.2.0.2.14. MSHAREDI<0>/MSHAREDO<0>--MSHARED Input/Output

The M-bus MSHARED signal indicates that the memory octaword referenced by the current transaction is shared. The FBIC MSHAREDI signal connects directly to the M-bus MSHARED signal. The FBIC MSHAREDO signal drives the M-bus MSHARED signal through a 74AS760-class open-collector buffer.

5.2.0.2.15. MDATINVI<0>/MDATINVO<0>--MDATINV Input/Output

The M-bus MDATINV signal indicates that the data on MDAL had an internal module-parity error. The FBIC MDATINVI signal connects directly to the M-bus MDATINV signal. The FBIC MDATINVO signal drives the M-bus MDATINV signal through a 74AS760-class open-collector buffer.

5.2.0.2.16. MID<2:0>--Module ID

The MID signals uniquely identify each M-bus backplane slot with a value from 0 to 7. The FBIC MID signals connect to the module M-bus MID signals through 47-ohm series resistors.

5.2.0.2.17. MRESET<0>--System Reset

When the MRESET signal is asserted, the entire workstation is reinitialized. The FBIC MRESET signal connects directly to the M-bus MRESET signal.

5.2.0.2.18. MABORTI<0>/MABORTO<0>--MABORT Input/Output

The M-bus MABORT signal indicates that an error has occurred on the M-bus. The FBIC MABORTI signal connects directly to the M-bus MABORT signal. The FBIC MABORTO signal drives the M-bus MABORT signal through a 74AS760-class open-collector buffer.

5.2.0.2.19. MIRQI<3:0>/MIRQO<3:0>--Interrupt Requests

The M-bus MIRQ signals are asserted to indicate a pending interrupt. The FBIC MIRQI signals connect directly to the M-bus MIRQ signals. The FBIC MIRQO signals drive the M-bus MIRQ signals through a 74AS760-class open-collector buffer.

5.2.0.2.20. MHALT<0>--Processor Halt

The M-bus MHALT signal is asserted to halt all processors. The FBIC MHALT signal connects directly to the M-bus MHALT signal.

5.2.0.2.21. MCLKA<0>--Clock-A Phase

MCLKA is the master clock for the M-bus. The FBIC MCLKA signal connects directly to the M-bus MCLKA signal for the backplane slot.

5.2.0.2.22. MCLKB<0>--Clock-B Phase

MCLKA is the slave clock for the M-bus. The FBIC MCLKB signal connects directly to the M-bus MCLKB signal for the backplane slot.

5.2.0.3. Cache-Control Pinout Group

The cache-control pinout group consists of those FBIC pins that control and transfer data to the external-cache tag and data stores. The pins that make up this group are described here.

5.2.0.3.1. TCACHE<12:0>--Tag Cache

The TCACHE signals transfer 13 bits of data between the FBIC and an external-cache tag store. The FBIC has exclusive control over these pins and uses them to read and write the tag store in order to perform tag compares and update the cache entry. The TCACHE signals are the high-order 13 bits of the CVAX pin-bus physical address, namely CDAL<28:16>.

5.2.0.3.2. TAGSH<0>--Tag Shared

The TAGSH signal transfers the *shared* bit between the FBIC and the external-cache tag store.

5.2.0.3.3. TAGDR<0>--Tag Dirty

The TAGDR signal transfers the *dirty* bit between the FBIC and the external-cache tag store.

5.2.0.3.4. TAGPAR<0>--Tag-Cache Parity

The TAGPAR signal transfers 1 bit of parity for the 15-bit external-cache tag-store entry. The tag entry is composed of a 13-bit address, a shared bit, and a dirty bit, as described earlier. Even parity is checked/generated for all 15 bits.

5.2.0.3.5. TAGWE<0>--Tag-Cache Write Enable

The TAGWE signal controls modification of the external-cache tag store. When TAGWE is a 0, the external tag-store RAMs are in write mode. When TAGWE is a 1, the external tag-store RAMs are in read mode.

5.2.0.3.6. TAGCE<0>--Tag-Cache Chip Enable

The TAGCE signal controls access to the external-cache tag store. When TAGCE is a 0, the external tag-store RAMs are enabled. When TAGCE is a 1, the external tag-store RAMs are disabled.

5.2.0.3.7. DATCE<3:0>--Data-Cache Chip Enable

The DATCE signals control access of individual bytes in the longword external-cache data store. If DATCE<3> is asserted, the RAMs transfer data to/from CDAL<31:24>. If DATCE<2> is asserted, the RAMs transfer data to/from CDAL<23:16>. If DATCE<1> is asserted, the RAMs transfer data to/from CDAL<15:8>. If DATCE<0> is asserted, the RAMs transfer data to/from CDAL<7:0>.

5.2.0.3.8. DATWE<0>--Data-Cache Write Enable

The DATWE signal controls direction of the external-cache data-store transceivers and the write enable to the RAMs. When DATWE is a 1, data is driven from the CVAX pin-bus into the RAMs. When DATWE is a 0, data is driven from the RAMs onto the CVAX pin-bus.

5.2.0.3.9. XOE<0>--Data-Cache Transceiver Output Enable

The XOE signal controls the output enable of the external-cache data-store transceivers. When XOE is a 1, the transceivers are tristated. When XOE is a 0, the transceivers drive data onto the CVAX pin-bus or toward the external-cache data store RAMs depending on the value of DATWE.

5.2.0.3.10. ECL<0>--Data-Cache Counter Enable

The ECL signal gates the clock signal to the two-bit, external-cache address latch fed by the external-cache address counter. When ECL is a 1, the clock signal should be enabled to the latch. When ECL is a 0, the clock signal should be disabled from reaching the latch, and the transparent latch should remain closed. This signal allows tag probes from the M-bus to preempt an ongoing CVAX pin-bus cycle that may need to update the external-cache tag-store.

5.2.0.3.11. CTINDX_OE<0>--CVAX Pin-Bus Tag-Index Output Enable

The CTINDX_OE signal enables the CVAX pin-bus address latch for the external-cache tag store. The latch captures the middle 12 bits of an octaword address, namely CDAL<15:4>. The CTINDX_OE signal connects directly to the CVAX pin-bus address-latch output-enable pin.

5.2.0.3.12. MTINDX_LE<0>--M-Bus Tag-Index Latch Enable

The MTINDX_LE signal is a latch-enable signal for the external latch that indexes into the external tag store from the M-bus. The latch captures the middle 12 bits of an M-bus octaword address, namely MDAL<15:4>. When MTINDX_LE and MCLKB are asserted, the latch should be transparent.

5.2.0.3.13. MTINDX_OE<0>--M-Bus Tag-Index Output Enable

The MTINDX_OE signal enables the M-bus address latch for the external-cache tag store. The latch captures the middle 12 bits of an octaword address, namely MDAL<15:4>. The MTINDX_OE signal connects directly to the M-bus address-latch output-enable pin.

5.2.0.4. Miscellaneous Pinout Group

The miscellaneous pinout group consists of all FBIC pins that do not fall within one of the other three groups. They are described here.

5.2.0.4.1. MODCL<1:0>--Module Class

These input-only signals indicate to the FBIC the class of module on which the chip is physically located. To be able to identify the correct module to the FBIC, these signals must be tied to power and ground through resistors in the manner shown in Table 5-17.

Table 5-17: M-Bus Defined Module Classes

MODCL	Class
00	Bus adapter
01	Graphics
10	I/O
11	CPU

5.2.0.4.2. TYPDUAL<0>--Dual-FBIC Module

The TYPDUAL input pin indicates whether the module has one or two FBICs. If TYPDUAL is deasserted, the FBIC is the only one on a module, and it responds to the full 32-Mbyte, slot-specific, M-bus I/O-space region. If TYPDUAL is asserted, the FBIC is one of two FBICs, and responds only to the upper or lower 16 Mbytes of the slot-specific, M-bus I/O-space region. If an FBIC is on a dual module, the FBIC always functions as a CVAX pin-bus grantee.

