SECTION 12
MASK ROM MODULE

12.1 Introduction

The mask ROM module for the Modular Embedded Controller Family is designed to be used with the family's inter-module bus (IMB3) and consequently any CPU capable of operating on it. The mask ROM module implementation for the MC68F375 is 8,192 (8K) bytes.

The array is arranged in a 16-bit configuration and is accessed via the device's internal bus. It may be read as either bytes, aligned words or misaligned words. Access times depend on the number of WAIT states specified at mask programming time, but can be as fast as 2 system clocks for byte and aligned word access. It is also capable of responding to back-to-back IMB3 accesses to provide 2 bus cycle (4 system clocks) access for aligned long word or misaligned word operations, and 3 bus cycles for misaligned long words. The ROM module may be used to contain program information only, or both program and data information.

The ROM module can be used as fast access memory to contain program code which must execute at high speed, or which gets executed often. Operating system kernels and standard subroutines benefit from this fast access time. It can also be programmed to insert WAIT states to accommodate migration from slower external development memory to on-chip ROM, without the need for retiming the system. The ROM module may be configured to generate bootstrap information on RESET, without the array being mapped to location 0x000000.

The ROM module can also operate in a special emulation mode, which simplifies emulation of the internal ROM by an external device, when used with a system integration module which makes use of the ICSMB IMB3 line.

12.2 Mask Programmable Options

Along with the contents of the ROM array, several configuration options must be specified by the customer. These options are mask programmed on the same mask layer as the contents of the array. The options comprise:

- Default reset state of the base address of the array.
- Default reset state of the $\overline{\text{BOOT}}$ control bit which determines if the ROM responds to bootstrap addresses.
- Default reset state of the LOCK control bit which controls write access to configuration registers.
- Default reset state of the WAIT field which controls the number of clocks for ROM accesses.
- Default reset state of the ASPC field which specifies the address space of the ROM array.

- The value of SIGHI and SIGLO which is the ROM signature pattern.
- The default values for the ROM bootstrap information words, ROMBS[0:3]."

12.3 Programmer's Model

The ROM module consists of two separately addressable sections. The first is a memory mapped control register block used for control and configuration information of the ROM module. The second section is the array itself.

12.3.1 ROM Control Block

A 32-byte control block contains registers which are used to control ROM module operation and provide configuration information about the ROM pattern contained in the module. Configuration information is specified and programmed at the same time as the contents of the ROM array and on the same mask layer.

The configuration information contained in this block includes array base address information, bootstrap information and ROM verification information. Control bits are provided to control array operation. A 19-bit field in the control block section contains a signature used for identification and verification of the particular ROM pattern contained in the ROM array, see **Table 12-1**.

12.3.1.1 ROM Module Control Block Addressing

The control block is restricted to supervisor data space. Unimplemented or reserved addresses will return 0's for read accesses. Write accesses to unimplemented or reserved control block addresses will have no effect. Accesses to unimplemented or reserved locations will result in bus error (IBERR) being asserted.

12.3.2 ROM Array

The base address of the memory block which the ROM array resides in is specified in the array base address registers.

The default reset address of the ROM array in the address map of the system is specified by the customer at ROM programming time. The only restrictions on the base address is that it must be on a 8-Kbyte boundary and not overlap the module control register block in the data space memory map. Accesses to unimplemented locations in the address block will be ignored, allowing another internal module or external device to respond.

If the base address is set such that the ROM array overlaps the control register block of the ROM, accesses to the 32 bytes in the array that overlap will be ignored, allowing the control block to remain accessible.

Note that this is only true with respect to the ROM module. If the control register blocks of other modules are overlapped by the ROM array, accesses to the overlapped addresses of other modules will be indeterminate.

12.3.2.1 ROM Array Addressing

The array may be specified to reside in supervisor space to restrict access to supervisor only, or it may be specified to exist in unrestricted space to allow access by both user and supervisor programs. The array may also be configured to respond to program space accesses only or to both data and program space accesses.

