SECTION 4
FASRAM

This document describes the fast access standby random access memory (FASRAM), an intermodule bus 3 (IMB3) module, designed to be the memory of a system in the modular embedded controller family of modular microcontrollers from Motorola. This family includes a series of modules from which numerous microcontrollers (MCUs) are being assembled. Currently available modules include CPU, timer, serial communications, system integration, and memory modules.

This memory module, in its fast access mode, currently works only with the CPU32X; however, other CPU modules which can use this memory will become available.

4.1 FASRAM Description

The FASRAM connects to both the IMB3 and the CPU of the MCU. To the IMB3, the FASRAM appears as a normal RAM memory module with normal (non-bursting) IMB3 timing. To the CPU, the private fast access bus (FAB) appears as a memory with half the access time of a normal IMB3 bus cycle, while permitting the CPU to simultaneously access another IMB3 module (via the IMB3) for instructions. The FASRAM is capable of arbitrating between the IMB3 and the FAB. When simultaneous accesses are requested, the FAB will have access priority.

The normal use of the FASRAM will be as stack space or as frequently accessed variable storage. The processor will also be able to execute code from the FASRAM. However, the instruction fetches must take place over the IMB3 since the CPU32x will dedicate the FAB to data accesses. These instruction fetches will not be burstable.

The FASRAM includes two comparators to facilitate debugging software, and a visibility bus (VB) that reflects the activity on the FAB.

4.2 FASRAM Features

- One clock fast access RAM for the attached CPU
- Two-clock access to the IMB3
- Long word, word, and byte accesses supported
- Low power and low voltage (3 V) design
- Two Kbytes expandable to 32 Kbytes in 2-Kbyte increments
- Variable size standby static RAM (mask option)
- Dynamic dual access (IMB3 and FAB) with operand coherency
- Two address comparators (equality or range comparisons)
- External visibility bus (as a bond-out option)

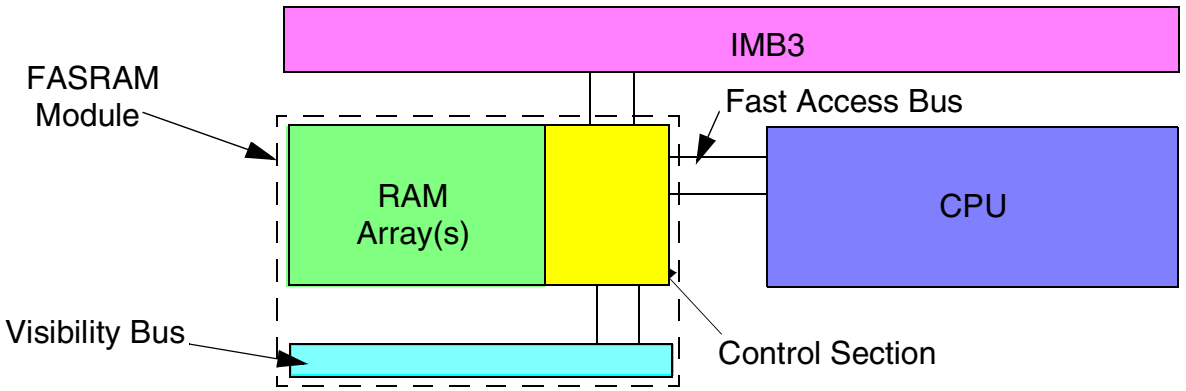

Figure 4-1 An MCU With a FASRAM Module

4.3 Operating Modes

The FASRAM module has five modes of operation: normal, standby, reset, test, and stop. After reset, the FASRAM is in stop mode.

4.3.1 Normal Operation

The normal operating mode for the FASRAM is to supply data to the IMB3 and the FAB whenever either bus requests an access. Priority is given to the FAB in case of simultaneous requests. The FAB access takes a minimum of one clock period, while the IMB3 access takes a minimum of two clock periods. Operating power is supplied by V_{DD} .

The memory can be read and written in bytes, words, or long words, by either bus. To guarantee coherency of operands, the FASRAM will refuse to grant access to one bus until the other completes its entire operand. Address acknowledge will be asserted to both sides upon a successful decode, however, the bus cycle to the “losing” side will have DTACK delayed until the “winning” side has finished. The maximum amount of data transferred for any single bus cycle is one word (16 bits). A byte or aligned word access takes only one bus cycle to complete. A long word or misaligned word access will take two bus cycles to complete.

The RAM array may be programmed to respond to either program space only or to both data and program space.The array may be programmed to respond to either supervisor space only or to both user and supervisor space. Each byte of the array appears at the same address and space for both an IMB3 access and a FAB access.

Operand coherency is provided by the IBLOCK signal and the FBLOCK signal. When either signal is asserted by a bus master during a RAM array access, the other bus master will not be granted access to the RAM array until the signal is negated. For the CPU32x, this signal is asserted only by the TAS instruction and during long word cycles. If a bus master does not use the IBLOCK signal or the FBLOCK signal, then the operand is not coherent; ISIZ and FSIZ do not affect coherency determination.

When the debugging comparators are in use and are forcing FAB accesses to be run on the IMB3, operand coherency can not be guaranteed.

For the initial configuration, the FASRAM will only perform data accesses across the FAB for the CPU32X. However, nothing in the design of this module shall preclude using the FAB for instruction accesses also. The FASRAM can still perform instruction accesses across the IMB3.

Access to the control registers is possible only over the IMB3. The FAB can only access the RAM array. The decoding for the module control register addresses will be local to this module.

An IMB3 access to the FASRAM will look exactly like an access to any other internal module. The IMB3 state machine in the FASRAM will assert the IMB3 handshaking signals with the required IMB3 timing.

The minimum V_{STBY} is 2.5 V and should not be greater than V_{DD} during normal operation. V_{STBY} should be equal to V_{DD} during the array current test.

4.3.2 Standby Operation

When the chip is to be powered down, the contents of the RAM array are maintained by the standby power supply, V_{STBY}. If the standby voltage falls below the minimum required voltage, then the RAM contents may be corrupted. The FASRAM will automatically switch to standby operation when the voltage on V_{DD} is below the voltage on V_{STBY}, with no loss of data. In this mode, the FASRAM does not respond to any bus cycles, and will set the stop bit in the FMCR. The system may experience unexpected operation when the CPU requests data from the FASRAM, since it will not be responding. If there is a bus cycle in progress during the time the power supply is switching, reads may be inaccurate and writes may corrupt the RAM contents. If standby operation is not desired, then the V_{STBY} pin should be connected to V_{DD}. The amount of RAM preserved in standby operation is determined by a mask option.