5.2.0.4.3. TYPAGNTE<0>--A/B Processor or CVAX Pin-Bus Grantee

When the TYPDUAL pin is asserted, the TYPAGNTE input pin indicates whether the FBIC is associated with the A or B processor of a dual-processor module. If TYPAGNTE is deasserted, the FBIC is the B processor and responds to the lower 16 Mbytes of the 32-Mbyte, slot-specific, M-bus I/O-space region. If TYPAGNTE is asserted, the FBIC is the A processor and responds to the upper 16 Mbytes of the 32-Mbyte, slot-specific, M-bus I/O-space region.

When the TYPDUAL pin is deasserted, the TYPAGNTE indicates whether the FBIC is the default CVAX pin-bus master. If TYPAGNTE is deasserted, the FBIC is the CVAX pin-bus grantor and monitors CDMR and drives CDMG. If TYPAGNTE is asserted, the FBIC is the CVAX pin-bus grantee and drives CDMR and monitors CDMG.

5.2.0.4.4. TYPRET<0>--CVAX Pin-Bus Retriable

The TYPRET input pin indicates whether the FBIC can retry the CVAX pin-bus master to complete a deadlocked M-bus transaction. If TYPRET is deasserted and the CVAX pin-bus has a pending M-bus transaction, the FBIC retries M-bus I/O-space and interrupt-acknowledge transactions that require the CVAX pin-bus. If TYPRET is asserted, and the CVAX pin-bus has a pending M-bus transaction, the FBIC retries the CVAX pin-bus when M-bus I/O-space and interrupt-acknowledge transactions require the CVAX pin-bus.

5.2.0.4.5. TYPSYNC<0>--CVAX Pin-Bus Synchronous

The TYPSYNC input pin indicates whether the CVAX pin-bus is synchronous or asynchronous. If TYPSYNC is deasserted, the CVAX pin-bus is asynchronous, and the FBIC performs a full handshake of the CDS and CRDY/CERR signals. If TYPSYNC is asserted, the CVAX pin-bus is synchronous and the FBIC initiates external cache tag-store probes before CAS is asserted; in addition, it does not monitor CDS with respect to CRDY/CERR assertion/deassertion.

5.2.0.4.6. DEVIRQ<3:0>--Device-Interrupt Requests

The DEVIRQ input pins are edge-sensitive equivalents of the IPDVINT<IPL17:IPL14> bits when the IPDVINT register is operating as a device-interrupt unit.

5.2.0.4.7. ROMOE<0>--External-ROM Output Enable

When ROMOE is asserted, the external diagnostic/self-test ROM(s) drive CDAL. The FBIC asserts ROMOE during ROM assembly.

5.2.0.4.8. ROMWID32<0>--External-ROM Width

If ROMWID32 is deasserted, the ROM is 16 bits wide and requires two word reads to assemble a longword. If ROMWID32 is asserted, the ROM is 32 bits wide and does not require assembly. The ROMWID32 input pin connects to power/ground through a 47-ohm series resistor, as appropriate.

5.2.0.4.9. ROMWADDR<0>--External-ROM Word Address

The ROMWADDR signal specifies the LSB of the word address for 16-bit ROM. It connects directly to the LSB address line of an external 16-bit ROM. For external 32-bit ROM, it is left unconnected.

5.2.0.4.10. MNFMOD--Manufacturing Mode

The MNFMOD input pins specify the value of the FBICSR<MFMD> bits. These pins require an external 1K-ohm pull-up resistor.

5.2.0.4.11. LEDS<0>--LEDs Value

The LEDS output pins drive the module LEDs through a 74LS244 buffer or equivalent. These LEDS pins continually reflect the value of the FBICSR<LEDS> bits.

5.2.0.4.12. TESTOUT<0>--Test Output

The TESTOUT pin is the end of the gate-array, I/O-pad, NAND tree used during functional and parametric testing of components.

5.2.0.4.13. TRISTATE<0>--Tristate All FBIC Pins

When the TRISTATE input pin is asserted, the FBIC tristates all of its output pins.

5.3. FBIC Transactions

This section describes the transactions involving the FBIC on both the CVAX pin-bus and the M-bus.

5.3.1. CVAX Pin-Bus Transactions

5.3.1.1. Read

Figure 5-19 shows the general format of synchronous CVAX pin-bus read transactions. The transaction shown is a memory-space octaword read. Memory-space longword/quadword/hexword reads, I/O-space reads, EPR reads, and interrupt acknowledges are of the same form but with the appropriate number of data-transfer cycles. The CVAX pin-bus slave may stall data transfers in cycle increments by not asserting CRDY. For an asynchronous CVAX pin-bus, additional dead cycles may occur before and after each longword transfer because of synchronizer delays in the CDS and CRDY paths.

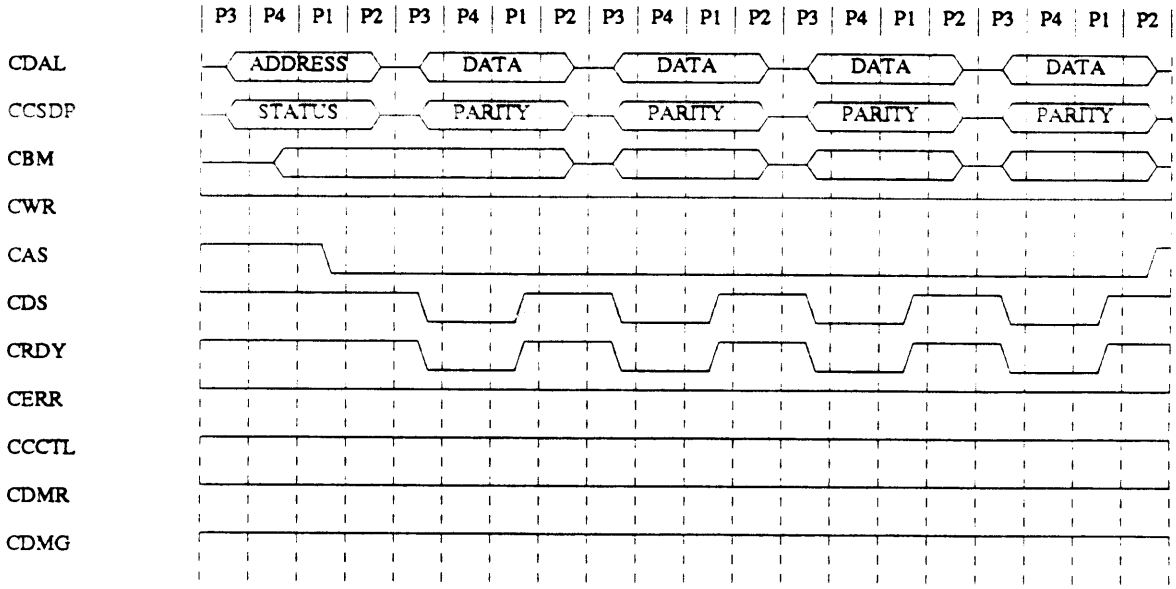


Figure 5-19: Read Transaction

5.3.1.2. Write

Figure 5-20 shows the general format of synchronous CVAX pin-bus write transactions. The transaction shown is a memory-space octaword write. Memory-space longword/quadword/hexword writes, I/O-space writes, and EPR writes are of the same form but with the appropriate number of data-transfer cycles. The CVAX pin-bus slave can stall data transfers in cycle increments by not asserting CRDY. For an asynchronous CVAX pin-bus, additional dead cycles can occur before and after each longword transfer because of synchronizer delays in the CDS and CRDY paths.