The BIU of the ROM module compares internal address [23:13] of the IMB3 with ROMBAH, ROMBAL[23:13]. If they match, the remaining address bits and ISIZ[1:0] are used to access the ROM location in the array. Function codes are also checked for the correct access rights. If the array is specified to exist in supervisor space, user accesses will be ignored allowing an external device to respond to the address. If the array is specified to exist in unrestricted space, it will respond to both user and supervisor accesses. If the array is specified to exist in program space only, data space accesses to the array will be ignored allowing implementation of separate data and program space address maps."

12.4 ROM Module Control and Configuration Registers

This section describes the ROM control and configuration registers. Each register field is described in detail with regard to operation. The ROM module register map is shown in **Table 12-1**.

Table 12-1 ROM Module Register Map

Access	Address[1]	15 8	7 0
S	0xYF F820	ROM Module Configuration Register (ROMMCR) See **Table 12-2** for bit descriptions.	
S	0xYF F824	ROM Base Address High Register (ROMBAH) See **Table 12-3** for bit descriptions.	
S	0xYF F826	ROM Base Address Low Register (ROMBAL) See **Table 12-3** for bit descriptions.	
S	0xYF F828	ROM Signature High Register (SIGHI) See **Table 12-3** for bit descriptions.	
S	0xYF F82A	ROM Signature Low Register (SIGLO) See **Table 12-3** for bit descriptions.	
S	0xYF F830, 0xYF F832, 0xYF F834, 0xYF F836	ROM Bootstrap Information Words (ROMBS[0:4])	

NOTES:
1. Y = M111, where M is the logic state of the module mapping (MM) bit in the SCIM2E configuration register (SCIMMCR).

12.4.1 Module Configuration Register (ROMMCR)

The ROM module configuration register is used to control the operation of the ROM array and provide status information. It also provides the production tester and the customer with configuration information.

ROMMCR — ROM Module Configuration Register **0xYF F820**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
STOP	RESERVED		BOOT	LOCK	EMUL	ASPC[1:0]{2}		WAIT[1:0]{3}		RESERVED					
RESET:															
0	0	0	U[1]	U[1]	0	U[1]	U[1]	U[1]	U[1]	0	0	0	0	0	0

NOTES:
1. The default state of this bit is defined by customer specified options.
2. Indicates bits protected by LOCK and STOP.
3. Indicates bits protected by LOCK only.

Table 12-2 ROMMCR Bit Settings

Bit(s)	Name	Description
15	STOP	Inverted state of D[14]. If STOP is set to a 1 by the inverted D[14] during master reset, the array may be re-enabled by clearing STOP after master reset. If bootstrap mode is enabled, ($\overline{\text{BOOT}}$ = 0), clearing STOP will not cause the ROM to enter bootstrap mode. The STOP bit must be set in order to change the value of the array base address (ROMBAH, ROMBAL), the state of the EMUL control bit, or the value of the ASPC field in ROMMCR. Clearing the STOP bit also deactivates emulation mode, but does not disable it. 0 = The ROM module is in normal mode of operation. 1 = Causes the ROM module to enter STOP mode.
14:13	—	Reserved
12	$\overline{\text{BOOT}}$	Bootstrap enable. At mask programming time, the ROM module may be specified to function as a bootstrap ROM after RESET or only as a ROM array at a specified base address. The $\overline{\text{BOOT}}$ bit is forced to its default reset state by master reset. The default reset state of the $\overline{\text{BOOT}}$ bit is specified by the user at mask programming time and is programmed on the same mask layer as the contents of the array. 0 = ROM module will respond to the bootstrap addresses after RESET. 1 = ROM module will not respond to the bootstrap addresses after RESET.
11	LOCK	Lock registers. Once the LOCK bit is set, via an IMB3 write, it cannot be cleared again until after a master reset. ***If the default reset state is 1, all registers and bits protected by the LOCK bit can never be changed.*** The LOCK bit is forced to its default reset state by master reset. The default reset state of the LOCK bit is specified by the user at mask programming time and is programmed on the same mask layer as the contents of the array. To ensure that inadvertent reconfiguration of the ROM cannot occur, the user's initialization program should always write this bit to a one to invoke the write lock mechanism, if it's default reset state is 0. 0 = Write lock is disabled. 1 = Write-locked registers are protected.
10	EMUL	Emulation mode. Emulation mode allows the ROM array to be emulated externally, with access controlled by the ROM module BIU. Bootstrap operation is not affected by emulation mode, nor is it emulated. Emulation mode may be entered by writing EMUL to a 1. The STOP bit in the ROMMCR must be a 1 in order to change the state of the EMUL bit by writing to it via the IMB3. In emulation mode the ROM will not respond to an array address with IAACKB, however, it will respond to control register accesses. Instead of responding with IAACKB to array addresses, it will assert the ICSMB line on the IMB3, allowing the device's external bus interface to assert an external chip select and run a special external bus cycle which is terminated by the ROM asserting IDTACKB on the IMB3. The ROM will assert IDTACKB after the proper number of WAIT states, as specified in the WAIT field of the ROMMCR register. The ROM module will not drive the IMB3 data lines however. Data must be provided by an external device at the proper time. See **4.7.8.6 Emulation Mode Selection**. 0 = The ROM module is in normal mode of operation. 1 = Causes the ROM module to enter emulation mode.

