SECTION 3
BURST INTEGRATION MODULE (BIM)

3.1 Introduction

3.1.1 Features

The burst integration module (BIM) provides two-fold improvement in instruction bandwidth, as compared to previous integration modules, by utilizing a synchronous burst protocol. BIM applications also realize decreased memory costs due to a nearly one-half clock speed improvement in memory access timing.

The BIM has a very low number of pins which are required for basic operation and test. It has the ability to support additional incremental features if pins are available, without the need to redesign the BIM. The pins required to support the incremental features are shown in **Figure 3-1** and denoted by an asterisk (*). Incremental features are configured by external (from the module) connections to optional pads, or jumpers if pads are not available for the function.

The BIM also has the capability to control dual voltage drivers. When a pin is configured as its primary function, then the BIM will enable a 3-V driver in the associated pad. When a pin is configured as digital I/O, then the BIM will enable a 5-V driver with slow rise time in the associated pad to minimize electromagnetic interference (EMI) in the application.

3.1.2 Sub Modules

Figure 3-1 is a BIM module block diagram.

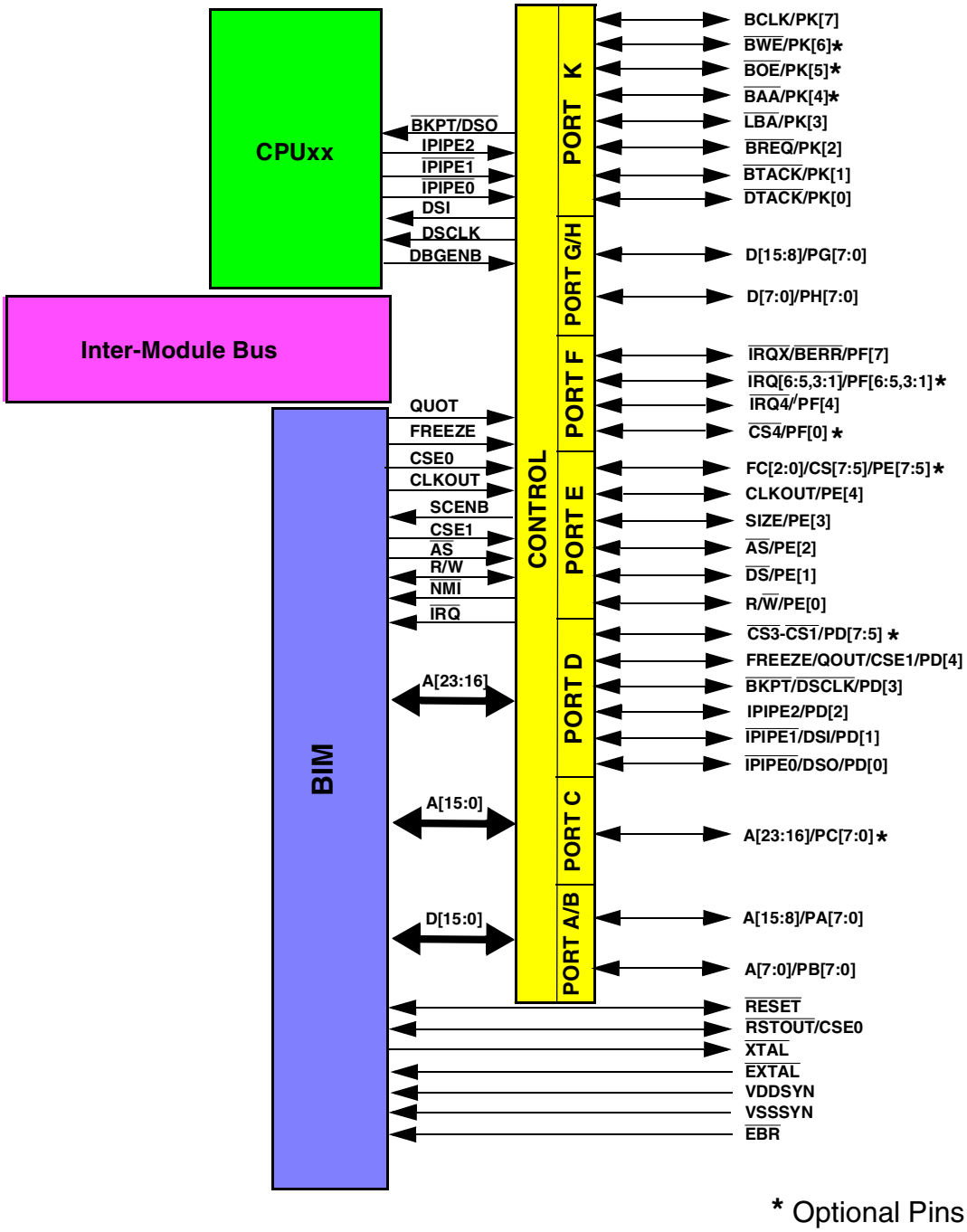


Figure 3-1 BIM Block Diagram

3.1.3 Clocks

The BIM provides the system clocks for the device on which it is implemented, as well as an external clock signal (CLKOUT). The system clocks are generated from the internal phase locked loop (PLL) at a frequency which is derived from a reference fre-

quency. The reference frequency source can be either the crystal oscillator reference or an external clock reference. The BIM provides the user with several options to control the clock frequency of the host device and external clocks, both in normal operation and standby modes. A complete description of the clock circuit is included in **3.4 Clocks**.

3.1.4 BIM Operation

The BIM can operate in master mode, single chip mode, or emulation mode. The BIM's operating mode is determined at reset and thereafter cannot be changed.

Master mode operation allows the internal CPU to access both internal and external memories and peripherals. The external bus consists of a 16-bit data bus, 16-24 address lines, and optional function code lines. Available bus control signals include $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, SIZE, $\overline{BREQ}$, $\overline{BTACK}$, $\overline{BERR}$ and $\overline{DTACK}$. With the burst chip select (BCS) interface, an external burst memory can be selected to increase the performance of the system. The asynchronous chip selects (ACS) can be programmed to select and control external devices, and provide bus cycle termination. Seven interrupt levels are supported with up to seven external interrupt pins. A number of the external pins can be configured for digital I/O. This configuration is shown in **Figure 3-2**.