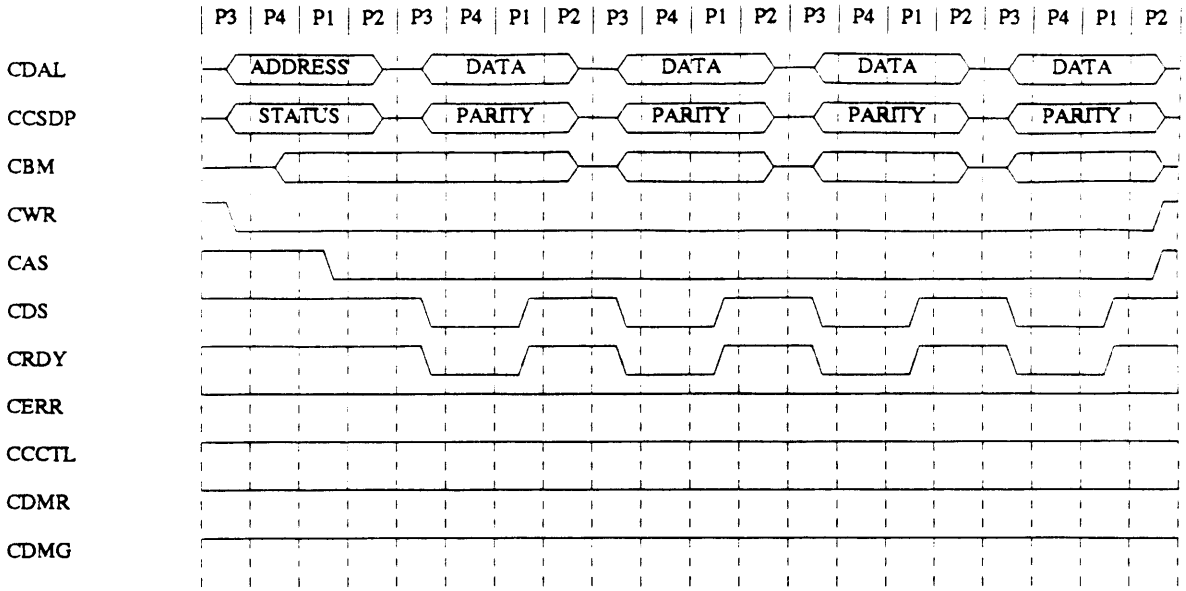


Figure 5-20: Write Transaction

5.3.1.3. External-Cache Miss

Figure 5-21 shows an external-cache read miss transaction and the start of the victim-read and octaword cache-invalidate transaction from the FBIC. An external-cache write miss transaction has the same form.

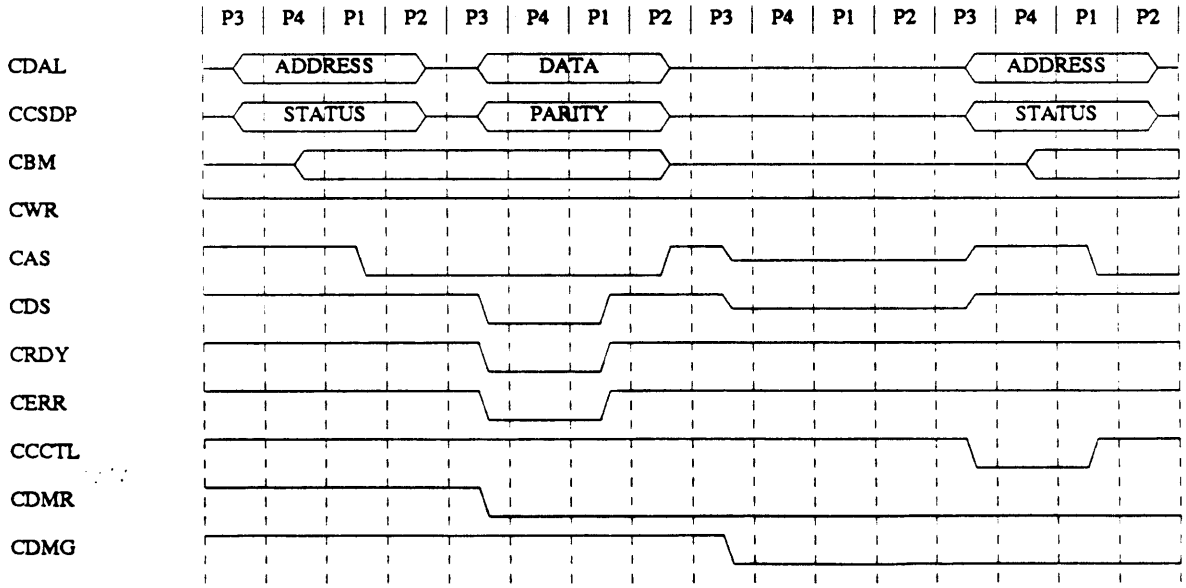


Figure 5-21: External-Cache Read Miss

5.3.1.4. External-Cache Victim-Read Transaction

Figure 5-22 shows an external-cache data-store transaction involving victim read and octaword cache invalidate. Whenever an external-cache miss occurs, the FBIC issues an external-cache victim-read transaction to remove the victim line from the external-cache data store and the CVAX internal cache.

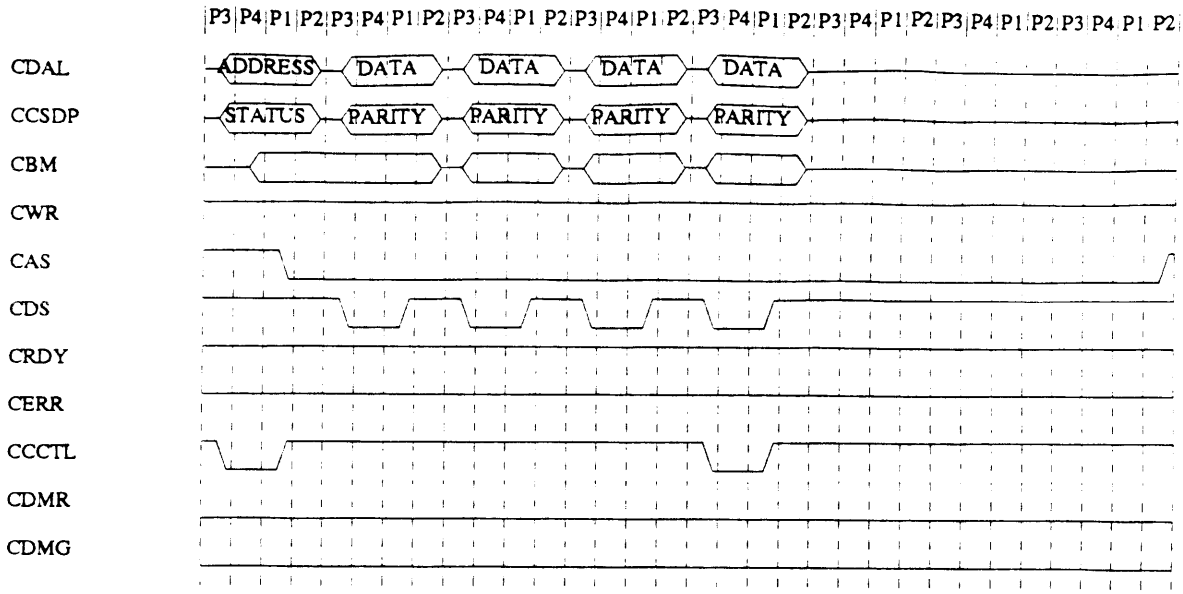


Figure 5-22: External-Cache Victim-Read Transaction

5.3.1.5. External-Cache Fill

Figure 5-23 shows a data-store fill transaction in the external cache. The FBIC issues such a transaction to load an external-cache line that missed.