Differing implementations will have differing minimum amounts (and granularity) of standby RAM. The amount of current used by the V_{STBY} pin is proportional to the amount of RAM placed in standby mode.

Portions of the RAM array not selected by the mask option will not be preserved if V_{DD} falls below V_{STBY}. The current on V_{STBY} may exceed its specified maximum value at some time during the transition time that V_{DD} is at or below the voltage switch threshold to a threshold above V_{SS} . If the standby power supply cannot provide enough current to maintain V_{STBY} above the required minimum value, then a capacitor must be provided from V_{STBY} to V_{SS} The value of the capacitor may be calculated as C = I *t /V, where ‘I’ is the difference between the transition current requirement (approximately 1 mA per standby switch) and the power supply’s maximum current, ‘t’ is the duration of the V_{DD} transition near the voltage switch threshold. (The typical switching voltage is in the range $V_{DD} = V_{STBY} \pm V_T$).

4.3.3 Reset Operation

When a reset occurs, the FASRAM completes its current bus cycle before resetting. If a byte or word sized access was in progress, it will be completed. If a long word access was in progress, it is possible that only half (one word) of the operation will be completed. Reset will not directly affect the RAM array contents. (If reset is caused by power-on reset, then the parts of the RAM array not using V_{STBY} will of course be corrupted.)

System reset only affects the bus state machine and does not cause any bits to be reset in the control registers. Master reset will cause the control register bits to go to their reset state (as described in **4.7 Programmer's Model**). Most importantly, the FASRAM is forced into stop mode by a master reset.

4.3.4 Test Operation

Test mode is entered by asserting ITSTMODB from the test module. Many different tests may be run on the FASRAM by writing the appropriate bit(s) in the FTEST register while ITSTMOD is asserted. Test mode provides a way for the user to test the RAM, even after the MCU has been installed in a system. While in test mode (ITSTMODB asserted), the FASRAM may be used normally by a system, unless tests are actually in progress; however, running a test may corrupt the data in the RAM array.

When not in test mode, writes to the FTEST register will have no effect, and reads will always return zeros. All FASRAM tests must be conducted at a 3.3-V supply voltage and at room temperature. When performing the array current tests, operating power to the bit cells of the RAM array is provided by the V_{STBY} pin, while the V_{DD} pin powers the control circuits. It is required that V_{STBY} is present when performing the array current tests. This facilitates measuring the array current.

4.3.4.1 Open Circuit Tests

[Description of test modes and operation of same.]

4.3.4.2 Array Current Tests

[Description of test modes and operation of same.]

4.3.4.3 Scan Tests

[Description of test modes and operation of same.]

4.3.5 Stop Operation

Setting the STOP bit of the FMCR causes the FASRAM to enter stop mode. In this mode, the RAM array cannot be accessed by either bus. All data in the RAM array and in the control registers is retained, and the module can still transition to standby mode. The control registers may still be accessed by the IMB3. A bus cycle (from either the IMB3 or the FAB) that would have gone to the RAM array is ignored, although FAB accesses continue to be shown on the VB (provided it is enabled). The debug comparators will not operate. External logic can then decode the RAM array access space.

Stop mode differs from standby mode in that stop mode is entered under software control (or forced by reset) and is independent of the power source. Stop mode is exited by clearing the STOP bit in the FMCR.

4.3.6 FREEZE Operation

When the CPU entered in the background mode, the CPU asserts IFREEZEB and the FASRAM completes its current cycle. In this mode, the FASRAM acts as same way as it is in test mode. When the background mode is entered, the RLCK bit of the FMCR may be cleared. The FTEST becomes writable, just like the FASRAM is in the test mode.

4.4 Visibility Bus

Because the IMB3 may be in use for instruction fetches, the FAB on the FASRAM module will not be visible to the development system (via a show cycle on the IMB3).

An extra visibility bus is available as a bond-out option. This visibility bus is unidirectional, and shows the address and data for the previous bus cycle on the FAB. The visibility bus will continue to operate even in stop mode, provided that it is enabled by setting the VEN bit in the FMCR. The frequency of access may be as great as once per clock cycle. The bus cycle on the FAB will be shown on the VB as soon as it completes. The VB data strobe for the bus cycle will be delayed from the completion of the FAB cycle by one-half or one clock.