Table 12-2 ROMMCR Bit Settings (Continued)

Bit(s)	Name	Description
9:8	ASPC[1:0]	ROM array space. The ASPC[1:0] field is forced to the default reset state by master reset. The default reset state of the ASPC[1:0] field is specified by the user at mask programming time and is programmed on the same mask layer as the contents of the array. IFC[2:0] is checked by the ROM module BIU to determine if a user or supervisor is requesting array access. If a supervisor program is accessing the array, normal read operation will occur. If a user program is attempting to access the array, the access will be ignored and the address may be decoded externally. 00 = Unrestricted program and data space IFC[2:0] = x01, x10. 01 = Unrestricted program space only. IFC[2:0] = x10. 10 = Supervisor data and program space only. IFC[2:0] = 101, 110. 11 = Supervisor program space only. IFC[2:0] = 110.
7:6	WAIT[1:0]	Wait states. The WAIT field is used to specify the number of WAIT states inserted by the ROM BIU during accesses to the ROM module before asserting IDTACKB. A WAIT state has a duration of one system clock cycle. This affects both control block access and array access. This feature allows the migration of storage space from a slower emulation or development system memory to the onboard ROM module without the need for retiming the system. The default reset state of the WAIT field is specified by the user at mask programming time and is programmed on the same mask layer as the contents of the array. The WAIT field may be written any time the LOCK bit is 0. 00 = 3 clocks per transfer. 01 = 4 clocks per transfer 10 = 5 clocks per transfer 11 = 2 clocks per transfer
5:0	—	Reserved

12.4.2 ROM Base Address Register (ROMBAH, ROMBAL)

The default reset ROM base address fields are specified by the user along with the contents of the array and are programmed on the same mask layer as the contents of the array.

ROMBAH — ROM Base Address High Register **0xYF F824**

MSB 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	LSB 16
RESERVED								ROMBAH[1]							
RESET:															
0	0	0	0	0	0	0	0	U[2]	U[2]	U[2]	U[2]	U[2]	U[2]	U[2]	U[2]

NOTES:
1. Indicates bits protected by LOCK and STOP. The default state of these bits is defined by customer-specified options.
2. The default state of these bits is defined by customer specified options.

ROMBAL — ROM Base Address Low Register **0xYF F826**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
ROMBAL[1]				RESERVED											
RESET:															
U[2]	U[2]	U[2]	U[2]	0	0	0	0	0	0	0	0	0	0	0	0

NOTES:
1. Indicates bits protected by LOCK and STOP. The default state of these bits is defined by customer-specified options.