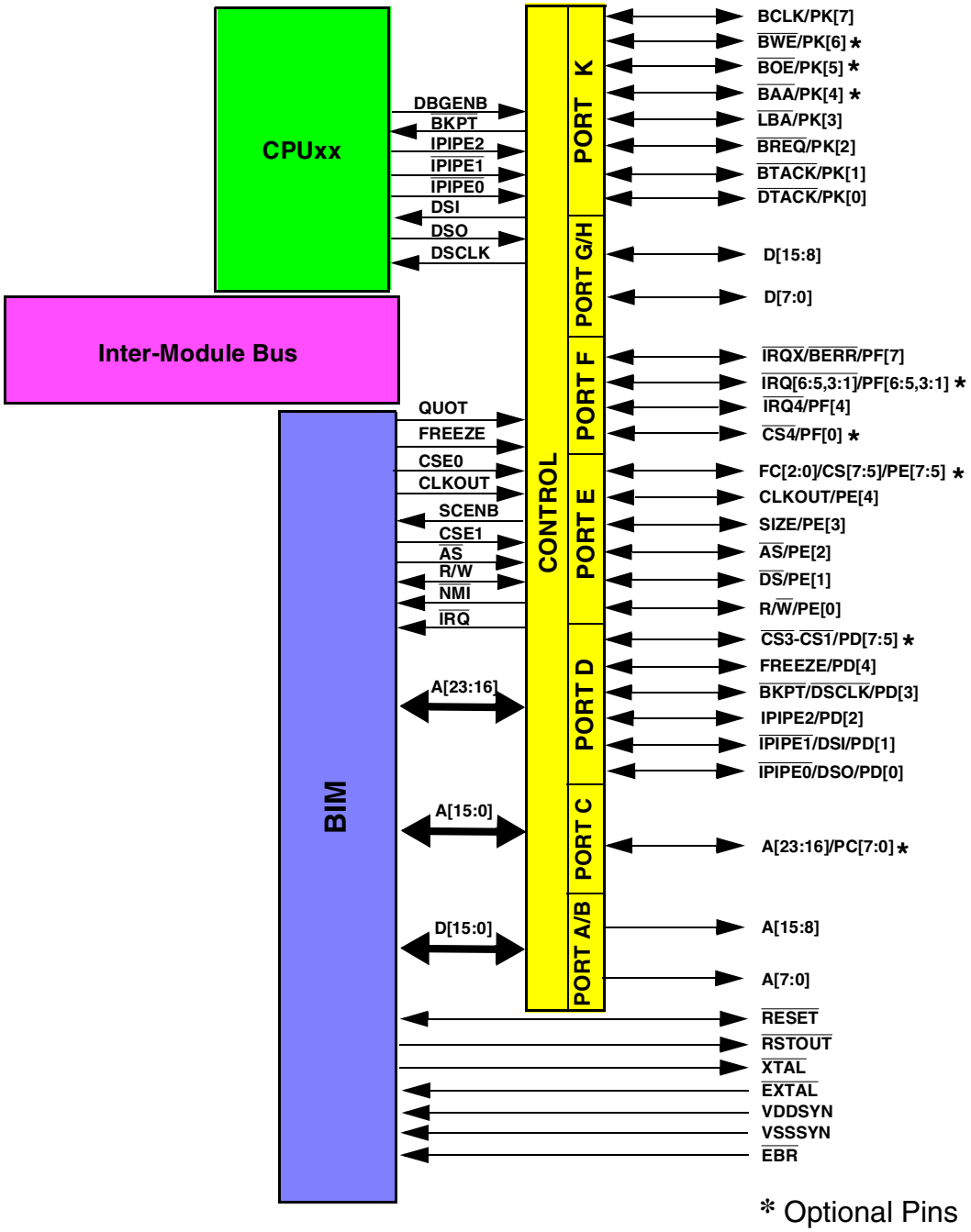


Figure 3-2 BIM Master Mode Configuration

In single chip mode, the CPU and system memory are on chip. External bus pins are either configured as specified by the PCON options or as digital I/O as specified in

Table 3-5.Seven interrupt levels are supported with up to seven external interrupt pins. CLKOUT can be used to provide a system clock. This configuration is shown in **Figure 3-3**.

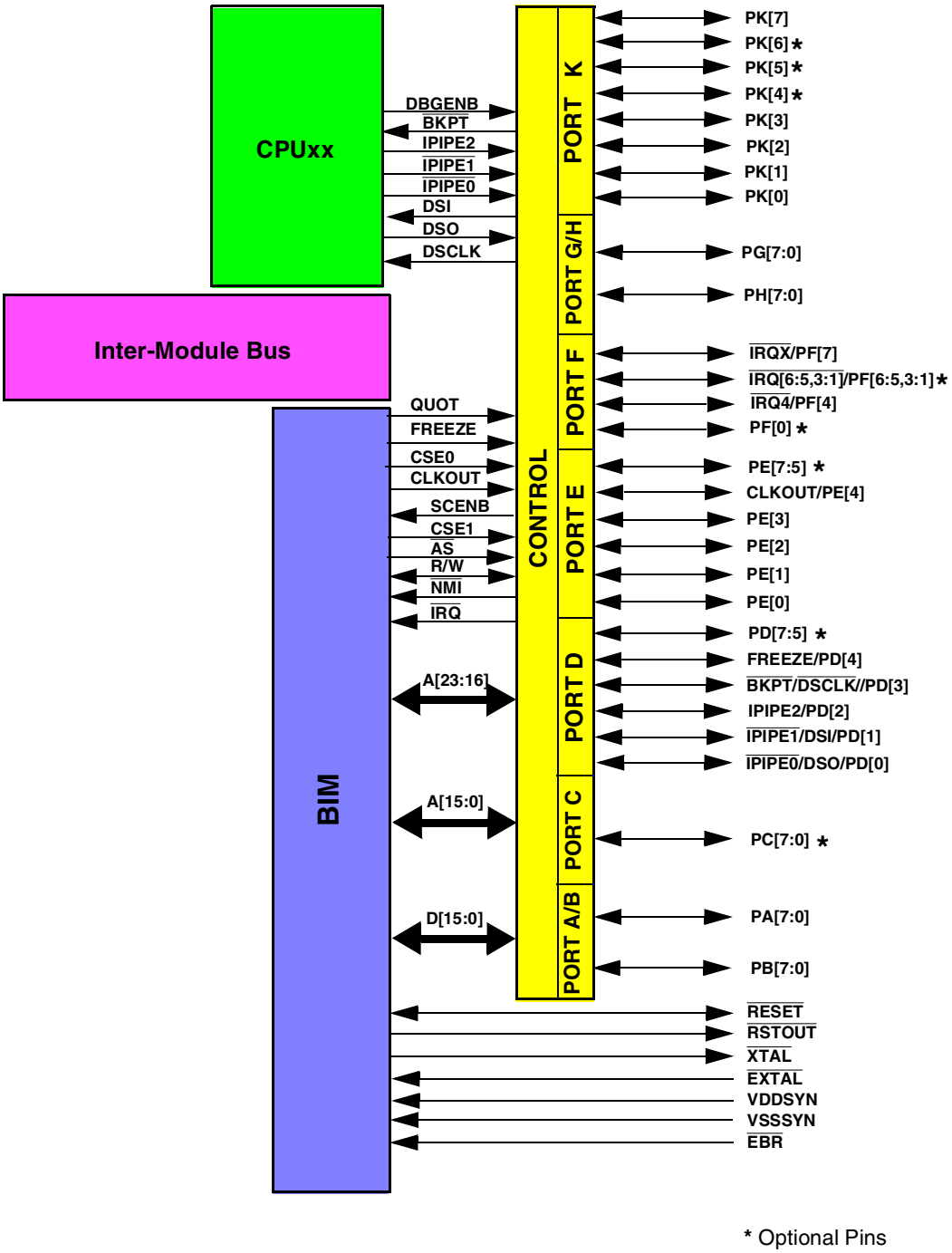
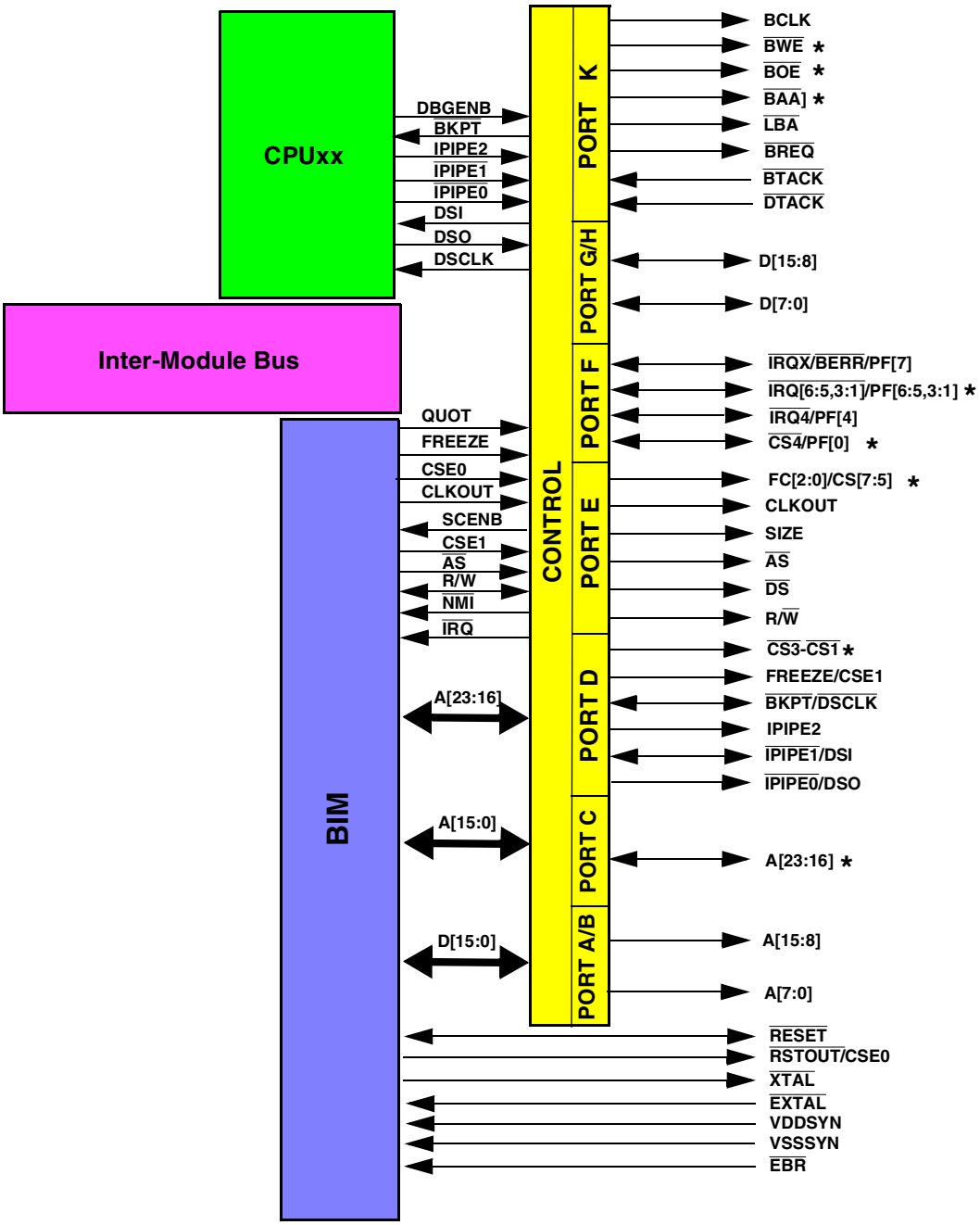