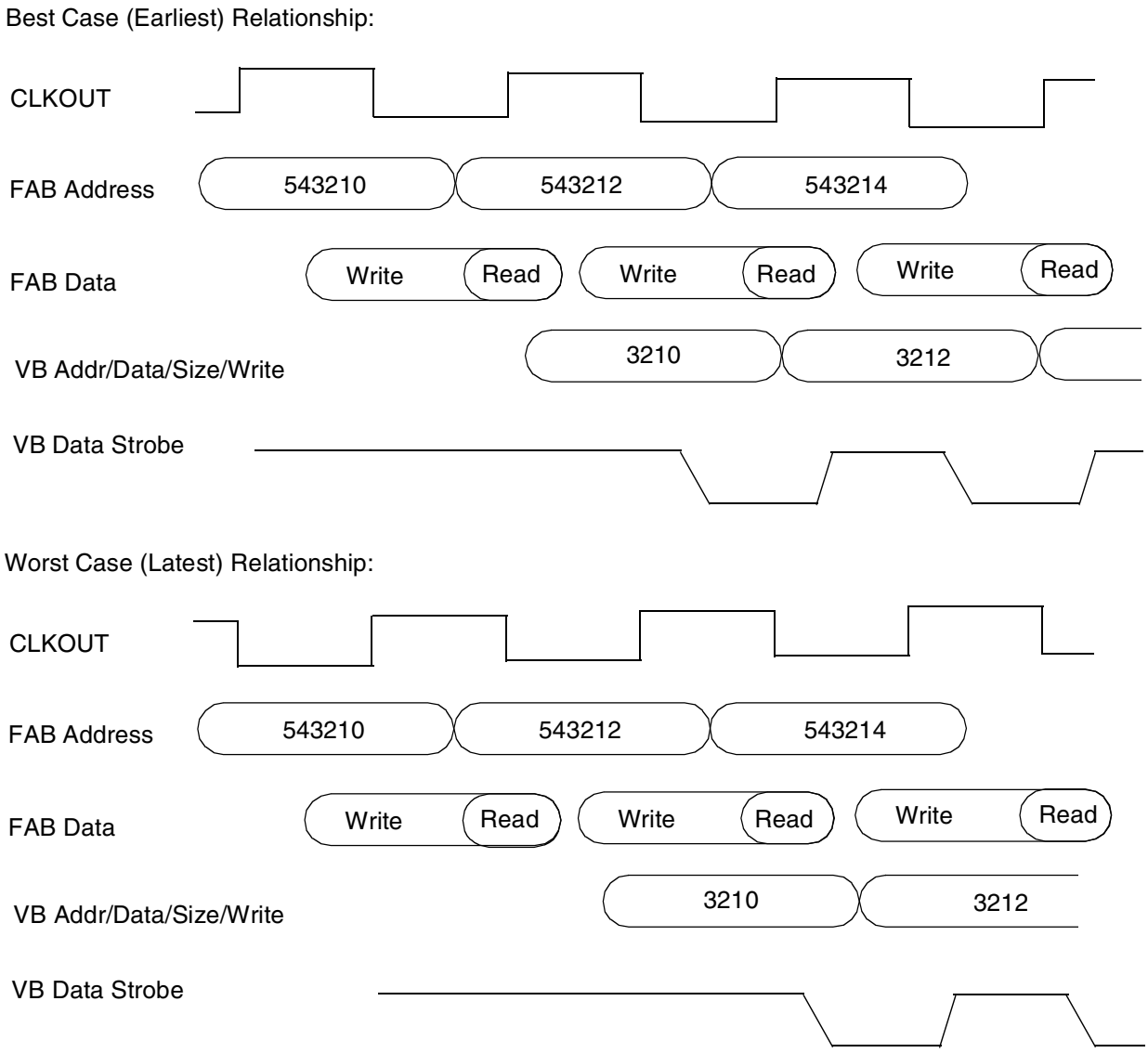

Figure 4-2 Timing Relationships Between the FAB and VB

The visibility bus will be driven to a new state shortly after the rising edge of CLKOUT, and the data strobe will be asserted shortly after the subsequent falling edge of CLK-OUT. If reset occurs before the VB has a chance to drive its pins, it is possible that the final FAB cycle will not be seen over the VB.

The visibility bus consists of up to 15 bits of the address decoded by the RAM array, and the data read or written to the RAM array. There is also a size pin, a read/write pin, and a data strobe. A bus cycle to any slave module on the FAB will appear on the visibility bus.

The visibility bus may be disabled by clearing the VEN bit in the FMCR. When the visibility bus is disabled (perhaps to reduce RFI), its address bus, data bus, and size pin are driven to their inactive state. The write pin and the data strobe will become comparator indicators. If a comparator is active, and it recognizes an in-range address, then the appropriate strobe will be asserted. The timing of the assertion of the comparator strobe will be the same as it would have been for a VB data strobe for that FAB cycle. To completely eliminate all activity on the visibility bus, you must disable the visibility bus and disable both comparators.

The write (VWRITE) pin and the data strobe (VDS) pins are a separate bond-out option. The rest of the visibility bus may be removed, while keeping these two pins.

4.4.1 Visibility Bus Address Pins (VADDR)

The portion of the address decoded by the RAM array is shown on the VADDR address bus. For example, if the array is six Kbytes in size, 13 address lines will be driven. If more than the necessary address lines are bonded out, the unused upper address lines should be ignored by the development system. Although the upper bits of the address are not shown, they are still used to match the base address of the RAM array as well as matching the comparator addresses. When the visibility bus is disabled, these pins are driven to a low state.

4.4.2 Visibility Bus Data Pins (VDATA)

The data for the RAM array access is driven on the VDATA data bus. A word sized access will drive all 16 bits of VDATA. When the visibility bus is disabled, these pins are driven to a low state.

If the access is byte sized, address bit 0 indicates which byte of the data bus contains valid data. If address bit 0 is one, then only data bits 7-0 are valid, and if address bit 0 is zero, then only data bits 15-8 are valid. The other byte should be ignored.

4.4.3 Visibility Bus Size Pin (VSIZE)

This VSIZE pin indicates the size of the RAM access. Word accesses will have a size of zero, and bytes will have a size of one. If the cycle is for the first word of a long word, a size indication of word is given. When the visibility bus is disabled, this pin is driven to a low state.

4.4.4 Visibility Bus Write Pin (VWRITE)

The direction of the RAM access is indicated on the VWRITE pin. If this pin is high, then the cycle was a read access. If this pin is low, then the cycle was a write access.

If the visibility bus is disabled, VWRITE becomes the comparator zero strobe. When comparator zero matches its address, this strobe will be asserted. Compares will be indicated for the FAB accesses when an action is taken (breakpoint is asserted over the FAB, or the FAB address acknowledge is inhibited). There is no indication of the address, data, size, or direction of the bus cycle.

4.4.5 Visibility Address Comparators

If the comparators are both set to range compare mode, then only VWRITE will be asserted on a compare; VDS will be asserted if the access is not in the range. For more details on range compares, see **4.5.3 Range Compare Mode**.

When the visibility bus and the comparators are all disabled, this pin is driven to a high state.

4.4.6 Visibility Bus Data Strobe (VDS)

When the address and data buses of the visibility bus have been driven to the value of the FAB address and data, then this strobe will be asserted. The development system will latch the contents of the visibility bus at the negative edge of this signal.

If the visibility bus is disabled, VDS becomes the comparator one strobe. When comparator one matches its address, this strobe will be asserted. Compares will be indicated for FAB accesses when an action is taken (breakpoint is asserted over the FAB, or the FAB address acknowledge is inhibited). There is no indication of the address, data, size, or direction of the bus cycle.

If the comparators are both set to range compare mode, then only VWRITE will be asserted on a compare; VDS will be asserted if the access is not in the range, provided that the address does reside in the FASRAM. For more details on range compares, see **4.5.3 Range Compare Mode**.

When the visibility bus and the comparators are all disabled, this pin is driven to a high state.