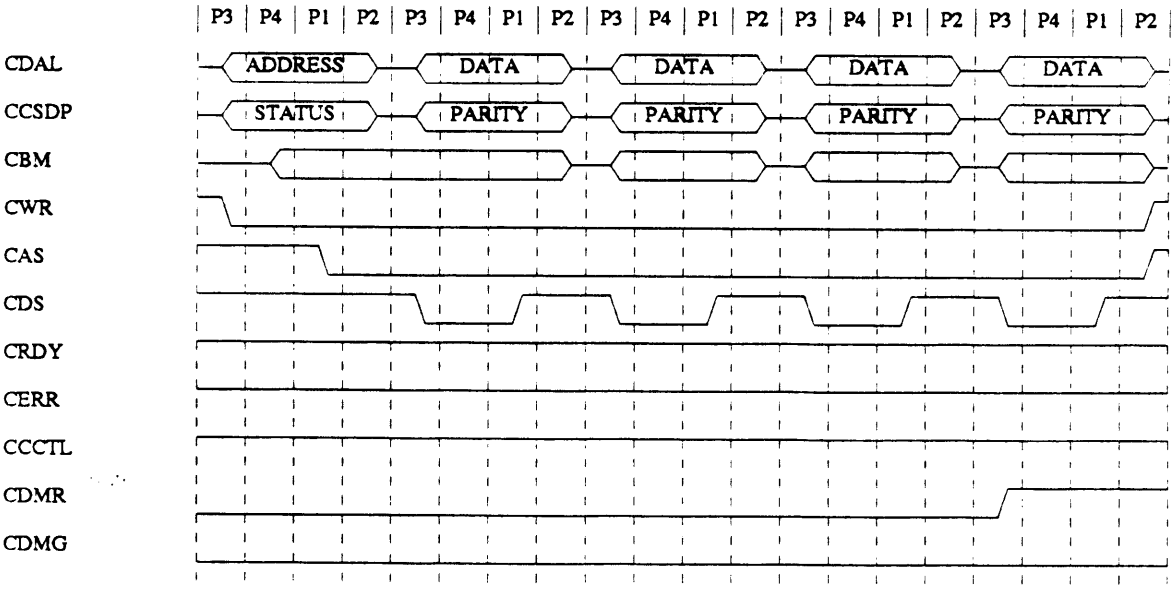


Figure 5-23: External-Cache Fill Transaction

5.3.1.6. External-Cache Shared-Read

Figure 5-24 shows an FBIC external-cache shared-read transaction that supplies read data to the M-bus.

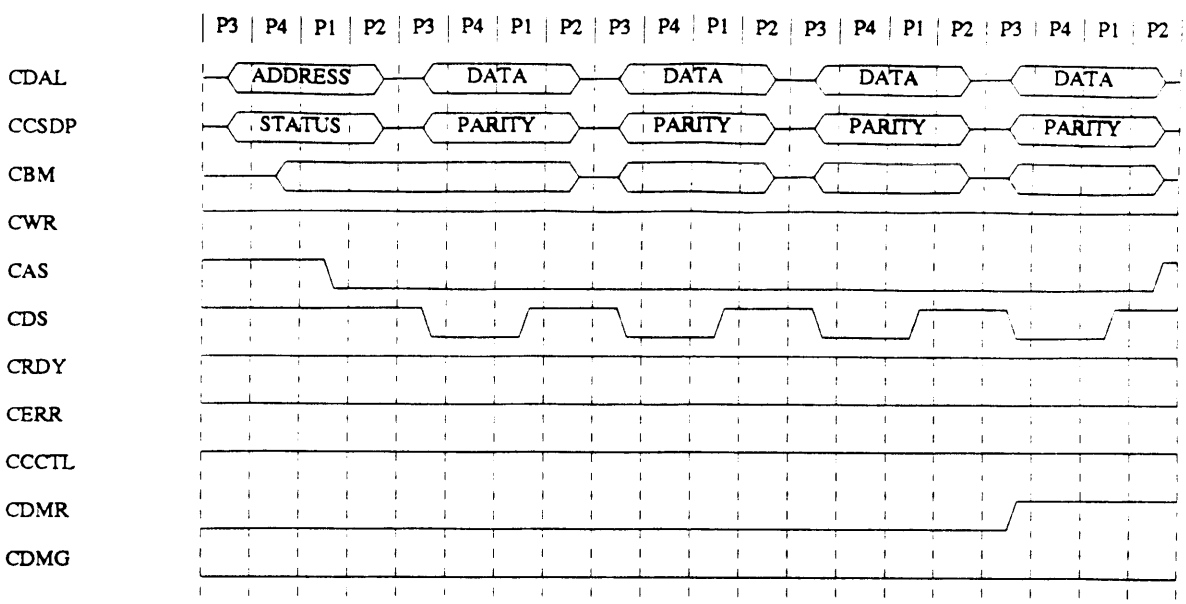


Figure 5-24: External-Cache Shared-Read

5.3.1.7. External-Cache Data-Write-Through Update

Figure 5-25 shows a transaction in the external-cache data store involving a write-through update and octaword cache invalidate transaction. Whenever it receives an M-bus write through, the FBIC issues such a transaction, which updates the external-cache data store and invalidates the CVAX internal cache.

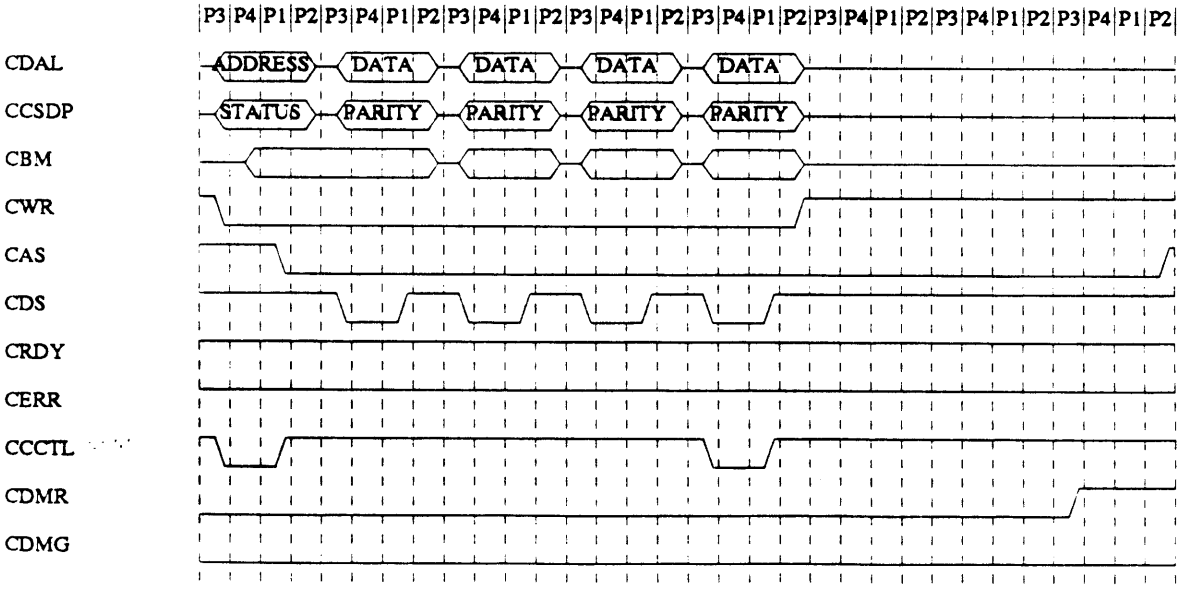


Figure 5-25: External-Cache Write-Through Update Transaction

5.3.1.8. 16-Bit-ROM Read Transaction

Figure 5-26 shows a ROM read for a module with 16-bit ROMs. The FBIC reads the ROM twice, assembles the two words into a longword, and supplies the longword to the CVAX pin-bus master. The figure does not accurately reflect the duration of the ROM read cycle because the ROM will drive data for two additional CVAX pin-bus cycles for each half of the longword and there is insufficient room on the page to correct the diagram.