Figure 3-3 BIM Single Chip Configuration

Emulator mode is designed to improve the visibility of the overall system, in applications which utilize digital I/O pins. In emulator mode, the MCU, external memory and port replacement logic are used in combination to mimic the final system application. Since the MCU's external pins are not available to perform their digital I/O function in emulator mode, all port functions (except port F) need to be provided by the port replacement logic.

In emulator mode, primary pin functions are enabled (per PCON), except port F which can be either interrupts or digital I/O. Since the full external bus must be visible to support the external memory and port replacement logic, the emulation mode pin configuration resembles master mode. The external bus consists of a 16-bit data bus, 16-24 address lines, and optional function code lines. Available bus control signals include $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, SIZE, $\overline{BREQ}$, $\overline{BTACK}$, $\overline{BERR}$ and $\overline{DTACK}$. The burst chip select interface can be used with an external burst memory to emulate an internal burst memory. The asynchronous chip selects can be programmed to select and control external devices, and to provide bus cycle termination. Also emulation mode chip selects, CSE1 and CSE0, are provided to give additional information about the bus cycle.This configuration is shown in **Figure 3-4**.



* Optional Pins - not required for minimum BIM implementation

Figure 3-4 BIM Emulator Mode Configuration

3.1.5 RESET Configuration

The BIM has several sources of reset which include double bus fault monitor, software watchdog reset, power-on reset, loss of crystal reset, test module reset, and external reset. For more information, refer to **3.4.8 RESET Operation**.

During reset, the state of data pins D[15:12, 10] are driven onto IMB data lines ID[15:12, 10], to provide internal module configuration information. The state of data bus pins (D[11, 10:0]), $\overline{\text{SIZE}}$, $\overline{\text{LBA}}$, $\overline{\text{BTACK}}$, $\overline{\text{BOE}}$, $\overline{\text{BREQ}}$ and $\overline{\text{DTACK}}$ are used to configure various functions in the pin assignment registers. During reset these pins have [internal] weak pull-up or pull-down devices controlled by the PCON/override characteristics.

NOTE

Special care must be taken to ensure that the application uses these override pins as OUTPUTS. Futhermore, these pins should not have any devices connected to them that would pull these signals high or low during RESET unless the application truly intends to override the PCON value. If they are used in the application as inputs, during RESET the state of application signals may erronously override the PCON value causing undefinded behavior.

A[23:0], $\overline{\text{IPIPE0}}$, $\overline{\text{IPIPE1}}$, $\overline{\text{IPIPE2}}$, $\overline{\text{AS}}$, $\overline{\text{DS}}$, R/$\overline{\text{W}}$, $\overline{\text{CS1}}$ – $\overline{\text{CS7}}$, FREEZE, $\overline{\text{IRQX}}$, $\overline{\text{IRQ}}$[6:1], $\overline{\text{BAA}}$, and $\overline{\text{BWE}}$ are all tristated in reset (internal pull-ups will pull the pins high).

To be compatible with the SLIM operating in peripheral mode, A[15:0] will tristate during reset (internal pull-ups will pull the pins high).

The $\overline{\text{RSTOUT}}$ pin is driven low while the RESET pin is asserted, to indicate that the internal reset controller has reset the chip.

All pins which are configured as digital I/O during reset will be configured as inputs when the BIM comes out of reset.

NOTE

Port F unimplemented pins must be reassigned to digital I/O after reset to avoid spurious interrupts.

For more information on pin assignments at reset, refer to **3.2 BIM Ports**.

3.1.5.1 Operation Mode Selection

The BIM's operating mode is selected during reset and recorded in bits [11:9] of the BIM module configuration register (MCR). The encodings are shown in **Table 3-1**. Once reset is exited, the operating mode cannot be changed. The operating mode selection process is tailored for customer applications, but overrides are defined to allow any mode to be specified.

To simplify the customer application, bits [11:10] of the MCR are initialized to ones and a customer-specified, mask-programmable bit in the MCR is provided to allow the BIM

to be automatically configured for either master mode or single chip mode. If this default operating mode is not desired, the mode can be overridden at reset by driving D[0], to select the alternate application operating mode.

Table 3-1 Operation Mode Encoding

Operation Mode	MCR[11:9]
Master mode	1 1 1
Single chip mode	1 1 0
Slave access mode	1 0 0
Emulation mode	0 X X

Table 3-2 Operation Mode Selection

Operation Mode	BIT 11	BIT 10	BIT 9
Master mode or Single chip mode	1 (default)	1 (default)	D[0] may be driven to override mask-programmable value
Emulation mode	Set to 0 by D[10]	Don't care	Don't care

3.1.5.2 Pin Function Selection

Most BIM pins support at least two functions. Immediately after reset (internally or from external logic), the BIM pins are configured based upon the operating mode, PCON values, and whether background mode is enabled. These default pin functions are shown in **Table 3-3**.

NOTE

In some cases the default pin function is not fixed, but is selected by customer-specified, mask-programmable bits in the port configuration register (PCON). These mask-programmable defaults can be overridden at reset by driving external pins.

In most cases, the pin function can be changed after reset by writing to the appropriate pin assignment register (PAR). For specific information see **3.2 BIM Ports**.

Table 3-3 Default Pin Function Immediately after RESET

Signal	Port	RESET Pin Function in Each Mode (Background Enabled)			
		Master	Single Chip	SLAM	Emulation
A[15:0]	A,B	A[15:0]	PA[7:0], PB[7:0]	A[15:0] (input)	A[15:0]
		A[15:0]	PA[7:0], PB[7:0]	NA	A[15:0]
A[23]	C[7]	PCON[9][1]	PCON[9][1]	PCON[9][1]	PCON[9][1]
		PCON[9][1]	PCON[9][1]	NA	PCON[9][1]
A[22:16] (optional)	C[6:0]	PCON[13:11][1]	PCON[13:11][1]	PCON[13:11][1]	PCON[13:11][1]
		PCON[13:11][1]	PCON[13:11][1]	NA	PCON[13:11][1]
CS3 – CS1 (optional)	D[7:5]	PCON[5][1]	PCON[5][1]	PCON[5][1]	PCON[5][1]
		PCON[5][1]	PCON[5][1]	NA	PCON[5][1]
FREEZE / QUOT / CSE1	D[4]	PD[4]	PD[4]	FREEZE (input)	FREEZE /CSE1
		FREEZE	FREEZE	NA	FREEZE /CSE1
$\overline{\text{BKPT}}$ / DSCLK	D[3]	PD[3]	PD[3]	PD[3]	BKPT
		BKPT	BKPT	NA	BKPT / DSCLK
IPIPE2	D[2]	PD[2]	PD[2]	PD[2]	IPIPE2
		IPIPE2	IPIPE2	NA	IPIPE2
$\overline{\text{IPIPE1}}$ / DSI	D[1]	PD[1]	PD[1]	$\overline{\text{LATCH}}$ (input)	IPIPE1
		IPIPE1	IPIPE1	NA	IPIPE1
$\overline{\text{IPIPE0}}$ / DSO	D[0]	PD[0]	PD[0]	$\overline{\text{CYS}}$ (input)	IPIPE0
		$\overline{\text{IPIPE0}}$]	IPIPE0	NA	IPIPE0
FC[2:0] / $\overline{\text{CS}}$[7:5] (optional)	E[7:5]	PCON[8:7][1]	PCON[8:7][1]]	PCON[8:7][1]	PCON[8:7][1]
		PCON[8:7][1]	PCON[8:7][1]	NA	PCON[8:7][1]
CLKOUT	E[4]	PCON[3][1]	PCON[3][1]	CLKOUT	CLKOUT
		PCON[3][1]	PCON[3][1]	NA	CLKOUT
SIZE	E[3]	SIZE	PE[3]	PE[3]	SIZE
		SIZE	PE[3]	NA	SIZE
AS	E[2]	AS	PE[2]	$\overline{\text{AS}}$ (input)	AS
		AS	PE[2]	NA	AS
DS	E[1]	DS	PE[1]	$\overline{\text{DS}}$ (input)	DS
		DS	PE[1]	NA	DS
R/$\overline{\text{W}}$	E[0]	R/$\overline{\text{W}}$	PE[0]	R/$\overline{\text{W}}$ (input)	R/$\overline{\text{W}}$
		R/$\overline{\text{W}}$	PE[0]	NA	R/$\overline{\text{W}}$
IRQX / $\overline{\text{BERR}}$	F[7]	IRQX	IRQX	IIRQX (output)	IRQX
		IRQX	IRQX	NA	IRQX
IRQ[6:5,3:1] (optional)	F[6:5,3:1]	IRQ[6:5,3:1]	IRQ[6:5,3:1]	IIRQ[6:5,3:1] (output)	IRQ[6:5,3:1]
		IRQ[6:5,3:1]	IRQ[6:5,3:1]	NA	IRQ[6:5,3:1]
IRQ4	F[4]	IRQ4	IRQ4	IIRQ4 (output)	IRQ4
		IRQ4	IRQ4	NA	IRQ4