4.5 Address Comparators

Two address comparators are provided for development system support. These registers are capable of detecting a particular address during an access from the FAB.

The address may be optionally qualified by read or write, or user or supervisor mode. The comparators operate only during a RAM array access, never during a control register access. The comparators will not operate when the FASRAM is in stop mode. Bits 15 and 0 of the address are not used in any comparison. Both comparators may match on the same access. A comparator will not match an address if the address does not reside in the FASRAM.

When the comparators recognize that they should respond to an access, one of two actions may be taken. The CPU can be told that the address is not in the RAM array even if it actually is. Therefore, the CPU will be forced to run an IMB3 cycle to get the data. This IMB3 access may be seen externally if show cycles are enabled. This IMB3 access will be decoded by the FASRAM, and it will be treated just like any other cycle.

A forced IMB3 access will not cause the assertion of any MATCH bit in the FSTATUS register. Operand coherency cannot be guaranteed if the comparators force the bus cycle to the IMB3.

The other action that may be taken is to assert a breakpoint signal to the master. When the master runs an IMB3 breakpoint acknowledge cycle and a comparator has caused a breakpoint, then the FASRAM will respond to the breakpoint acknowledge cycle with a bus error signal. The breakpoint handler in the master should read the MATCH bits for each comparator in the FSTATUS register to determine which comparator caused the breakpoint, and then clear the asserted MATCH bit(s).

The comparators are enabled by setting at least one of the space bits (user or supervisor) and at least one of the read and write bits.

Each comparator has two basic modes of comparing addresses, equality and range. Equality mode matches an address if the access is within the word pointed to by the comparator address register. For range mode, comparator zero detects an address greater than or equal to the compare address, and comparator one detects an address less than or equal to the compare address. A comparator set to range mode matches an address only if the access is within the RAM array, never outside the RAM array.

The FASRAM will not make any response to IFREEZE if the CPU enters background mode in response to the breakpoint. The FASRAM will continue to operate normally.

The background mode program should negate the MATCH bits in the FSTATUS register to avoid confusing other software. If the visibility bus is disabled, then the VWRITE or VDS pins will be asserted for one-half clock period if their respective comparators match the current RAM array access address and function codes, in addition to performing the action specified in their control registers.

4.5.1 Equality Compare Modes

In equality compare mode, both comparators operate independently. The address of the bus cycle must exactly match the address in the compare register. Function codes and read or write are used to qualify the comparison. Bits 15 and 0 of the address are not used in the comparison.

If the comparator matches, then the selected action will be taken. If the action was to assert a breakpoint then the respective MATCH status bit will be set. The MATCH bit will not be set if the action was to force an IMB3 access.

If a comparator matches, and the visibility bus is disabled, then the appropriate strobe on the visibility bus will be asserted for one-half clock period. Comparator zero asserts VWRITE and comparator one asserts VDS.

4.5.2 Mixed Compare Modes

It is possible for one comparator to be in range compare mode, and the other to be in exact compare mode. In this mode, both comparators operate independently. The comparator configured for range mode will detect an access anywhere between the comparator address and the end of the RAM array. The direction of the range depends on which comparator is set to range mode. Comparator zero matches addresses equal to or greater than the comparator register, and comparator one matches addresses

less than or equal to the comparator register. Bits 15 and 0 of the address are not used in the comparison.

If the comparator matches, then the selected action will be taken. If the action was to assert a breakpoint then the respective MATCH status bit will be set. The MATCH bit will not be set if the action was to force an IMB3 access.

If a comparator matches, and the visibility bus is disabled, then the appropriate strobe on the visibility bus will be asserted for one-half clock period. Comparator zero asserts VWRITE and comparator one asserts VDS.

Setting a comparator to match on a range of addresses which are used for stack space is unwise. The CPU might be forced to take multiple breakpoint exceptions when only one was intended.

4.5.3 Range Compare Mode

In full range compare mode, when both comparators are in range mode, the comparator addresses are combined to form a range. The lower bound of the address range is in comparator zero, and the upper bound is in comparator one. Function codes and read or write are used to qualify the comparison. Bits 15 and 0 of the address are not used in the comparison. Since the individual comparators have separate qualification bits, it is strongly urged that both comparators be set to match on the same type of bus cycle, read or write and user or supervisor, or else unintended operation may occur.

When any access is made anywhere in the range, such that the access address is greater than or equal to the address for comparator zero and less than or equal to the address for comparator one, then the action specified by the comparator zero control bit will be performed. If the action was to assert a breakpoint then the MATCH status bit for comparator zero will be set. Only comparator zeros MATCH bit will be set on a range compare. The MATCH bit will not be set if the action was to force an IMB3 access. If the comparator range matches, and the visibility bus is disabled, then the VWRITE strobe on the visibility bus will be asserted for one-half clock period. VDS will be asserted on a non-compare. Therefore, when the visibility bus is disabled, any access to the FASRAM will result in the assertion of one of the two strobes.

If the comparator addresses are reversed, such that the lower bound is in comparator one and the upper bound is in comparator zero, then the comparators will never match an address. VDS will be asserted if the access resulted in only one comparator matching.

Setting the comparators to match on a range of addresses that are used for stack space is unwise. The CPU might be forced to take multiple breakpoint exceptions when only one was intended.

4.6 Usage

The following are a few examples of the procedures for operating the FASRAM.

4.6.1 Initialization

After a master reset, the FASRAM is in the stop mode. The control registers FMCR, FTEST, FCCR0, FCCR1, and FSTATUS have been forced to the reset values as specified in **4.8.1 FASRAM Module Configuration, Control, and Status Registers**. Control registers FBAR, comparator values zero and one, FMATCH, and some bits of FSTATUS are in a random state, even if standby power has been on. (Standby power supports only the RAM array.) In order to activate the FASRAM, do the following:

1. Write the FBAR registers to the value used for the base address of the RAM array.
2. Write the RLCK bit in the FMCR to a one. No subsequent instruction will be able to change the FBAR registers. This may be done at the same time as step three.
3. Write the STOP bit in the FMCR to a zero. This enables the FASRAM.

There are several other 'housekeeping' things that may be done now. The array space bits, the delay bit, and enable the visibility bus may be configured.