2. The default state of these bits is defined by customer specified options.

Table 12-3 BAR (ROMBAH, ROMBAL) Bit Settings

Bit(s)	Name	Description
31:24	—	Reserved
23:16	ROMBAH	The 32-bit base address of the ROM array memory address block is contained in the ROMBAH and ROMBAL array base address registers. The base address register (BAR) is formed by concatenating the contents of ROMBAH and ROMBAL. ROMBAH contains the high order 16 bits of the address (A[31:16]) and ROMBAL contains the next lower order bits. The value of ROMBAH and ROMBAL are forced to the user defined value programmed in ROM shadow registers on master RESET. ROMBAH and ROMBAL can be written to relocate the ROM array to an alternate block of memory only when the LOCK bit is 0 and the ROM module is in STOP mode..
15:12	ROMBAL	
11:0	—	Reserved

12.4.3 ROM Signature High (SIGHI)

SIGHI — ROM Signature High Register **0xYF F828**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
RESERVED													RSP18	RSP17	RSP16
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	U[1]	U[1]	U[1]

NOTES:
1. The default state of these bits is defined by customer specified options.

Table 12-4 SIGHI Bit Settings

Bit(s)	Name	Description
15:3	—	Reserved
2:0	RSP[18:16]	ROM signature pattern. These 3 bits, when concatenated with the 16 bits contained in SIGLO, form a 19-bit unique signature used to verify the contents of the ROM array. This information is programmed along with the contents of the array, on the same mask layer.

12.4.4 ROM Signature Low (SIGLO)

SIGLO — ROM Signature Low Register **0xYF F82A**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
RSP15	RSP14	RSP13	RSP12	RSP11	RSP10	RSP9	RSP8	RSP7	RSP6	RSP5	RSP4	RSP3	RSP2	RSP1	RSP0
RESET:															
U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]

NOTES:
1. The default state of these bits is defined by customer specified options.

Table 12-5 SIGLO Bit Settings

Bit(s)	Name	Description
15:0	RSP[15:0]	ROM signature pattern. These 16 bits, when concatenated with the 3 bits contained in SIGHI, form a 19-bit unique signature used to verify the contents of the ROM array. This information is programmed along with the contents of the array, on the same mask layer.

12.5 Bootstrap Information Words (ROMBS0–ROMBS3)

Typically, reset vectors for the system CPU are contained in non-volatile memory and are only fetched when the CPU comes out of reset. The bootstrap information for the processor controlling the system are contained in the ROM module in the four words ROMBS0–ROMBS3. ROMBS0 responds to address 0x000000, ROMBS1 responds to 0x000002, ROMBS2 to 0x000004 and ROMBS3 to 0x000006 on the IMB3. The bootstrap information is specified by the user along with the contents of the array and is programmed along with the contents of the array, on the same mask layer. These registers are read only from the IMB3 in normal mode or bootstrap mode. IMB3 writes do not affect the contents of these registers. In bootstrap mode, ROMBS0–ROMBS3 only respond to supervisor program space accesses. In normal mode, they only respond to supervisor data space accesses.

ROMBS0 — ROM Bootstrap Word 0 **0xYF F830**

MSB 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	LSB 16
SP[31:16]															
RESET:															
U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]

NOTES:
1. The default state of these bits is defined by customer-specified options.

ROMBS1 — ROM Bootstrap Word 1 **0xYF F832**

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

NOTES:
1. The default state of these bits is defined by customer-specified options.

ROMBS2 — ROM Bootstrap Word 2 **0xYF F834**

MSB 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	LSB 16
PC[31:16]															
RESET:															
U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]	U[1]

NOTES:
1. The default state of these bits is defined by customer-specified options.

ROMBS3 — ROM Bootstrap Word 3 **0xYF F836**

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

NOTES:
1. The default state of these bits is defined by customer-specified options.

12.6 Operation

The ROM module is accessed via the IMB3 by a bus master. It can be used to contain either program information only or both data and program information. On master reset, it can operate as a bootstrap ROM to provide CPU internal initialization information during the CPU's reset sequence or can be configured to never respond to the bootstrap addresses.

12.6.1 RESET Operation

The ROM uses master RESET to initialize all register bits to their reset values. The LOCK bit will also be cleared if it's default reset state is 0. During master reset, the ROM BIU will monitor three inputs to the module (STOPIN, EMULIN, and EMULEN) to determine if it should respond normally after reset or disable itself for testing purposes, and if emulation mode is enabled.