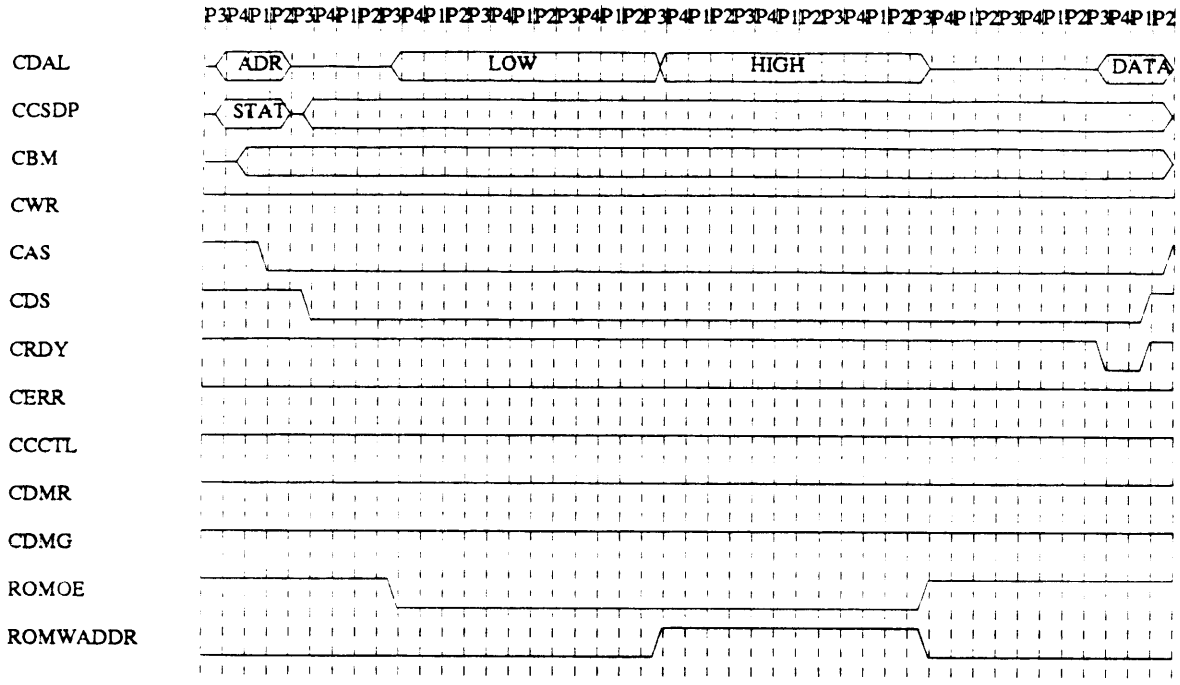


Figure 5-26: 16-Bit-ROM Read Transaction

5.3.1.9. 32-Bit-ROM Read Transaction

Figure 5-27 shows a ROM read for a module with 32-bit ROMs. The FBIC enables the ROM onto CDAL, latches it, disables the ROM, redrives the data, and asserts CRDY. Because of timing restrictions, the FBIC must redrive the data.

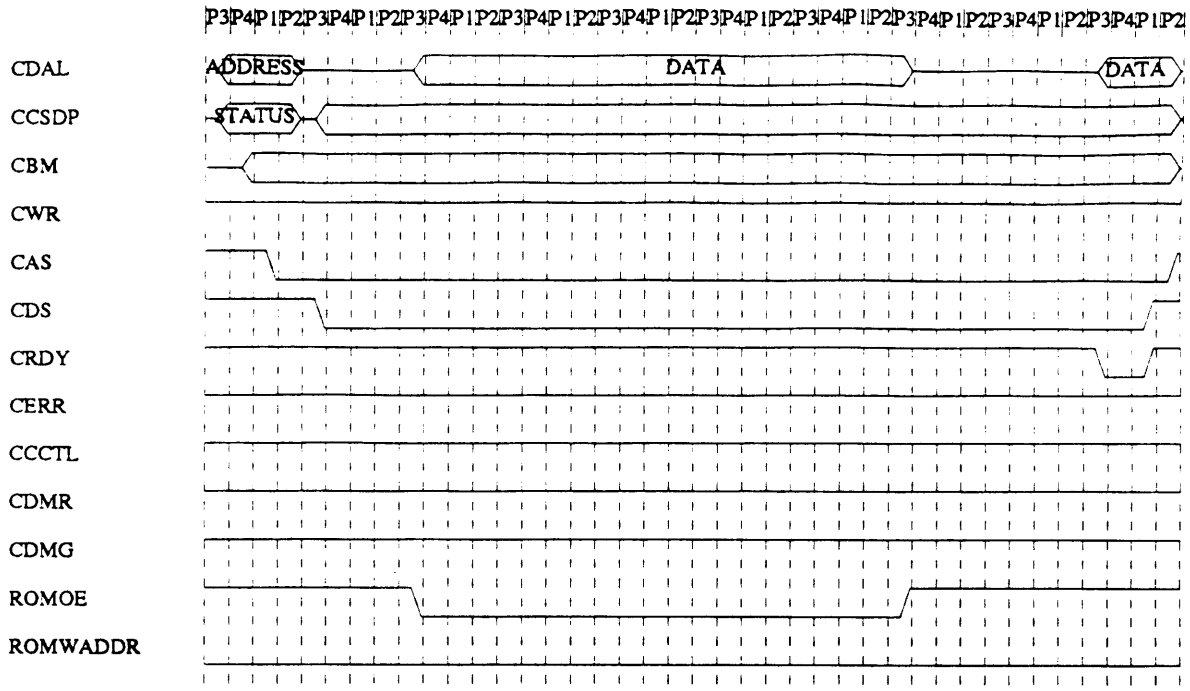


Figure 5-27: 32-Bit-ROM Read Transaction

5.3.2. M-bus Transactions

FBIC M-bus transactions conform to the *Firefox M-Bus Specification*. Please refer to that document for a detailed description of M-bus transactions.

5.4. FBIC Performance During System Use

The following tables contain examples of FBIC performance for external-cache hits, clean/dirty misses, write-throughs, shared reads, shared writes; global I/O space reads; and M-bus interrupt acknowledges. In all cases, the M-bus is otherwise idle and M-bus slaves respond in minimal time.

The data is from a DECSIM simulation run on 4 Dec 87 using the FBIC structural model. To guarantee fixed-length synchronizer delay and to approximately model the expected M-bus/CVAX pin-bus clock cycle relationship, the simulation was run with an M-bus to CVAX pin-bus clock cycle ratio of 3 to 2. This implies that a cycle in the M-bus column is 1.5 times longer than a cycle in the CVAX pin-bus column.

Under each transaction analysis, an M or a C in a column indicates that the cycle is counted in the overall transaction duration or penalty calculation.

Table 5-18 shows the external-cache hit performance.

Table 5-18: External-Cache Hit Performance

C	Cycles	M	Cycles
C	Address		
C	Data		
—		—	
2		0	

Table 5-19 shows the external-cache clean-miss penalty.

Table 5-19: External-Cache Clean-Miss Penalty

C	Cycles	M	Cycles
C	Address		
C	Data		
C	Retry/DMR Asrt		
	Grant Delay	M	Sync
	Grant	M	Request
	Pipe	M	P1 Arb
	Victim adr	M	P2 Miss adr
	Victim data 0	M	P3 Wait 0
	Victim data 1	M	P4 Wait 1
	Victim data 2	M	P5 Wait 2
	Victim data 3	M	P6 Wait 3
	Inval Delay	M	P7 Data 0
	Inval Delay	M	P8 Data 1
	Inval Delay	M	P9 Data 2
	Tag Write	M	P10 Data 3
C	Sync	Done	
C	Pipe		
C	Fill data 0		
C	Fill data 1		
C	Fill data 2		
C	Fill data 3/DMR DeAsrt		
C	Ungrant Delay		
C	Ungrant		
11		12	

Table 5-20 shows the external-cache dirty-miss penalty.

Table 5-20: External-Cache Dirty-Miss Penalty

C	Cycles	M	Cycles
C	Address		
C	Data		
C	Retry/DMR Asrt		
C	Grant Delay		
C	Grant		
C	Pipe		
C	Victim adr		
C	Victim data 0		
C	Victim data 1		
	Victim data 2	M	Sync
	Victim data 3	M	Request
	Inval Delay	M	P1 Arb
	Inval Delay	M	P2 Victim adr
	Inval Delay	M	P3 Data 0
	Wait	M	P5 Data 1
	....	M	P6 Data 2
	Wait	M	P7 Data 3
C	Sync		Done
C	Pipe		....
C	Pipe/Tag write		....
	Wait	M	Sync
	....	M	Request
	....	M	P1 Arb
	....	M	P2 Miss adr
	....	M	P3 Wait 0
	....	M	P4 Wait 1
	....	M	P5 Wait 2
	....	M	P6 Wait 3
	Sync	M	P7 Data 0
	Tag Write	M	P8 Data 1
	Wait	M	P9 Data 2
	Wait	M	P10 Data 3
C	Sync		Done
C	Pipe		
C	Fill data 0		
C	Fill data 1		
C	Fill data 2		
C	Fill data 3/DMR DeAsrt		
C	Ungrant Delay		
C	Ungrant		
20		20	

Table 5-21 shows the external-cache write-through penalty.