4.6.2 Comparator Setup

The comparators are off after a master reset. In order to turn them on for range compares, do the following:

1. Set the visibility bus to be on or off. If the VB is off, then when the comparators are enabled, the VDS and VWRITE pins become comparator strobes.
2. Write the range's low address to comparator zero, and the high address to comparator one.
3. Set both comparators to range mode. This may be done at the same time as step five.
4. Set comparator zero to perform the desired action, either a breakpoint or a forced IMB3 cycle. This may be done at the same time as step five.
5. Set both comparators to activate on the same type of cycle. The read, write, user, and supervisor bits should be the same for both comparators. This action actually enables the comparators.

Now the comparators are set to perform range compares. It is also possible to do exact compares by clearing the range bits. In this case, the bits indicating the cycle type do not have to be the same for each comparator.

4.6.3 Servicing the Comparators

If the comparators have asserted a breakpoint, then the CPU may or may not run a breakpoint acknowledge cycle. If it does, and the FASRAM has actually asserted a breakpoint, then the FASRAM will terminate the breakpoint acknowledge cycle with a bus error. This should cause the CPU to perform its breakpoint handler routine. The breakpoint handler should do the following:

1. Read the MATCH bit in the FSTATUS register. If set, then this is probably the comparator that caused the breakpoint. If both MATCH bits are set, you must use the WHICH bit or the saved address to tell which comparator actually

caused the most recent breakpoint.

2. Write a zero to the MATCH bit. This clears the indication of a breakpoint. It is permissible to write a zero even if it is already cleared. Steps one and two may be done with a BCLR instruction on the CPU32x.
3. If the only interest is in this breakpoint address for a single access, then turn off the comparator by writing zeros to the comparator control register.
4. If needed, read the saved breakpoint address and the associated information.
5. Check all other sources of breakpoints, including the other comparator. It is possible for multiple breakpoint sources to result in a single breakpoint exception. Software must decide what to do in that situation.
6. If the CPU runs a breakpoint acknowledge cycle, but neither MATCH bit is set, then the FASRAM will not respond to the cycle.

4.6.4 Testing the FASRAM

To test the FASRAM:

1. Enable test mode for all modules. Consult the integration module specification for guidelines.
2. Write to the test block of the integration module, and enable test mode. The RLCK bit can now be cleared.
3. Write appropriate commands to the FTEST register in the FASRAM.

Be certain that no attempt is made to access the FASRAM in the middle of a test. The FASRAM is not defined to operate correctly while a test is in progress.

4.7 Programmer's Model

The FASRAM module consists of two separately addressable sections, the RAM array itself and the control and status registers. The address mapping of the RAM array is governed by the control registers. Once the RLCK bit is set, the base address of the array is locked and cannot be re-mapped, unless the RLCK bit is cleared either in test mode or background mode. The control registers are mapped to a space within the 32-Kbyte module control block starting at 0x007FFB00 or 0xFFFFFB00 depending on the modmap signal on the IMB3, within supervisor data space. The mapping within the module control block is mask programmable.

All of the programmer's control and status registers are contained in the control section and are accessible via the IMB3 only. This includes any debugging or test features.

The FASRAM memory map occupies 32 bytes. **Figure 4-3** shows the decoding of the address lines for the FASRAM internal registers. A[6:0] must match the individual register decodes as detailed in **Table 4-1**.

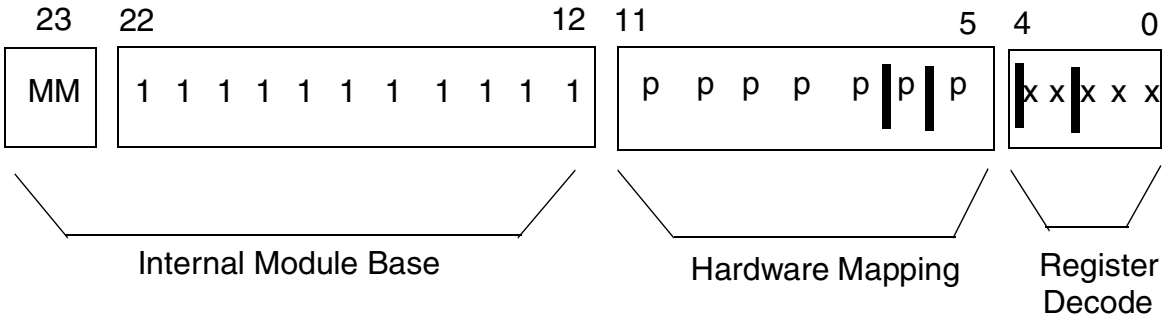

Figure 4-3 FASRAM Module Address Decoding

Table 4-1 shows all of the control registers as well as the RAM array. The registers are logically grouped as those registers needed to affect:

1. Array address mapping
2. Testing features
3. Debugging features

4.8 FASRAM RAM Array Addressing / Mapping

The RAM array can be placed anywhere in the address map of the MCU by means of the FASRAM base address register (FBAR) provided the following conditions hold:

1. The base address of the RAM array must be an even multiple of the RAM array size rounded up to the next power of two. Therefore, while the FASRAM may be im-plemented with RAM array sizes from two to 32 Kbytes in increments of two Kbytes, the lowest address within the array, the base address, must be on a 2-, 4-, 8-, 16-, or 32-Kbyte boundary. Only those address lines which define placement of the block will be compared (e.g., for an 8-Kbyte FASRAM only FBAR bits A[23:13] will be compared). This is mask programmable and based on the size of the implemented RAM array.
2. The RAM array also must not overlap any part of the 4-Kbyte internal peripheral register area, within user or supervisor data space. In the event that this occurs, the RAM array will not be accessible, allowing the control registers to be accessible so that they can be modified.
3. The FBAR registers may be written multiple times after reset, but will be locked once the RLCK bit in the FMCR is set. The FASRAM must be in stop mode and RLCK cleared for the FBAR registers to be writable. Once RLCK is set, the FBAR registers may not be altered and RLCK may not be cleared until master reset is asserted. This prevents accidental re-mapping of the array. An exception to this functionality is that when either test mode or background mode is entered, RCLK may be cleared. For more information on this operation, see **4.3.4 Test Operation**.