The value of STOPIN is determined by the state of D[14] during master reset. If the state of STOPIN is 1, the STOP bit in the ROMMCR register will be cleared to 0 and the array will respond normally to the bootstrap address range and the ROM array base address. If STOPIN is 0, the STOP bit will be set and the ROM array will be disabled until the STOP bit is cleared either by an IMB3 write or until the next master reset which occurs with STOPIN = 1. It will not respond to the bootstrap address range or the ROM array base address in BAR (ROMBAH and ROMBAL), allowing an external device to respond to the ROM array's address space, and/or provide bootstrap information. This allows the ROM to be disabled from outside of the device if necessary.

The value of EMULIN is determined by the state of D[10] and the value of EMULEN is determined by the state of D[13] during master reset. If the state of either EMULIN or EMULEN is 1, the EMUL bit in the ROMMCR register will be cleared to 0, ROM emulation mode will not be enabled, and the array will respond normally to valid accesses. If EMULIN and EMULEN are both 0, the EMUL bit in ROMMCR will be set and ROM emulation mode will be enabled until the EMUL bit is cleared by either an IMB3 write or the next master reset occurs with either EMULIN or EMULEN =1.

STOPIN, EMULIN and EMULEN are forced to the value of external pins during master reset. These pins may be data pins for devices that have an external data bus, in which case STOPIN, EMULIN and EMULEN will be driven by corresponding IMB3 lines. This function is performed by the SCIM2E.

Bootstrap mode will be disabled if STOP is set during master reset, regardless of the default reset state of $\overline{\text{BOOT}}$.

12.6.2 Bootstrap Operation

If $\overline{\text{BOOT}}$ = 0 in the ROMMCR, the ROM module may be used as a bootstrap ROM by the system. The ROM module will only respond to the bootstrap addresses if the STOP control bit in the ROMMCR register is a 0. If STOP is a 1, bootstrap addresses will be ignored. Subsequently clearing the STOP bit will not cause the ROM to enter bootstrap mode.

In bootstrap mode, the ROM module will respond only to the bootstrap addresses in supervisor program space, and provide the information contained in the ROMBS0–ROMBS3 registers. Bootstrap mode will terminate automatically after the last word of bootstrap information, ROMBS1 or ROMBS3 is fetched. On the next system clock, the ROM module will begin responding to control block and array addresses only. If $\overline{\text{BOOT}}$ = 1, the ROM module will only respond to its control block address and the array base address specified in ROMBAH and ROMBAL.

12.6.3 Normal Operation

The control registers and the array may be accessed via the IMB3 as byte, and aligned or misaligned word. The ROM array will respond to read operations only. Write operations will be ignored. **12.6.5 Emulation Mode Operation** describes the read/write operation in emulation mode.) If ASPC[1] =1, the ROM will only respond to supervisor mode reads. If ASPC[1] = 0, the ROM will respond to both supervisor and user reads. If ASPC[0] = 1, the array will respond only to program space accesses. If ASPC[0] = 0, the ROM array will respond to both program and data space accesses.

Access to any address which falls between the last byte of the array and the end of the address block containing the array (specified by ROMBAH and ROMBAL), will be ignored, allowing external devices or other modules to adjoin the ROM array in the address map.

Accesses to the ROM array are only allowed if the STOP control bit in the ROMMCR register is a 0. If STOP is a 1, ROM array accesses and bootstrap accesses will be ignored. This allows an external device, or another internal memory module, to respond to the array address range.

12.6.4 Read/Write Access

The ROM module allows a byte or aligned word read/write in one IMB3 bus cycle. Long word read/write or misaligned word read/write will require an additional bus cycle. Misaligned long word read/write will require a total of 3 bus cycles.

An IMB3 bus cycle requires 2 system clocks minimum. See **SECTION 4 SINGLE-CHIP INTEGRATION MODULE 2 (SCIM2E)** for detailed timing of read/write accesses.