Table 5-21: External-Cache Write-Through Penalty

C	Cycles	M	Cycles
	Address		
	Data		
C	DMR Asrt		
	Rdy/Grant Delay	M	Sync
	Grant	M	Request
	Wait	M	P1 Arb
	....	M	P2 Write adr
	....	M	P3 Data 0
	....	M	P4 Data 1
	....	M	P5 Data 2
	Wait	M	P6 Data 3
C	Sync		Done
C	Pipe		
C	Tag Write		
C	DMR DeAsrt		
C	Ungrant Delay		
C	Ungrant		
7		8	

Table 5-22 below shows the cache shared read penalty. The M-bus penalty for read from cache versus read from memory is as follows:

17 C-cycles (Sync to Data 3 - request ==> data ready)
\+ 2 M-cycles (Sync to Done - synch done ==> fifo pipeline started)
– 4 M-cycles (Required 4 M-bus waits for probe)

17 C – 2 M

The CVAX pin-bus penalty for shared read from its cache during CVAX pin-bus read hit is as follows:

2 C-cycles (Tag Probe to Tag Write)
12 C-cycles (Sync to Data 2 - request ==> data ready)
\+ 6 M-cycles (Sync to Data 3 - synch done ==> empty fifo)
\+ 4 C-cycles (Sync to Ungrant - synch fifo emptied ==> bus idle)

18 C + 6 M

The CVAX pin-bus penalty for shared read from its cache during CVAX pin-bus write hit is as follows:

6 C-cycles (Tag Probe to Tag Write, wait to do write thru)
12 C-cycles (Sync to Data 2 - request ==> data ready)
\+ 6 M-cycles (Sync to Data 3 - synch done ==> empty fifo)
\+ 4 C-cycles (Sync to Ungrant - synch fifo emptied ==> bus idle)

22 C + 6 M

Table 5-22: External-Cache Shared-Read Penalty

C	Cycles	M	Cycles
			P1 Arb
			P2 Adr
C	Sync		P3 Wait 0
C	Tag Probe		....
C	Tag Write		P4 Wait 1
C	Wait		....
C	Wait		P5 Wait 2
C	Wait		P6 Wait 3/Shared
C	Pipe/Sync		P7 Data Wait
C	Pipe		P7 Data Wait
C	DMR Asrt		P7 Data Wait
C	Grant Delay		P7 Data Wait
C	Grant		P7 Data Wait
C	Pipe		P7 Data Wait
C	Pipe		P7 Data Wait
C	Pipe		P7 Data Wait
C	Adr		P7 Data Wait
C	Data 0		P7 Data Wait
C	Data 1		P7 Data Wait
C	Data 2		P7 Data Wait
	Data 3		M
	Wait	M	P7 Data Wait/Done
	....	M	P7 Data 0
	....	M	P8 Data 1
	....	M	P9 Data 2
	Wait	M	P10 Data 3
C	Sync		
C	DMR DeAsrt		
C	Ungrant Delay		
C	Ungrant		
22		6	

Table 5-23 shows the external-cache shared-write penalty.

The M-bus penalty for write-through to cache is as follows:

15 C-cycles (Pipe/Sync to Inval Delay 3 - synch request ==> inval done)
\+ 2 M-cycles (Sync to Done - synch done ==> delay pipeline started)
\+ 8 M-cycles (Required 8 Delay cycles to guarantee fair arbitration)
– 4 M-cycles (Required 4 data transfer cycles)

15 C + 6 M

The CVAX pin-bus penalty for shared write to its cache during CVAX pin-bus read hit is as follows:

2 C-cycles (Tag Probe and Tag Write)
18 C-cycles (Data Invalidate cycle)

20 C-cycles

The CVAX pin-bus penalty for shared write to its cache during CVAX pin-bus write hit is as follows:

6 C-cycles (Tag Probe, Tag Write, wait to do write thru)
18 C-cycles (Data Invalidate cycle)

24 C-cycles

Table 5-23: External-Cache Shared-Write Penalty

C	Cycles	M	Cycles
			P1 Arb
			P2 Adr
C	Sync		P3 Data 0
C	Tag Probe		....
C	Tag Write		P4 Data 1
C	Wait		....
C	Wait		P5 Data 2
C	Wait		P6 Data 3/Shared
C	Pipe/Sync		Busy
C	Pipe		Busy
C	DMR Asrt		Busy
C	Grant Delay		Busy
C	Grant		Busy
C	Pipe		Busy
C	Pipe		Busy
C	Pipe		Busy
C	Adr		Busy
C	Data 0		Busy
C	Data 1		Busy
C	Data 2		Busy
C	Data 3		Busy
C	Inval Delay		Busy
C	Inval Delay		Busy
C	Inval Delay		Busy
C	DMR DeAsrt	M	Busy/Sync
C	Ungrant Delay	M	Busy/Done
C	Ungrant	M	Busy/Delay
	Idle	M	Busy/Delay
	Idle	M	Busy/Delay
	Idle	M	Busy/Delay
	Idle	M	Busy/Delay
	Idle	M	Busy/Delay
	Idle	M	Busy/Delay
	Idle	M	Busy/Delay
	Idle	M	Busy/Delay
25		10	

Table 5-24 shows the global I/O space performance.

Table 5-24: I/O Transaction

C	Cycles	M	Cycles
C	Address		
C	Decode		
	Wait	M	Sync
	....	M	Request
	....	M	P1 Arb
	....	M	P2 Address
	....	M	P3 Mask
	Wait	M	P4 Status
	C	Sync	
	C	Pipe	
	C	RDY	
	--		--
	5		6

Table 5-25 shows the M-bus interrupt-acknowledge performance.

Table 5-25: Interrupt-Acknowledge Transaction

C	Cycles	M	Cycles
C	Address		
C	Decode		
	Wait	M	Sync
	....	M	Request
	....	M	P1 Arb
	....	M	P2 Address
	....	M	P3 Decode
	....	M	P4 Slave arb
	Wait	M	P5 Vector
C	Sync		Done
C	Pipe		
C	RDY		
—		—	
5		7	

5.5. Testability and System Diagnostic Support

The FBIC provides a variety of diagnostic functions for both chip-level testability and system self-test diagnostics, including the following:

- M-bus error detection and error logging
- M-bus module-type identification
- General-purpose scratch register support
- Thirteen programmable diagnostic functions
- Access to external and internal tags in I/O space
- External- and internal-cache miss detection
- M-bus and CVAX pin-bus timeout detection
- 6-bit status-indicator output
- 16/32-bit ROM control
- 2-bit manufacturing-mode input

TBD.

5.6. DC Characteristics

The following section lists the FBIC steady-state DC characteristics as published by LSI Logic for the L2000 ceramic chip package.

5.6.1. Absolute Maximum Ratings

Table 5-26 shows the absolute maximum ratings.

Table 5-26: Absolute Maximum Ratings

Parameter	Range
Storage temperature range	–40 to +125 C
Active temperature range	0 to +70 C
Supply voltage range	–0.3 to +7 V
Input or output voltage applied	–0.1 to +7.3 V

5.6.2. Electrical Characteristics

The DC characteristics of the FBIC appear in Table 5-27.