Table 4-1 FASRAM Module Programmers Model

Offset Address from the Control Register Mapping Address	FASRAM Module Control Registers	
	15 8	7 0
0xYF F6C0	Module configuration register (FMCR)	
0xYF F6C2	Test register (FTEST)	
0xYF F6C4	Base address (FBAR-H)	
0xYF F6C6	(FBAR-L)	
0xYF F6C8	Comparator 0 (≥) value FCMP0)0	
0xYF F6CA	Comparator 0 (≤) value FCMP1)0	
0xYF F6CC	Comparator control 0 (FCCR0)	Comparator control 1 (FCCR1)
0xYF F6CE	Most recent match address (FMATCH)	
0xYF F610	Not Used[15:8] (Reserved)	FSTATUS[7:0]
0xYF F612	Not used (Reserved)	
—	—	
0xYF F61E	Not used (Reserved)	
FASRAM Module RAM Array		
ADDRESS	15 8	7 0
*___*0000	—	
*___*0002	—	
—	—	
*___*FFFE	—	

NOTES:
1. Yellow areas indicate locations which are *not used* and are *reserved* for future use. *Reserved* locations cannot be modified and alsways return 0's when read. No addresses are labeled *unused*.
2. Implemented memory array size can vary from 2 Kbytes to 32 Kbytes in 2-Kbyte increments.

The register referred to as "FBAR" is actually the FASRAM base address registers FBAR-H and FBAR-L, concatenated to form a long word. These may be treated as two distinct word sized registers.

For each bus cycle from the IMB3 or the FAB, the high order address lines A[23:N] (where N = 11, 12, 13, 14, 15 for a FASRAM of size rounded up to 2, 4, 8, 16, and 32 Kbytes) are compared with the value of the FBAR registers. The function codes are also qualified with the RASP bits in the FMCR for restrictions on the type of access allowed. If these values match, then the low order address lines and the SIZ[1:0] lines are latched and used to access the array.

The FASRAM will only respond to addresses which are within the implemented RAM array size.

EXAMPLE

For a 10-Kbyte FASRAM implementation, FBAR can be set to any address which is a multiple of 16 Kbytes. The RAM array will then be mapped to the addresses which fall between the base address and the base address plus 10 Kbytes. In this example, the FASRAM will not respond to the addresses from the top of the 10-Kbyte RAM array to the next 16-Kbyte boundary.

Accesses to the FASRAM RAM array operate identically, except for bus cycle timing differences, for accesses from both the IMB3 and the FAB. For both buses, the memory resides in the same function code space and at the same address.

4.8.1 FASRAM Module Configuration, Control, and Status Registers

All configuration and control of the FASRAM module can be accomplished through the registers described in this section. To protect these registers from accidental modification, they are mapped to supervisor data space only. In addition, they can be accessed only via the IMB3 and not the FAB. The FASRAM registers will not respond to a user data space access, or a program space access, therefore any user-mode or program-mode access of the registers will be acknowledged by other modules on the chip or by an external device.

While at present only 18 bytes have been used for the control registers, this module decodes a 32-byte register block. Unimplemented registers are read as zeros and writes have no effect.

All of the FASRAM control registers can be accessed by byte-sized or word-sized bus cycles. There are no word-sized-only registers.

The following sections describe the operation of each register in the FASRAM control block. Any restrictions to making changes to the registers will be noted in the description of the register. Unless otherwise noted, a master reset will cause register bits to be forced to their reset state, while a system reset will have no effect on the registers.

4.8.1.1 FASRAM Module Configuration Register (FMCR)

All modules on the IMB3 contain a module configuration register. The FMCR bits configure the FASRAM module for stop operation and for proper access rights to the RAM array. They also control the functionality of the V_{STBY} power and visibility bus (VB). Register bits which are labeled with a “0” are reserved for future use. Reserved locations cannot be modified and always return zeros when read.

FMCR — FASRAM Module Configuration Register **0xYF F6C0**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
STOP	VEN	RESERVED		RLCK	DLY1	RASP		Reserved							
RESET:															
1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

Table 4-2 FMCR Bit Descriptions

Bit(s)	Name	Description
15	STOP	Stop control. Setting the STOP control bit in the FMCR register forces the FASRAM module to enter into the STOP state. When the STOP bit is set, RAM array accesses are ignored. When the STOP bit is cleared, the FASRAM array base address register (FBAR) is write protected. This bit is automatically set when VDD is less than VSTBY, and during the array current test. 0 = FASRAM module is in normal mode of operation 1 = Causes the FASRAM module to enter the low power stop mode
14	VEN	Visibility bus enable. The visibility bus control signals VDS and VWRITE are reconfigured as debug comparator strobe out-put signals. The visibility bus enable bit enables the external visibility bus (VB) to show accesses from the fast access bus (FAB) to the FASRAM RAM array. For more information, see **4.4 Visibility Bus**. When the VB is disabled, the VWRITE and VDS lines act as comparator strobes for the two comparators. For the comparator strobe to be generated, one or both of the comparators must be enabled. For more information, see **4.4.4 Visibility Bus Write Pin (VWRITE)** and **4.4.6 Visibility Bus Data Strobe (VDS)**. 0 = Visibility bus (VB) is driven to an inactive level (V_{SS} or V_{DD}) 1 = Visibility bus (VB) shows the accesses on the fast access bus (FAB)
13:12	—	Reserved
11	RLCK	RAM base address lock. The FASRAM base address registers are writable until the RLCK bit is set. Once RLCK is set, writes can not affect the base address registers. This bit may only be set via the IMB3 and is reset only by a master reset. Once set, this bit may be cleared by software only when the FASRAM is either in test mode or in background mode. This bit may be written any number of times with a ‘0’ until it is written with a ‘1’. 0 = FASRAM base address registers are writable from the IMB3 1 = FASRAM base address registers are write locked
10	DLY1	Bus cycle delay. The FASRAM uses this bit to increase internal array access times by one internal clock (one-half of a system clock). This causes IMB3 cycles to delay completion by one system clock, and FAB cycles to delay completion by one-half of a system clock. This feature may be required depending on supply voltages and memory array implementation. 0 = FASRAM accesses are normal speed 1 = FASRAM accesses are delayed by one-half system clock
9:8	RASP	FASRAM array space. The RAM array is placed either in supervisor or in unrestricted space. When placed in supervisor space, (RASP[1] = “1”), only a supervisor program can access the array. If a user program is attempting to access the FASRAM while it is in supervisor space, the FASRAM will ignore the access. If RASP[1] = “0”, the RAM array is placed in unrestricted space and accesses by both supervisor and user programs are allowed. The RAM array is placed either in program or in program/data space. When placed in program space, (RASP[0] = “1”) only program instructions, or program counter relative addressing modes for operand fetches can be read from the array. All data space accesses to the RAM array will be ignored. If RASP[0] = “0”, the RAM array will respond to both program and data space accesses. Accesses from both the FAB and the IMB3 are affected by the RASP bits. The RASP bits have no effect on control register accesses.. 00 = Unrestricted program and data 01 = Unrestricted program 10 = Supervisor program and data 11 = Supervisor program
7:0	—	Reserved