Table 3-3 Default Pin Function Immediately after RESET (Continued)

Signal	Port	RESET Pin Function in Each Mode (Background Enabled)			
		Master	Single Chip	SLAM	Emulation
CS4 (optional)	F[0]	PCON[6][1]	PCON[6][1]	PCON[6][1]	PCON[6][1]
		PCON[6][1]	PCON[6][1]	NA	PCON[6][1]
D[15:0]	G/H	D[15:0]	PG[7:0], PH[7:0]	D[15:0]	D[15:0]
		D[15:0]	PG[7:0], PH[7:0]	NA	D[15:0]
BCLK	K[7]	PCON[2][1]	PCON[2][1]	PCON[2][1]	PCON[2][1]
		PCON[2][1]	PCON[2][1]	NA	PCON[2][1]
BWE (optional)	K[6]	PCON[2][1]	PCON[2][1]	PCON[2][1]	PCON[2][1]
		PCON[2][1]	PCON[2][1]	NA	PCON[2][1]
BOE (optional)	K[5]	PCON[2][1]	PCON[2][1]	PCON[2][1]	PCON[2][1]
		PCON[2][1]	PCON[2][1]	NA	PCON[2][1]
BAA (optional)	K[4]	PCON[2][1]	PCON[2][1]	PCON[2][1]	PCON[2][1]
		PCON[2][1]	PCON[2][1]	NA	PCON[2][1]
LBA	K[3]	PCON[2][1]	PCON[2][1]	PCON[2][1]	PCON[2][1]
		PCON[2][1]	PCON[2][1]	NA	PCON[2][1]
BREQ	K[2]	PCON[1][1]	PCON[1][1]	$\overline{\text{BREQ}}$ (input)	BREQ
		PCON[1][1]	PCON[1][1]	NA	BREQ
BTACK	K[1]	PCON[1][1]	PCON[1][1]	$\overline{\text{BTACK}}$ (output)	BTACK
		PCON[1][1]	PCON[1][1]	NA	BTACK
DTACK	K[0]	PCON[14][1]	PCON[14][1]	$\overline{\text{DTACK}}$ (output)	DTACK
		PCON[14][1]	PCON[14][1]	NA	DTACK
$\overline{\text{RSTOUT}}$ /CSE0	CSE0	RSTOUT	RSTOUT	RSTOUT	$\overline{\text{RSTOUT}}$ /CSE0
		RSTOUT	RSTOUT	NA	$\overline{\text{RSTOUT}}$ /CSE0

NOTES:
1. PCON value can be overridden by driving external pin(s) at reset NA: Not Allowed mode combinations.

3.1.5.3 PLL Mode Selection

The PLL mode is determined at reset. The state of V_{DDSYN} is sampled on the rising edge of RESET to determine whether the PLL provides the internal BIM clock or whether the clock is provided from an external clock source. The state of the $\overline{\text{LBA}}$ pin determines whether the on-chip PLL operates in normal PLL mode or whether it operates in 1:1 mode. Refer to **3.4 Clocks** for more details.

3.1.5.4 Background Debug Mode Selection

When $\overline{\text{BKPT}}$ is asserted during reset, background debug mode is enabled. Background mode can be entered if the $\overline{\text{BKPT}}$ pin is asserted during a bus cycle, providing that the background mode is enabled, the BGND instruction is executed, a double bus fault occurs, or an internal peripheral asserts the $\overline{\text{IBKPT}}$ signal.

When background debug mode is entered, certain pin functions are required. These pin functions are noted in the colored areas of **Table 3-3**.

In background mode, FREEZE is asserted and the BIM response is determined by the FRZ bits in the MCR. Based on these encodings, the software watchdog, the real-time clock, and the prescaler may stop counting. In addition, all BIM registers are accessible by the CPU. If a register contains any write-once bits, these bits are readable and writable during FREEZE. When FREEZE is negated, the BIM resumes operation based on the current values contained in its registers. The mask-programmable options are summarized in **Table 3-4**.

Table 3-4 Customer-Specified Mask-Programmable Options

Option	Register
BCS — BAA function	PCON[15]
Port K[0]/ $\overline{\text{DTACK}}$ function	PCON[14]
Configuration of port C / A[22:16] function	PCON[13:11]
Boot device size	PCON[10]
A[23] / port C[7] function	PCON[9]
FC[2:0] / CS7 — CS5 / port E[7:5] function	PCON[8:7]
CS4 / port F[0] function	PCON[6]
CS3-CS1 / port D[7:5] function	PCON[5]
Normal PLL mode or 1:1 mode	PCON[4]
Port E[4] / CLKOUT pin function	PCON[3]
Burst chip select / port K[7:3] function	PCON[2]
Port K[2:1]/ $\overline{\text{BREQ}}$, $\overline{\text{BTACK}}$ function	PCON[1]
Master mode / single chip mode	PCON[0]

3.1.5.5 Default Configuration Override During RESET

While the BIM is held in reset, data bus pins D[11:0] are configured as inputs with weak pull-ups or pull-downs to reflect the state of the MCR bits or PCON bits corresponding to each D[i] input. D[15:12] have internal pull-ups as in previous integration modules. Default reset values for the PCON register can be overridden by driving the PCON override pins to predefined logic levels during reset as shown in **Table 3-5**. Please refer to **3.1.12 Port Configuration Shadow Register (PCON)** for complete description of all the default reset options.

Table 3-5 Default Configuration During RESET

Pin(s) Affected	RESET Shadow Bit	Override Pin	Effect of Override Pin High During RESET	Effect of Override Pin Low During RESET
All ports	VDD, VDD, PCON[0]	D[10,1,0]	State of override pins 1 1 1 1 1 0 1 0 1 1 0 0 0 x x	Mode selected Master mode Single chip mode Master test mode (MISR enabled) Slave access mode Emulator mode
BCS	PCON[15]	BOE	$\overline{BAA}$ = BAA waveform	$\overline{BAA}$ = HP$\overline{CE}$
A[22:16] / port C[6:0]	PCON[13:11]	D[11,4,2]	State of Override Pins 0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1	Configuration of Pins Port C[6:0] Port C[6:1], A[16] Port C[6:2], A[17:16] Port C[6:3], A[18:16] Port C[6:4], A[19:16] Port C[6:5], A[20:16] Port C[6], A[21:16] A[22:16]
Boot device size	PCON[10]	SIZE	8-bit boot device	16-bit boot device
A[23] / port C[7]	PCON[9]	D[9]	A[23]	Port C[7]
FC[2:0] / CS7 – CS5 / port E[7:5]	PCON[8:7]	D[8:7]	State of override pins 0 0 1 0 1 1	Configuration of pins port E[7:5] FC[2:0] CS7 - CS5
CS4 / port F[0]	PCON[6]	D[6]	CS4	Port F[0]
CS3-CS1/ port D[7:5]	PCON[5]	D[5]	CS3 – CS1	Port D[7:5]
Internal / external clock source	NA	V_{DDSYN}	PLL	External clock
PLL Mode (If V_{DDSYN} = 1)	PCON[4]	LBA	Normal PLL mode	1:1 PLL mode
CLKOUT/Port E[4]	PCON[3]	D[3]	CLKOUT	Port E[4]
Burst chip select/port K[7:3]	PCON[2]	BREQ	BCLK, BWE, BOE, BAA, LBA	Port K[7:3]
$\overline{BTACK}$, $\overline{BREQ}$/ port K[2:1]	PCON[1]	BTACK	$\overline{BTACK}$, $\overline{BREQ}$	Port K[2:1]
$\overline{DTACK}$/ port K[0]	PCON[14]	DTACK	DTACK	Port K[0]
MCR[9]	PCON[0]	D[0]	Master mode	Single chip mode
BKPT	NA	BKPT	Background mode disabled	Background mode enabled

3.1.6 EBR Operation

The external bus may be granted to an alternate bus master in master, emulation, or background debug operation mode. The external master cannot access internal MCU resources.