Write operations are only allowed to the control block. Write accesses to the array will be ignored (IAACKB will not be asserted) unless the ROM is in emulation mode and the device is in FREEZE mode. **12.6.5 Emulation Mode Operation** describes array read/write operation in emulation mode.)

12.6.5 Emulation Mode Operation

The ROM module has a special emulation mode which allows external emulation of the ROM array, with the module itself performing address decode and bus cycle termination for the external memory which replaces the array. Bootstrap operation is not affected by emulation mode, nor is it emulated.

NOTE

To emulate bootstrap operation, the emulation system should monitor another chip select which is programmed to assert CSBOOT on the SCIM2E, or the emulation system must monitor the external bus to determine when a bootstrap vector is being requested by the CPU.

STOP affects array operation in emulation mode, the same way it does in normal operation. Accesses to the ROM module control registers are unaffected by emulation mode.

Emulation mode is enabled by the EMUL bit in ROMMCR. EMUL can be written via the IMB3 if STOP = 1, and it's reset state is controlled by the state of the EMULIN and EMULEN inputs to the module during master reset. EMULIN and EMULEN are driven from external pins during RESET. If both EMULIN and EMULEN are low during reset, the EMUL bit will be set in ROMMCR and ROM emulation mode will be enabled. If either input is high during reset, EMUL is cleared and ROM emulation mode is disabled.

EMULIN should be connected to a signal which is used to select whether or not the ROM module should be placed in emulation mode, along with the external bus interface. EMULEN should be connected to the same signal used to enable emulation mode for the external bus interface. In this way, the ROM module cannot be placed in emulation mode, unless the external bus interface is also in emulation mode. It also allows the external bus interface to be placed in emulation without placing the ROM module in emulation mode.

In emulation mode, the ROM will assert the ICSMB line on the IMB3 whenever an access is made to a valid ROM array location, instead of asserting IAACKB. Valid array access corresponds to an array location within the address range indicated by the value in the ROMBAH and ROMBAL registers, and FC[2:0] indicate address space requirements specified in ASPC[1:0] field of the ROMMCR register. The ROM will assert ICSMB on all read accesses and will also assert ICSMB on write accesses if the IFREEZE line on the IMB3 is asserted. The ROM will not assert the IAACKB line if it asserts ICSMB. In response to ICSMB, the external bus interface of the device will assert CSM on the SCIM2E, indicating an emulation mode access to the array, and will run an special external bus cycle which will be terminated by the internal signal IDTACKB instead of DSACKx. The ROM module will assert IDTACKB to terminate the

cycle at the proper time, based on the number of WAIT states programmed into the WAIT field in the ROMMCR register. The ROM will not drive the IMB3 data bus, since the data for the cycle will come from the external data pins of the device.

Table 12-6 Minimum ROM Module Access Times

Type of Access	Bus Cycles for Read[1]	Number of System Clocks
Byte	1	2
Aligned Word	1	2
Misaligned Word	2	4
Aligned Long Word	2	4
Misaligned Long Word	3	6

NOTES:

1. Access time is shown for 2 clock access, (WAIT[1:0] = 11). An additional clock must be added for each additional WAIT state programmed in WAIT.

12.6.6 Stop Operation

If the Stop bit is asserted, the ROM module will not respond with IAACKB to any attempts to access the array or the bootstrap information in bootstrap mode. Only the control register block may be accessed at its normal address. The ROM module must be in STOP mode in order to allow the EMUL control bit to be changed via an IMB3 write. The ROM module must be in STOP mode and LOCK = 0 in order to allow the ROMBAH and ROMBAL registers, and the ASPC field of ROMMCR to be written. STOP also disables bootstrap mode if it set during master reset.

12.6.7 FREEZE Operation

Although the ROMMCR register does not contain a FREEZE mode control bit, FREEZE mode will affect ROM emulation mode operation. The ROM module monitors the IFREEZE line on the IMB3 when the ROM is in emulation mode. If IFREEZE is in its asserted state when the ROM is in emulation mode, the ROM module will respond to write accesses to the array as well as read accesses, see **12.6.5 Emulation Mode Operation**