4.8.2 Array Base Address Registers (FBAR)

The RAM array base address registers (FBAR-H and FBAR-L, referred to collectively as FBAR) are provided to allow the flexibility of placing the RAM array anywhere in the memory map. The FBAR contains an address field used to specify the most significant bits of the lowest addressable value in the RAM array address block. The lower eleven

bits of FBAR (FBAR[10:0]) and the upper eight bits of FBAR (FBAR[31:24]) read as ‘0’ and are not affected by writes.

The restrictions on the mapping of the RAM array are detailed in **4.8 FASRAM RAM Array Addressing / Mapping**.

FBAR-H — Base Address Register **0xYF F6C4**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
Reserved								BAR [A23:A16]							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FBAR-L — Base Address Register **0xYF F6C6**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
BAR [A15:A11]					Reserved										
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4-3 FBAR-H, FBAR-L Bit Descriptions

Bit(s)	Name	Description
FBAR-H 15:8, FBAR-L 10:0	—	Reserved
FBAR-H 7:0, FBAR-L 15:11	BAR [A31:A11]	The RAM array’s base address is contained in the base address field of the array base address registers. With STOP set and RLCK cleared the base address field of FBAR may be changed so that the array may be placed at the desired address in the memory map. This must be done by a supervisor program since the registers are in supervisor data space. To lock the base address field RLCK should be set. This will prevent the base address field from being changed until the next master reset. Once RLCK has been set, it cannot be cleared except either in test mode or in background mode. Supervisor read from IMB3, writable from IMB3 if STOP = 1 and RLCK = 0.

4.9 FASRAM Debugging Features

The FASRAM has a set of control registers in addition to those contained in the standard SRAM module. The purpose of these registers is to control the debugging and development features found in the FASRAM module.

The debugging features consist of two comparators which can be used to detect accesses to addresses within the FASRAM in one of three manners:

- To trigger on two specific addresses
- To trigger on a range of addresses by specifying the top and bottom of the range
- To trigger on one specific address and one range which is bounded by the memory border

One comparator is a greater than or equal to function while the other is a less than or equal to function. The comparators can be used together in their RANGE mode to detect accesses within a range, or they may be used separately to detect an access to one of two specific addresses.

There are also two status-only registers associated with the comparators. The two registers show information relevant to the most recent match address which was detected by either comparator. There is a most recent match address register and a status register.

The FASRAM debugging comparators are disabled after master reset. It should be noted that enabling the comparators will impact system performance in two ways:

1. Processor performance will be somewhat impaired due to breakpoints and IMB3 data accesses where they would otherwise not occur. This can be kept to a minimum by using the debugging comparators wisely, limiting their scope only to vital areas of the FASRAM RAM array space.
2. Slightly more power will be consumed when one or more comparators are enabled.

4.9.1 Comparator Control Registers (FCCR0, FCCR1)

There are two control registers, one for each of the two comparators.

The comparator control registers also contain four bits which control when a compare is to take place, a bit which controls what type of compare to do (range or equals), and a bit which controls the action which will occur when a compare triggers. The comparator is turned off by clearing all four of the control bits. This is the reset state of the control registers. The comparator is effectively disabled when the control bits are cleared. The comparator control registers (FCCR0, FCCR1) shows the position and reset states of these bits.

FCCR0, FCCR1— FASRAM Comparator Value 0,1 Registers **0xYF F6C8**
0xYF F6CA

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
Comparator 0								Comparator 1							
	RANGE	FRCI	USER	SUPV	READ	WRITE		0	0	RANGE	FRCI	USER	SUPV	READ	WRITE
RESET: S															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

One or both of the comparators should be enabled by writing to the comparator control registers. At least one of the READ or WRITE bits as well as at least one of the USER or SUPV bits must be set to enable a given comparator. These bits determine what type of access will trigger the comparator. Register bits which are labeled with a “0” are reserved for future use. Reserved locations cannot be modified and always return zeros when read.

Table 4-4 FCCR0, FCCR1 Bit Descriptions

Bit(s)	Name	Description
13, 5	RANGEI	Type of compare to perform. The RANGE control bits set a specific comparator to the range compare mode. This means that the comparator will now "trigger" when an access is "≤" or "≥" depending on which comparator is used. If RANGE = "0" then the comparator will act as an "==" comparator. When the RANGE bits for both comparators are set, the two comparators work together as a range detection where an access must "trigger" both comparators to cause an action. This is useful when the user wishes to detect accesses between two arbitrary addresses within the FASRAM, neither of which is a RAM array boundary. If only one of the comparators' RANGE bits is set then the associated comparator must be used to specify a range of addresses whose other bound is the end of the RAM array.
12, 4	FRCI	Setting actions resulting from a compare 'trigger'. The debugging comparators have two possible actions they can perform on an access to the FASRAM. 0 = Comparators cause a breakpoint to occur on the memory access 1 = Force access to be performed via the IMB3
11, 3	USER	Compare on a user access. These control bits enable the specific comparator to detect a user access to a FASRAM RAM array location via the FAB. SUPV may also be set. READ and / or WRITE must be set.
10, 2	SUPV	Compare on a supervisor access. These control bits enable the specific comparator to detect a supervisor access to a FASRAM RAM array location via the FAB. USER may also be set. READ and / or WRITE must be set.
9, 1	READ	Compare a read access. These control bits enable the specific comparator to detect a read access to a FASRAM RAM array location via the FAB. WRITE may also be set. SUPV and / or USER must be set.
8, 0	WRITE	Compare a write access. These control bits enable the specific comparator to detect a write access to a FASRAM RAM array location via the FAB. READ may also be set. SUPV and / or USER must be set.