3.1.7 Interrupt Operation

When there is a pending interrupt at an instruction boundary, and no higher priority exceptions are pending, the CPU initiates interrupt processing for that interrupt priority level (IPL). The CPU performs an internal interrupt acknowledge cycle (IACK) to fetch the vector number corresponding to the interrupt service routine. All modules with pending interrupt requests at the interrupt priority level arbitrate for service. If no module arbitrates for the IACK and the IACK cycle is terminated with BERR, a spurious interrupt exception will be taken.

BIM interrupt sources are:

- Real-time counter/software watchdog counter (RTC/SWC)
- Port F edge-detect interrupt logic, and
- the IRQ7-1 pins

Since the IRQ pins may assert an interrupt at the same priority level as the RTC/SWC and/or the port F edge-detect logic, a priority scheme gives the highest priority to the RTC/SWC, followed by the interrupt pins, and then the port F edge-detect logic. An interrupt from a lower priority source remains pending until the next allowable interrupt time.

The RTC/SWC and the port F edge-detect logic have programmable interrupt vector registers that provide the interrupt vector during an IACK cycle.

The BIM supports both autovector and external vector sources for interrupt requests indicated by the external IRQ pins. The BIM can terminate external IACK cycles by asserting IAVEC, indicating that the autovector corresponding to the interrupt priority level (IPL) should be used. Alternately, the IACK cycle can be terminated by providing an external interrupt vector, and asserting $\overline{\text{DTACK}}$. $\overline{\text{DTACK}}$ can be generated by an external device or by a chip select.

An autovector response is selected for a specific IRQ pin by setting the corresponding AV bit in the IACK response register. When an IACK cycle is executed for the IPL corresponding to the level of the IRQ pin, the BIM asserts IAVEC to cause the CPU to terminate the IACK cycle.

The BIM executes external IACK cycles when the AV bit is negated for the respective IPL. Chip selects can be programmed to assert during IACK cycles and to provide cycle termination. See **3.6.4 Asynchronous Chip Select Operations**.

3.1.8 LPSTOP Operation

To facilitate low power standby operation, the BIM is designed to operate with CPUs which support a low power STOP instruction (LPSTOP). When the CPU executes the LPSTOP instruction, the BIM decodes a CPU space type 3 cycle. A copy of the mask

priority level is written to the BIM, and an external indication of LPSTOP execution is available through chip select option programming. The BIM terminates the LPSTOP cycle by asserting internal DTACK. Once LPSTOP mode is entered, the only way to restart the system is by reset or by any interrupt request that is higher than the interrupt mask priority level. The interrupt source can be the port F I/O pins, IRQ pins (level or edge), or either modulus counter in the system protection sub-module. See **3.4.5 LPSTOP Operation** for information on programming which system clocks are disabled in LPSTOP and other options.

3.1.9 Emulator Support

The BIM contains support for emulator logic which can be used to faithfully replicate on-chip digital I/O ports externally. In emulation mode, accesses to internal port registers become external accesses to the port replacement logic. Access to port replacement logic is transparent to the application software, including access time. All other BIM register accesses are unaffected in emulation mode.

Emulation mode is entered by pulling D[10] low during reset. To ensure proper operation of the emulator logic, the LOAD control bit in the MCR is overridden, forcing the pad drivers to full load.

While in emulation mode, special emulator chip selects (CSE1, CSE0) are driven externally to allow internal accesses to be tracked by external hardware. See **Table 3-6**. CSE1 is time-multiplexed with FREEZE and CSE0 is time-multiplexed with the $\overline{\text{RSTOUT}}$ pin. The special emulation chip selects are asserted with the same timing as address strobe ($\overline{\text{AS}}$). Emulation mode chip selects indicate whether the bus cycle is an external access, external port register access, or internal module control block access.

3.1.9.1 External Port Replacement Indication

In emulation mode, all port A, B, C, D, E, G, H, and K registers are mapped externally **(IAACK is not driven by the BIU, but IDTACK is driven in two clocks to improve external timing)**. The states of MODMAP and IFC[2:0] = Z01 are included in the address decode. CSE1 and CSE0 = 10 whenever any emulated port registers are addressed without privilege violation (0xY FFA10 — 0xYF FA2F). **The BIM does not respond to these accesses, other than to terminate the cycle in two clocks, so that external emulator logic can respond**. The CSE pins indicate external port replacement access on byte accesses to individual registers, as well as word accesses to data and data direction register pairs.

When the BIM is in emulation mode, port accesses to supervisor only registers (FC = 101) are checked by the BIM only to Z01 functionality. Emulation logic should further qualify emulated registers for supervisor only access.

3.1.9.2 Internal Module Control Block Indication

Accesses to the 32-Kbyte address block which contains the module control blocks (0xYF 8000 — 0xYF FFFF), exclusive of those registers listed in **3.1.9.1 External**

Port Replacement Indication, are indicated by CSE1 and CSE0 = 00. The lower 15 bits of the address bus indicates the register accessed within the 32-Kbyte block.

3.1.9.3 Other Access Indication

In previous integration modules, the external DTACK pin could not be driven on a cycle by cycle basis to terminate two clock external cycles. Due to this limitation, ICSM technology was developed to cope with terminating two clock external emulated cycles. The BIM allows DTACK, BTACK, and BERR to terminate two clock external cycles. The BIM does not support ICSM technology.

To emulate internal modules in BIM based systems, the emulated module is simply disabled allowing its cycles to go external to access emulation hardware.

If an access is not covered by the encodings specified in the previous sections, CSE1 and CSE0 will equal 11.

Table 3-6 Emulator Mode Chip Select Summary

CSE1	CSE0	Indication in Emulator Mode
1	1	All access not covered by CSE encodings = {10, 00}
1	0	Access to emulated port register in emulator mode
0	1	Reserved
0	0	Access to 32-Kbyte module control block, not an emulated port register.

3.1.10 Module Mapping

All of the MCU internal module registers occupy a 32-Kbyte block of memory. This 32-Kbyte register block is movable to one of two locations in the map, starting at 0x7F 8000 or 0xFF 8000, by programming the module mapping (MM) bit in the BIM module configuration register. This allows the user the choice of placing the module registers in the upper short addressing range (0xFF 8000 — 0xFF FFFF), or reserving the entire short addressing range for external devices.

The BIM memory map occupies 128 bytes. **Figure 3-5** shows the decoding of the address lines for the BIM internal registers as follows: A[23] is compared to the MM line on the IMB, A[22:15] must be high, A[14:7] must match the assigned value for the module decode as specified in the device control and A[6:0] must match the individual register decodes as detailed in **Figure 3-5**.