Table 5-27: DC Characteristics

Symbol	Parameter	Condition	Minimum	Maximum	Units
Vih	High-level input voltage (TTL)	-	2.0	-	V
Vih	High-level input voltage (CMOS)	-	3.5	-	V
Vil	Low-level input voltage (TTL)		-	-	0.8
Vil	Low-level input voltage (CMOS)		-	-	1.5
Voh	High-level output voltage	B1: Ioh = −1mA		2.4	-
		B2: Ioh = −2mA	2.4	-	V
		B4: Ioh = −4mA	2.4	-	V
		B8: Ioh = −8mA	2.4	-	V
		B12: Ioh = −12mA	2.4	-	V
Vol	Low-level output voltage	B1: Iol = 1mA	-	0.4	V
			B2: Iol = 2mA	-	0.4
		B4: Iol = 4mA		-	0.4
		B8: Iol = 8mA		-	0.4
		B12: Iol = 12mA		-	0.4
Ioz	Three-state output leakage current	Voh = Vss or Vdd	−10	10	uA
Cin	Input capacitance	Any input		2	
Cout	Output capacitance	Any output		4	
Specified temperature range					0 to +70
Specified supply voltage range			+4.75 to +5.25		V

5.7. AC Characteristics

5.7.1. M-bus AC Characteristics

TBD.

5.7.2. CVAX pin-bus AC Characteristics

Table 5-28 outlines the FBIC CVAX pin-bus AC specification. This specification is subject to change upon characterization of the actual device; moreover, the parameters appearing therein are biased to be conservative estimates of the actual parameters.

Table 5-28: CVAX Pin-Bus AC Specification

Symbol	Parameter	Minimum	Maximum	Load (pF)
TDALD	P3 rising to valid CDAL<31:00> (addr)	4	25	100
TDALD	P3 rising to valid CDAL<31:00> (data)	4	22	100
TDALH	P2 rising to invalid CDAL<31:00> (address/data hold time)	3	15	100
TDALHLZ	P2 rising to CDAL<31:00> tristate	4	25	100
TDS	Setup of CDAL<31:00> to P1 rising	5	-	-
TDH	Hold of CDAL<31:00> after P1 risen	10	-	-
TDPS	Setup of CCSDP<3:0> to P1 rising (data parity)	5	-	-
TDPH	Hold of CCSDP<3:0> (data parity) after P1 risen	10	-	-
TADDS	Setup of CDAL<31:00> (address) to CAS_L assertion	5	-	-
TADDH	Hold of CDAL<31:00> (address) after CAS_L assertion	10	-	-
TSD	P3 rising to valid CCSDP<3:0> (CS)	4	25	100
TASD	P1 rising to CAS_L assertion	4	20	100
TASID	P2 rising to CAS_L deassertion	4	23	100
TADRH	Hold of CDAL<31:00> (adr) after CAS_L assertion (provided by FBIC)	15 + (P1-P2) – TASD(max)	-	100
TADRS	Setup of CDAL<31:00> (adr) to CAS_L assertion (provided by FBIC)	15 + (P3-P1) – TDALD(max)	-	100
TDSD	P3 rising to CDS_L assertion	4	21	100
TDSID	P1 rising to CDS_L deasserion	4	24	100
TDATH	Hold of CDAL<31:00> (data) after CDS_L deassertion (provided by FBIC)	15 + (P1-P2) – TDSID(max)	-	100
TDATS	Set-up of CDAL<31:00> (data) to CDS_L deassertion (provided by FBIC)	+ (P3-P1) – TDALD(max)	15	-

(continued)

Symbol	Parameter	Minimum	Maximum	Load (pF)
TBMH	P3 rising to valid CBM<3:0>	4	25	100
TSD	P3 rising to CWR_L assertion	4	21	100
TSID	P2 rising to CWR_L deassertion	4	23	100
TSWS	Setup of CRDY_L or CERR_L asserted to P1 rising	10	-	-
TSWH	Hold of CRDY_L or CERR_L asserted after P1 risen	8	-	-
TDPD	P3 rising to valid CCSDP<3:0> (DP)	4	25	100
TDPEDF	CClockC falling to CDPE_L assertion (fast)	2	13	100
TDPEDS	P3 rising to CDPE_L assertion (Slow; CClockC input pin tied to VDD)	4	22	100
TRMOED	P3 rising to ROMOE_L assertion or deassertion	4	22	100
TRMWAD	P3 rising to ROMWADDR assertion or deassertion	4	22	100
TDMRD	P2 rising to CDMR_L assertion or deassertion	4	22	100
TDMGS	Setup of CDMG_L asserted to P3 rising	10	-	-
TDMGH	Hold of CDMG_L asserted to P3 risen	0	-	-
TTWED0	P3 rising to TAGWE_L assertion	4	18	50
TTWED1	P1 rising to TAGWE_L deassertion	4	26	50
TTCED0	P3 rising to TAGCE_L assertion	4	20	50
TTCED1	P1 rising to TAGCE_L deassertion	4	23	50
TTAGD	P4 rising to valid TCACHE<15:0>	4	20	30
TTAGHLZ	P2 rising to TCACHE<15:0> tristate	4	20	30
TTAGS	Setup of TCACHE<15:0> to TAGCE_L deassertion (provided by FBIC)	13	-	30
TTAGH	Hold of TCACHE<15:0> after TAGCE_L deassertion (provided by FBIC)	7	-	30

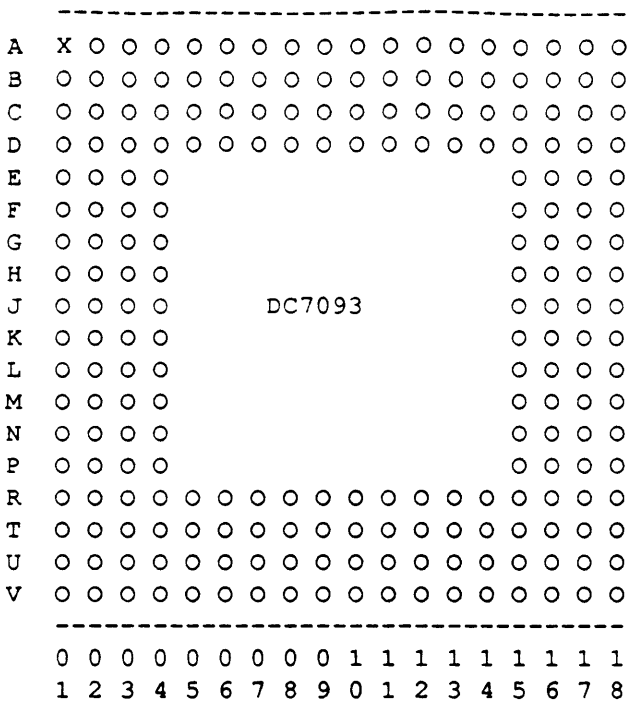
(continued)

Symbol	Parameter	Minimum	Maximum	Load (pF)
TDWED0	CAS_L assertion to DATWE_L assertion	3	14	50
TDWED1	CAS_L deassertion to DATWE_L deassertion	3	14	50
TDCED0	CWR_L deassertion, or CWR_L asserted and P3 rising to DATCE_L assertion	4	19	50
TDCED1	P1 rising, or CWR_L assertion to DATCE_L deassertion	4	21	50
TXOED0	CAS_L asserted and P2 rising to XOE_L assertion	4	20	50
TXOED1	P2 rising to XOE_L deassertion	4	22	50
TCTLDF0	CClock falling to CDPE_L assertion (fast)	2	12	50
TCTLDS0	P3 rising to CDPE_L assertion (Slow; CClockC input pin tied to VDD)	4	21	50
TRESETS	Setup of CRESET_L deasserted to CCLKA rising	10	-	-
TRESETH	Hold of CRESET_L asserted to CCLKA rising	2	-	-
TCNDXD0	P3 rising to CTINDXOE_L asserted	3	18	50
TMNDXD0	P3 rising to MTINDXOE_L asserted	3	17	50
TCNDXD1	P3 rising to CTINDXOE_L deasserted	3	19	50
TMNDXD1	P3 rising to MTINDXOE_L deasserted	3	18	50