4.9.2 Finding the Status of the Most Recent Compare

The FASRAM has two status registers which give the user information about the most recent comparator "trigger" which was generated. This information includes the following:

- Which comparator had the most recent match (WHICH)
- Which comparator had a match (MATCH1, MATCH0)
- The size of the access (SIZE)
- Was the access a read or a write (RWSTAT)
- Was the access from data or program space (DPSTAT)
- Was the access from supervisor or user space (SUSTAT)
- The actual address which caused the "trigger" (most recent match address)

4.9.3 Most Recent Match Address (FMATCH)

The most recent match address contains the address of the last match made by either comparator. The FMATCH register is affected when a breakpoint is caused, not for a forced IMB3 cycle. Although bits 15 and 0 are not used in the address comparison, they are captured in this register. The WHICH bit in the STATUS register indicates which comparator made the most recent match. This register is read-only.

If both comparators trigger before the FMATCH register can be read, the value in the FMATCH register will be for the last trigger. This may lead to software becoming confused as to which comparator the FMATCH register represents.

Reset has no effect on this register.

FMATCH — Most Recent Match Address Register **0xYF F6CE**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
RESET: S															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.9.4 Most Recent Match Status (FSTATUS)

The STATUS register contains all pertinent information about the most recent comparator match, if and only if that match caused a breakpoint. All bits in the FSTATUS register are read-only with the following exception. A MATCH bit (FSTATUS[6:5]) is clearable by writing it to a zero. This can only be done following a read to FSTATUS (address YMMFFB11). A read to the adjacent byte “not used[15:8] (reserved)” (address YMMFFB10) will have the same affect. A write of a one has no effect. Most status bits in the FSTATUS register have no reset state, with the exception that the MATCH bits are cleared. FSTATUS[15:8] and FSTATUS[4] are unused and reserved. This means writes have no effect and reads always return zeros.

FSTATUS— FASRAM Most Recent Match Status Register **0xYF F610**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
Reserved								WHICH	MATCH1	MATCH0	Reserved	SIZE	RWSTAT	DPSTAT	SUSTAT
RESET: S															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4-5 FSTATUS Bit Descriptions

Bit(s)	Name	Description
15:8	—	Reserved
7	WHICH	Most recent comparator match. The WHICH bit indicates which of the two comparators most recently had a match, provided a breakpoint action was caused. The WHICH bit is cleared if comparator zero had a match and is set if comparator one had a match. The WHICH bit is undefined if both RANGE bits are set, or if both comparators trigger on the same access.
6:5	MATCH1, MATCH0	Compare on a user access. The MATCH bits in the FSTATUS register indicate whether the associated comparator has been “triggered”. The MATCH bits are only set when a comparator trigger causes a breakpoint (FCRI = 0). Once these bits are set via a compare trigger, they will remain set until cleared by a master reset or by a supervisor write clearing these bits in the FSTATUS register. These bits may be written to a zero only if they have been read while set to one. This allows for proper determination of which comparator has breakpointed when the second comparator causes a breakpoint after the register has been read for a first breakpoint.
4	—	Reserved
3	SIZE	Size of the access. The SIZE bit in the FSTATUS register indicates the size of the most recent access for which the comparator has a match, provided a breakpoint action was caused. 0 = Access was a word access 1 = Access was a byte access
2	RWSTAT	Access a read or a write. The RWSTAT bit indicates whether the most recent comparator match was caused by a read or by a write access, provided a breakpoint action was caused. 0 = Match was caused by a write 1 = Match was caused by a read
1	DPSTAT	Access from data or program space. The DPSTAT bit indicates whether the most recent comparator match was caused by a data (or by a program space access, provided a breakpoint action was caused. The CPU32x is not capable of making a program space access over the FAB, therefore this bit will always be a one for that CPU. 0 = Most recent comparator match was caused by a program space access 1 = Most recent comparator match was caused by a data space access
0	SUSTAT	Access from data or program space. The SUSTAT bit indicates whether the most recent comparator match was caused by a supervisor or by a user space access, provided a breakpoint action was caused. 0 = Most recent comparator match was caused by a user access 1 = Most recent comparator match was caused by a supervisor space access

4.10 Bus Connections

The following paragraphs describe the various bus connections to the FASRAM.

4.10.1 Inter-Module Bus

The intermodule bus (IMB3) is an internal, bi-directional two-clock bus with the data for one cycle overlapping the address for the next cycle. The FASRAM will respond to an IMB3 request in a minimum of two clock periods, but may be slower if the FAB is currently accessing the RAM array.

4.11 Fast Access Bus

The fast access bus (FAB) is an internal, bidirectional one-clock interface to the CPU. It is a miniature IMB3. The FASRAM will respond to a FAB request in a minimum of one clock period, but may be slower if the IMB3 is currently accessing the FASRAM.

4.12 Visibility Bus

The visibility bus (VB) is the output-only bus that shows the external world the activity on the fast access bus. There are three bond-out options: the entire VB, just the write pin and the data strobe, or none.

The address and data for the FAB cycle are buffered internally until the bus cycle is complete. Then the VB is driven and the strobes asserted. This eliminates any variable timing due to the differences of the data availability between a read cycle and a write cycle, or between the address and data. No indication of a long cycle is given.The actual pin names and number of implemented bits are specified in the chip plan. See **Table 4-6**.

Table 4-6 Visibility Bus Pins

Pin	I/O	Size	Active	Function
VADDR (14:0)	O	15[1]	N/A	Address of the FAB cycle. Number of implemented pins depends on total memory size.
VDATA (15:0)	O	16	N/A	Data for the FAB cycle.
VSIZE	O	1	N/A	Size of the FAB cycle. Word = 0, Byte = 1
VWRITEB or VWRITE	O	1	Low	Direction of the FAB cycle. Read = 1, Write = 0. If the VB is disabled, then this is the comparator zero strobe.
VDSB or VDS	O	1	Low	External logic should latch the VB. If the VB is disabled, then this is the compara-tor one strobe.

NOTE

1. The number of address pins depends on the size of the RAM array. For example, 13 pins would be used for an 8-Kbyte array.

4.13 Electrical Characteristics

For electrical characteristics, please refer to **APPENDIX E ELECTRICAL AND AC CHARACTERISTICS**.