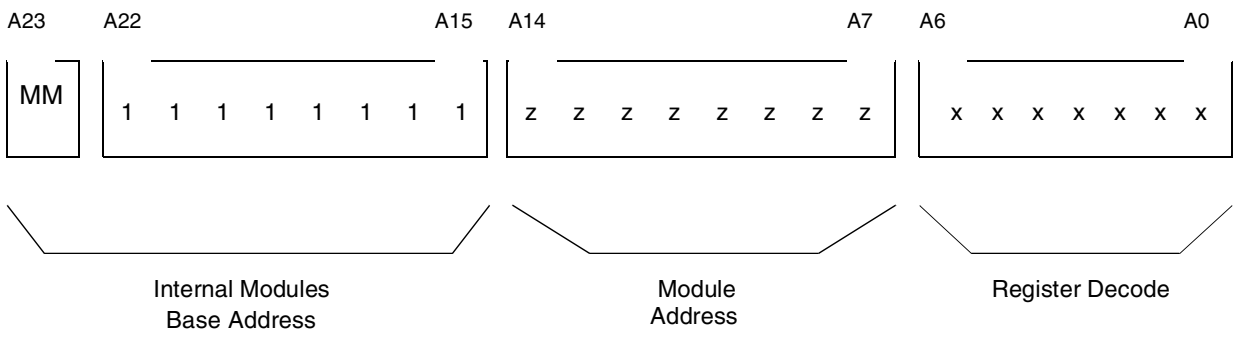


Figure 3-5 BIM Module Address Decoding

NOTE

Even though some CPUs have an internal address bus size other than 24 bits, the BIM is implemented to operate with a 24-bit address bus. The 24-bit (16-Mbyte) module memory map is replicated 256 times when viewed from a CPU with a 32-bit program counter.

3.1.10.1 BIM Memory Map

The BIM memory map occupies 128 bytes of address space. The address space is designated as supervisor-only data space or as supervisor/unrestricted data space. Some CPUs, such as the CPU32 family, support a supervisor state which allows access to all registers and a user state which restricts access to some registers. Function code line two (IFC2) on the IMB is used to determine whether an access is in supervisor or unrestricted mode. Most of the registers in the BIM must be accessed while in supervisor data space. Some registers, however, (refer to Z in the function code column in **Figure 3-5**) may be programmed to allow access in either supervisor or unrestricted mode. In MCU implementations which do not support separate supervisor and unrestricted spaces, IFC2 should be forced to always be asserted.

When the BIM is in emulation mode, normal port accesses to supervisor only registers (FC = 101) are checked only as Z01.

Table 3-7 shows the organization of all registers in the BIM. For more information about individual registers refer to the sub-module shown in the right column. Shaded areas of the memory map are reserved. Writes to reserved locations are ignored and reads return zeros.

Table 3-7 BIM Memory Map

Address	FC	15 8	7 0	Sub Module
0xYF FA00	101	MODULE CONFIGURATION REGISTER (MCR)		BIU
0xYF FA02	101	MODULE TEST REGISTER (MTR)		—
0xYF FA04	101	BIM TEST REGISTER (BIMTR)	MODULE DISABLE REGISTER (MDR)	—
0xYF FA06	101			
0xYF FA08	101	CLOCK SYTHESIZER CONTROL (SYNCR)		CLOCK
0xYF FA0A		CLOCK SYTHESIZER STATUS (SYNST)	RESET STATUS (RSR)	CLOCK
0xYF FA0C				
0xYF FA0E	Z01	PORT/CLOCK CONFIGURATION SHADOW REGISTER (PCON)		
0xYF FA10	Z01	PORT A OUTPUT DATA REGISTER (PORTA)	PORT B OUTPUT DATA REGISTER (PORTB)	PORT
0xYF FA12	Z01	PORT A PIN DATA REGISTER (POR-TAP)	PORT B PIN DATA REGISTER (PORTBP)	—
0xYF FA14	Z01		PORT A/B DATA DIRECTION (DDRAB)	—
0xYF FA16				—
0xYF FA18	Z01	PORT C OUTPUT DATA (PORTC)	PORT D OUTPUT DATA (PORTD)	—
0xYF FA1A	Z01	PORT C PIN DATA REGISTER (PORTCP)	PORT D PIN DATA REGISTER (PORTDP)	—
0xYF FA1C	Z01	PORT C DATA DIRECTION (DDRC)	PORT D DATA DIRECTION (DDRD)	—
0xYF FA1E	101	PORT C PIN ASSIGNMENT (PCPAR)	PORT D PIN ASSIGNMENT (PDPAR)	—
0xYF FA20	Z01	PORT K OUTPUT DATA (PORTK)	PORT E OUTPUT DATA REGISTER (PORTE)	—
0xYF FA22	Z01	PORT K PIN DATA REGISTER (PORTKP)	PORT E PIN DATA REGISTER (PORTEP)	—
0xYF FA24	Z01	PORT K DATA DIRECTION (DDRK)	PORT E DATA DIRECTION (DDRE)	—
0xYF FA26	101	PORT K PIN ASSIGNMENT (PKPAR)	PORT E PIN ASSIGNMENT (PEPAR)	—
0xYF FA28	Z01	PORT G OUTPUT DATA REGISTER (PORTG)	PORT H OUTPUT DATA REGISTER (PORTH)	—
0xYF FA2A	Z01	PORT G PIN DATA REGISTER (PORTGP)	PORT H PIN DATA REGISTER (PORTHP)	—
0xYF FA2C	Z01	PORT G DATA DIRECTION (DDRG)	PORT H DATA DIRECTION (DDRH)	—
0xYF FA2E				—
0xYF FA30	Z01		PORT F OUTPUT DATA (PORTF)	—
0xYF FA32	Z01		PORT F PIN DATA REGISTER (PORTFP)	—
0xYF FA34	Z01		PORT F DATA DIRECTION (DDRF)	—
0xYF FA36	101	PORT F PIN ASSIGNMENT (PFPAR)		—
0xYF FA38	101		PORT F EDGE FLAGS (PORTFE)	—
0xYF FA3A	101	PORT F IACK RESPONSE (PFIACK)	PORT F EDGE-DETECT ENABLE (PFEER)	—
0xYF FA3C	101	PORT F LEVEL (PFLVR)	PORT F INTERRUPT VECTOR (PFIVR)	—
0xYF FA3E				

Table 3-7 BIM Memory Map (Continued)

Address	FC	15 8	7 0	Sub Module
0xYF FA40	101	TEST MODULE MASTER SHIFT A (MSRA)		TEST
0xYF FA42	101	TEST MODULE MASTER SHIFT B (MSRB)		—
0xYF FA44	101	TEST MODULE SHIFT COUNT A (SCRA)	TEST MODULE SHIFT COUNT B (SCRB)	—
0xYF FA46	101	TEST MODULE REPETITION COUNTER (REPS)		—
0xYF FA48	101	TEST MODULE CONTROL REGISTER (CREG)		—
0xYF FA4A	101	TEST MODULE DISTRIBUTED REGISTER (DREG)		—
0xYF FA4C		BURST CHIP SELECT OPTION REGISTER 2 (BCSOR2)		BCS
0xYF FA4E				
0xYF FA50	101	SYSTEM PROTECT CONTROL (SYPCR)		SYSPROT
0xYF FA52	101	TIMER CONTROL (TIC)	TIMER INTERRUPT VECTOR (TIV)	—
0xYF FA54	101		SOFTWARE SERVICE (SWS)	—
0xYF FA56	101	PRESCALER (READ-ONLY) (PRE)		SYSPROT
0xYF FA58	101	SOFTWARE WATCHDOG INTERVAL REGISTER (SWI)		—
0xYF FA5A	101	REAL-TIME PERIOD REGISTER (RTP)		—
0xYF FA5C	101	SOFTWARE WATCHDOG INTERVAL TIMER OPERATION REGISTER (SWIT)		—
0xYF FA5E	101	REAL-TIME DOWNCOUNTER (RTDC) (READ-ONLY)		—
0xYF FA60	101	CHIP SELECT BASE ADDRESS REGISTER 1 (CSBAR1)		CHIP SEL
0xYF FA62	101	CHIP SELECT OPTION REGISTER 1 (CSOR1)		—
0xYF FA64	101	CHIP SELECT BASE ADDRESS REGISTER 2 (CSBAR2)		—
0xYF FA66	101	CHIP SELECT OPTION REGISTER 2 (CSOR2)		—
0xYF FA68	101	CHIP SELECT BASE ADDRESS REGISTER 3 (CSBAR3)		—
0xYF FA6A	101	CHIP SELECT OPTION REGISTER 3 (CSOR3)		—
0xYF FA6C	101	CHIP SELECT BASE ADDRESS REGISTER 4 (CSBAR4)		—
0xYF FA6E	101	CHIP SELECT OPTION REGISTER 4 (CSOR4)		—
0xYF FA70	101	CHIP SELECT BASE ADDRESS REGISTER 5 (CSBAR5)		—
0xYF FA72	101	CHIP SELECT OPTION REGISTER 5 (CSOR5)		—
0xYF FA74	101	CHIP SELECT BASE ADDRESS REGISTER 6 (CSBAR6)		—
0xYF FA76	101	CHIP SELECT OPTION REGISTER 6 (CSOR6)		—
0xYF FA78	101	CHIP SELECT BASE ADDRESS REGISTER 7 (CSBAR7)		—
0xYF FA7A	101	CHIP SELECT OPTION REGISTER 7 (CSOR7)		—
0xYF FA7C	101	BURST CHIP SELECT BASE (BCSBAR)		—
0xYF FA7E	101	BURST CHIP SELECT OPTION REGISTER (BCSOR)		—

3.1.10.2 Module Configuration Register (MCR)

This register controls the BIM configuration. The MCR is reset by SRESET.