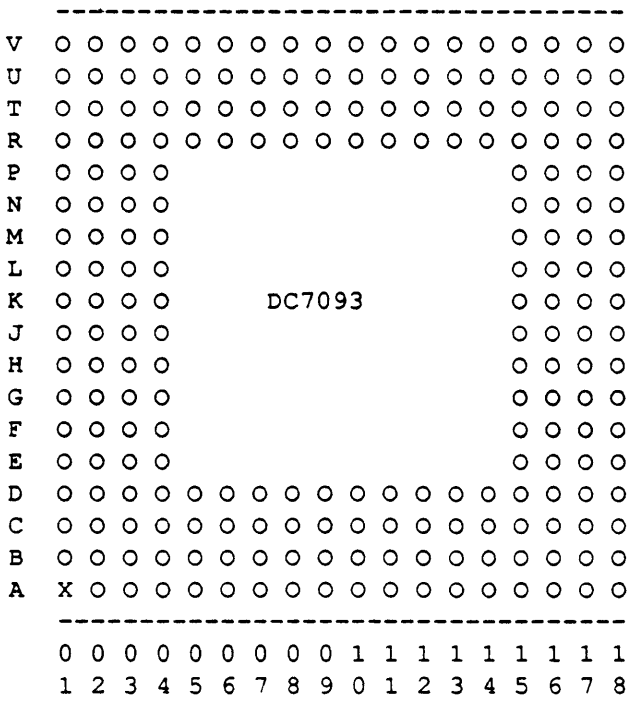
5.8. Package Diagram and Pin Assignment

The FBIC package pinout is pictured in Figure 5-28.

View from Top



View from Pin Side



Note: X is an index mark; there is no pin at this location.

Figure 5-28: Package Pinout

Table 5-29 shows the FBIC signal, power, and ground pin assignments.

Table 5-29: FBIC Signal, Power, and Ground Pin Assignments

CVAX Pin-Bus	M-Bus	Cache Control	Miscellaneous	Power/Ground
B11 - CCSDP<0>	P15 - MBRM<0>	E17 - TCACHE<0>	D13 - MODCL<0>	V02 - VDD
C10 - CCSDP<1>	R16 - MBRM<1>	G15 - TCACHE<1>	C14 - MODCL<1>	V09 - VDD
A11 - CCSDP<2>	T18 - MBRM<2>	E16 - TCACHE<2>	B14 - TYPDUAL	V18 - VDD
B10 - CCSDP<3>	T17 - MBRM<3>	F16 - TCACHE<3>	C13 - TYPAGNTE	U01 - VDD
G17 - CDPE	R15 - MBRM<4>	F15 - TCACHE<4>	A14 - TYPRET	U17 - VDD
D04 - CAS	T16 - MBRM<5>	D18 - TCACHE<5>	D12 - TYPSYNC	L18 - VDD
D03 - CDS	U16 - MBRM<6>	D17 - TCACHE<6>	R05 - DEVIRQ<0>	J01 - VDD
D11 - CBM<0>	R17 - MBRP	D16 - TCACHE<7>	V03 - DEVIRQ<1>	J18 - VDD
C11 - CBM<1>	R18 - MYMBRQ	E15 - TCACHE<8>	U03 - DEVIRQ<2>	B01 - VDD
A12 - CBM<2>	N18 - MBUSYI	C18 - TCACHE<9>	R04 - DEVIRQ<3>	B18 - VDD
D10 - CBM<3>	M15 - MBUSYO	C17 - TCACHE<10>	H18 - ROMOE	A02 - VDD
C01 - CWR	T10 - MCMD<0>	C16 - TCACHE<11>	B13 - ROMWID32	A09 - VDD
D02 - CRDY	V12 - MCMD<1>	D15 - TCACHE<12>	K16 - ROMWADDR	A17 - VDD
E04 - CERR	V11 - MCMD<2>	B16 - TAGSH	C12 - MNFMOD<0>	V01 - VSS
F18 - CCCTL	T09 - MCMD<3>	A16 - TAGDR	A13 - MNFMOD<1>	V08 - VSS
J16 - CDMR	V13 - MSTATUS<0>	C15 - TAGPAR	L15 - LEDS<0>	V10 - VSS
G18 - CDMG	R10 - MSTATUS<1>	F17 - TAGWE	L17 - LEDS<1>	V17 - VSS
E02 - CRESET	U12 - MCPAR	H16 - TAGCE	L16 - LEDS<2>	U02 - VSS
F03 - SYSRESET	T11 - MSPAR	C04 - DATCE<0>	K17 - LEDS<3>	U18 - VSS
U04 - CHALT	U10 - MDPAR	D05 - DATCE<1>	K15 - LEDS<4>	K01 - VSS
V16 - CIRQ<0>	U11 - MCDRV	A03 - DATCE<2>	J17 - LEDS<5>	K18 - VSS
U15 - CIRQ<1>	R11 - MSDRV	B03 - DATCE<3>	T02 - TESTOUT	H01 - VSS
T13 - CIRQ<2>	R03 - MDDRV	C03 - DATWE	R01 - TRISTATE	B02 - VSS
U14 - CIRQ<3>	N17 - MSHAREDI	C02 - XOE		B17 - VSS
T01 - CRD	N16 - MSHAREDO	H17 - ECL		A08 - VSS
P04 - MEMERR	M16 - MDATINVI	G16 - CTINDX_OE		A10 - VSS
E03 - CCLKA	M18 - MDATINVO	R02 - MTINDX_LE		A18 - VSS
F04 - CCLKB	A15 - MID<0>	H15 - MTINDX_OE		E18 - VSS
J15 - CCLKC	B15 - MID<1>			B12 - VSS (Unused)
B09 - CDAL<0>	D14 - MID<2>			C07 - VSS (Unused)
C09 - CDAL<1>	T03 - MRESET			M17 - VSS (Unused)
B08 - CDAL<2>	P18 - MABORTI			U13 - VSS (Unused)
D09 - CDAL<3>	N15 - MABORTO			T07 - VSS (Unused)
A07 - CDAL<4>	R14 - MIRQI<0>			N01 - VSS (Unused)
C08 - CDAL<5>	V15 - MIRQI<1>			H04 - VSS (Unused)
B07 - CDAL<6>	T14 - MIRQI<2>			
D08 - CDAL<7>	T12 - MIRQI<3>			
A06 - CDAL<8>	T15 - MIRQO<0>			
B06 - CDAL<9>	R13 - MIRQO<1>			
D07 - CDAL<10>	R12 - MIRQO<2>			
A05 - CDAL<11>	V14 - MIRQO<3>			
C06 - CDAL<12>	T04 - MHALT			
B05 - CDAL<13>	P17 - MCLKA			
D06 - CDAL<14>	P16 - MCLKB			
C05 - CDAL<15>	R09 - MDAL<0>			
A04 - CDAL<16>	U09 - MDAL<1>			
B04 - CDAL<17>	T08 - MDAL<2>			
D01 - CDAL<18>	U08 - MDAL<3>			
E01 - CDAL<19>	R08 - MDAL<4>			
G04 - CDAL<20>	V07 - MDAL<5>			

CVAX Pin-Bus	M-Bus	Cache Control	Miscellaneous	Power/Ground
F02 - CDAL<21>	U07 - MDAL<6>			
G03 - CDAL<22>	R07 - MDAL<7>			
F01 - CDAL<23>	V06 - MDAL<8>			
G02 - CDAL<24>	T06 - MDAL<9>			
H03 - CDAL<25>	U06 - MDAL<10>			
G01 - CDAL<26>	R06 - MDAL<11>			
J04 - CDAL<27>	V05 - MDAL<12>			
H02 - CDAL<28>	U05 - MDAL<13>			
J03 - CDAL<29>	T05 - MDAL<14>			
J02 - CDAL<30>	V04 - MDAL<15>			
K02 - CDAL<31>	P03 - MDAL<16>			
	N04 - MDAL<17>			
	P02 - MDAL<18>			
	N03 - MDAL<19>			
	P01 - MDAL<20>			
	M04 - MDAL<21>			
	N02 - MDAL<22>			
	M03 - MDAL<23>			
	L04 - MDAL<24>			
	M02 - MDAL<25>			
	L03 - MDAL<26>			
	M01 - MDAL<27>			
	K04 - MDAL<28>			
	L02 - MDAL<29>			
	K03 - MDAL<30>			
	L01 - MDAL<31>			