MCR — Module Configuration Register **0xYF FA00**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
LOAD	FRZ		SUPVA	MODE			SHEN	SPEED	MM	RESERVED		IARB			
RESET:															
—[1]	1	1	1	1	1	1	—[2]	0	1	0	0	1	1	1	1

NOTES:
1. 0 = Master and singlechip modes, 1 = All other modes
2. 0 = All other modes, 1 = If in emulation mode

Table 3-8 MCR Bit Descriptions

Bit(s)	Name	Description
15	LOAD	Pad driver LOAD. This bit controls the drive strength of all pad output drivers by selecting whether they will be at full or partial drive strength. If IFREEZE is asserted, the bit is write always in all modes. See **Table 3-9** for LOAD bit write accessibility. 0 = Drivers will be at partial drive strength, resulting in a reduction in power consumption 1 = Drivers will be at full strength, allowing the pad to drive the maximum capacitive load
14:13	FRZ	Freeze bit. These bits control how the system protection submodule responds to the assertion of the IFREEZE line on the IMB. See for the FREEZE fields. 00 = Counters ignore the assertion of the IFREEZE line 01 = SWDOG, RTC, prescaler disabled 10 = Bus monitor disabled 11 = Bus monitor, SWDOG, RTC, prescaler disabled
12	SUPVA	Supervisor data space. Used to enable/disable access protection to some of the BIM registers. Most of the registers in the BIM can only be accessed while in supervisor mode. Some register accesses are controlled by the SUPVA bit to allow either supervisor-only access or unrestricted access. Privilege violations to supervisor restricted registers will result in the generation of AACK and DTACK. A read access will return a data value equal zero, write data will be ignored. 0 = Access to registers is unrestricted 1 = Registers are restricted to supervisor-only access
11:9	MODE	Mode configuration. These bits determine the BIM operating mode. 0XX = Double-buffered mode 100 = Slave access mode 110 = Single chip mode 111 = Master mode
8	SHEN	Show cycle enable. Used by the EBI to determine whether to drive the external bus during internal transfer (show cycle) operations. Show cycles only affect the data bus and the control signals $\overline{DS}$. Even if show cycles are disabled, the address information is always driven by the BIM unless mastership of the external bus has been relinquished ($\overline{EBR}$ asserted). In emulator mode, the SHEN bit is forced to a 1. 0 = Show cycles disabled 1 = Show cycles enabled
7	SPEED	I/O pad speed. Controls the rise time of MCU pins configured as a digital outputs. When SPEED is cleared digital output pins, that support a slow rise time to minimize EMI, transistion slowly. When SPEED is set digital outputs are configured to transition in their “fast mode”. The actual rise times (slow/fast) is dependent on the pad type.
6	MM	Module map. etermines the position of the MMF internal modules within the overall device memory map. This bit is driven onto the internal bus and A[23] must match this value for an internal module to be selected by a bus cycle. 0 = Internal modules are located at 0x7FF000 – 0x7FFFFF 1 = Internal modules are located at 0xFFF000 – 0xFFFFFF

Table 3-8 MCR Bit Descriptions (Continued)

Bit(s)	Name	Description
5:4	—	Reserved
3:0	IARB	Interrupt arbitration. Used to assign an interrupt arbitration priority level for the BIM. The BIM arbitrates on the IMB for interrupt servicing based on its IARB value. When interrupts are enabled, the IARB field must be set to a non-zero value. The lowest priority IARB number is 0001 and the highest priority IARB number is 1111. The interrupt sources in the BIM are: port F interrupt logic and the SWDOG or RTC counters in the system protection submodule.

Table 3-9 LOAD Bit Write Accessibility:

Master	Single Chip	Emulation	MFTM, ETM = 0	MFTM, ETM = 1	SLAM, ETM = 0	SLAM, ETM = 1
Write once	Write once	Read only	Read only	Read/Write	Write once	Read/Write

3.1.11 Module Disable Register (MDR)

MDR — Module Disable Register **0xYF FA04**

MSB 7	6	5	4	3	2	1	LSB 0
STP7 (STPCPU)	STP6	STP5	STP4	STP3	STOPCS	STOPTEST	STOPSYS-PROT
RESET:							
0	0	0	0	0	0	0	0

The module disable register (MDR) is used to disable the clocks to IMB modules during LPSTOP. The MDR also contains bits to disable the clocks in the chip select, test sub-module, and system protection sub-modules. The MDR is reset by SRESET.

Table 3-10 MDR Bit Descriptions

Bit(s)	Name	Description
7:5	STP[7:5]	Stop IMB module clock. By convention STP7 is assigned to control the CPU clock and is historically aliased as STPCPU. The STP6 and STP5 bits are used for derivative-specific clock control. During LPSTOP, each STPx bit controls the assertion of a corresponding signal on the IMB. $\overline{\text{CLKDIS}}$ [4:2] are controlled by STP7 (STPCPU), STP6 and STP5, respectively. If a module is connected to an asserted CLKDIS signal, the module will disable its CLOCK signal to achieve the lowest power consumption. To support the use of the STP bits in the MDR to selectively disable the SYSCLKs to each IMB module, the STICLK bit in the SYNCR is set to zero to allow the EXTCLK, BIMCLKs and SYSCLKs to continue to function during LPSTOP. MDR[7:5] are cleared by SRESET or by exiting LPSTOP. 0 = If LPSTOP is executed, the corresponding $\overline{\text{CLKDIS[4:2]}}$ signal remains negated 1 = If LPSTOP is executed, the BIM asserts the $\overline{\text{CLKDIS[4:2]}}$ signal to disable the module
4:3	STP[4:3]	Stop IMB module clock. During MCU operation, STP4 and STP3 controls the assertion of a corresponding signal on the IMB to statically control an IMB module's clocks. $\overline{\text{CLKDIS[1:0]}}$ are controlled by STP4, and STP3, respectively. If a module is connected to an asserted CLKDIS signal, the module will gate off its CLOCK signal to achieve the lowest power consumption. 0 = The corresponding $\overline{\text{CLKDIS[1:0]}}$ signal remains negated 1 = The BIM asserts the $\overline{\text{CLKDIS[1:0]}}$ signal to disable the module

Table 3-10 MDR Bit Descriptions (Continued)

Bit(s)	Name	Description
2	STOPCS	Stop chip selects. Used to enable/disable access protection to some of the BIM registers. Most of the registers in the BIM can only be accessed while in supervisor mode. Some register accesses are controlled by the SUPVA bit to allow either supervisor-only access or unrestricted access. Privilege violations to supervisor restricted registers will result in the generation of AACK and DTACK. A read access will return a data value equal zero, write data will be ignored. 0 = Chip select logic is enabled 1 = Chip select logic is not enabled
1	STOP-TEST	Stop test. This control bit directs BIM logic to disable clocks to the system protection submodule. While the STOPSYSPROT bit is set, the system protect submodule does not function. 0 = The test submodule logic is enabled 1 = The test submodule logic is not enabled
0	STOP-SYS-PROT	Stop system protection. This control bit directs BIM logic to disable clocks to the system protection submodule. 0 = Not possible to read or write the system protect register 1 = Possible to read or write the system protect register

3.1.12 Port Configuration Shadow Register (PCON)

PCON — BIM Port Configuration Shadow Register **0xYF FA0E**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
BAA	DTACK	A[22:16]			BOOT SIZE	A[23]	$\overline{FC}$[2:0]		CS4	CS3-CS1	PLL REF	CLK-OUT	BURST CS	BTACK BREQ	MODE
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The port configuration shadow register is a read-only, mask programmable register that determines the default pin functions for certain operating modes and the default clock configuration at reset. The reset values of the corresponding bits and fields in the Port C, D, E, F, and K pin assignment registers are affected, see **Table 3-11**. Not all bits in the port pin assignment registers have corresponding shadow bits in PCON.

NOTE

PCON programming must be consistent with pin availability. If a pin does not exist on the device, PCON bits should be programmed to configure the pin function as digital I/O out of reset, since EBI operation may be affected by the presence or absence of these pins.

For single chip applications the PCON value must specify digital I/O options for all port pins. PCON[15] is a don't care if PCON[2] = 0 (Port K[7:3] is digital I/O].

Table 3-11 Port Configuration Shadow Register (PCON)

Bit(s)	Register	Pin(s) Affected	State	Pin Function
PCON[15]	BCSOR2	BAA	1 0	BAA HPCE
PCON[14]	PKPAR[0]	DTACK	0 1	PK[0] DTACK
PCON[13:11]	PCPAR[7:4]	A[22:16]/ PC[6:0]	0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1	Port C[6:0] Port C[6:1], A[16] Port C[6:2], A[17:16] Port C[6:3], A[18:16] Port C[6:4], A[19:16] Port C[6:5], A[20:16] Port C[6], A[21:16] A[22:16]
PCON[10]	BCSOR1	—	0 1	16-bit boot device 8-bit boot device
PCON[9]	PCPAR[7]	A[23] / port C[7]	1 0	A[23] Port C[7]
PCON[8:7]	PEPAR[7:6]	FC[2:0] / $\overline{CS7}$ – $\overline{CS5}$ / port E[7:5]	1 0 1 1 0 0	FC[2:0] $\overline{CS7}$ - $\overline{CS5}$ Port E[7:5]
PCON[6]	PFPAR[1:0]	$\overline{CS4}$ / port F[0]	1 0	CS4 Port F[0]
PCON[5]	PDPAR[7:5]	$\overline{CS3}$-$\overline{CS1}$ / port D[7:5]	1 0	CS3-CS1 Port D[7:5]
PCON[4]	SYNST[15]	—	1 0	Normal PLL mode 1:1 PLL mode
PCON[3]	PEPAR[4]	CLKOUT / port E[4]	1 0	CLKOUT Port E[4]
PCON[2]	PKPAR[3]	Burst chip select / port K[7:3]	1 0	Burst chip select Port K[7:3]
PCON[1]	PKPAR[2:1]	$\overline{BTACK}$, $\overline{BREQ}$/ port K[2:1]]	1 0	$\overline{BTACK}$, $\overline{BREQ}$ Port K[2:1]
PCON[0]	MCR[9]	None	1 0	Master mode Single chip mode

3.2 BIM Ports

Many of the pins associated with the BIM may be used for several different functions. Their primary function is to provide an external bus interface for production testing, emulation, and for applications which require off-chip resources to be accessed. When not used for their primary functions, the pins may be used as digital I/O pins. In some cases the pin function is set by the operation mode and the alternate pin functions are not supported.

To facilitate the I/O function, the BIM pins are grouped into 8-bit ports. Each port has associated registers which configure the pins for the desired function, monitor the pins, and control the pins within the port.

When a pin is configured for its primary function, the BIM will enable the 3-V output driver in the pads. When a pin is configured for digital I/O, the BIM will enable the 5-V output driver in the pads.

3.2.1 Port Operation

Pins are configured as digital I/O during reset configuration or by writing to the port's pin assignment register. Ports A, B, G and H do not have pin assignment registers and are configured as digital I/O only if the BIM is placed in single chip mode during reset. Ports C, D, E, F and K have pin assignment registers which allow the user to select between digital I/O or another pin function after reset. The pin assignments at reset are shown in **Table 3-20**.

All digital I/O pins are programmable as either input or output pins via an associated data direction register. In most cases the pins are individually defined as input or output. However, ports A and B are selected as input or output on a whole-port basis.

All ports have both an output data register and a pin data register to monitor and/or control the state of its pins. Writes to the output data register cause IMB data to be stored, and then driven to the corresponding pads of pins programmed as outputs. A read of the output data register returns the current state of this data register, regardless of the actual state of the pins. A read access of a pin data register returns the current state of its corresponding pins, regardless of whether they are input or output. Writes to the pin data registers have no effect.

In addition to the output data and pin data registers, port F has additional registers to support edge or level sensitivity, and interrupt capability. The port F edge-detect flag register indicates that a transition has occurred on an input or output pin. When the corresponding bit in the port F edge-detect I/O interrupt enable register is set, an interrupt is generated at the level specified in the port F interrupt level register. The interrupt vector is defined in the port F edge-detect vector register.

3.2.2 Port Register Access

Internal accesses to all Port A, B, C, D, E, F, G, H and K registers are always two clock accesses. In emulation mode, accesses to these port registers are ignored (IAACK not asserted) allowing the port access to go external so that emulation hardware can satisfy the port access request.

All port registers are byte-addressable and are grouped to allow coherent word access to port data register pairs. Writes to reserved bits in the port registers have no effect and reads return zeros.

3.2.3 Optional Pins

Some port F pins may not be implemented on a particular chip. When the missing pin is programmed to function as an input, the internal input feedback path is tied to V_{SS} and returns a data value of zero if read in the pin data register. Since port F pins are assigned to the IRQ input function after reset in all operation modes except SLAM, the

user must reassign the unimplemented pins to be digital io to avoid spurious interrupts at levels corresponding to the unimplemented pins.

As shown in **Figure 3-6**, special “loopback” logic is implemented to allow better verification of digital I/O and IRQ functionality when optional pins are not present. When a pin is not present but is programmed as an output (primary function outputs and digital outputs), the internal input feedback path normally driven by the pin is driven to the same logic value as the internal output. This loopback logic is controlled by an internal rom plug which indicates whether an optional pin is present in this product derivative. Loopback logic is implemented for the optional pins of ports C, D, E, F, and K.

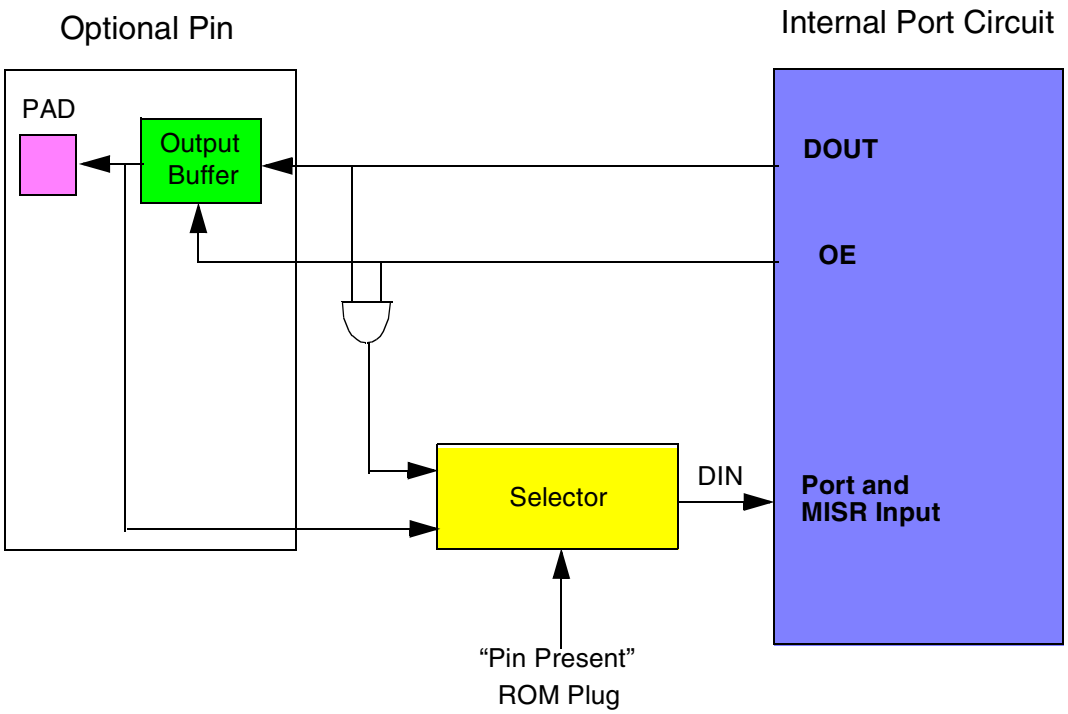


Figure 3-6 Port Loopback Logic on Optional Pins

The MISR is used to improve verification of optional pin’s primary functionality. To improve both observability and testability, all optional outputs are routed to the MISR.

3.2.4 Port A/B Operation

In master mode, MFTM, SLAM, and emulation mode, port A/B functions as external address bus pins A[15:0]. In single chip mode port A/B functions as digital I/O. These pin functions cannot be changed, as there is no pin assignment register. When configured as A[15:0], the BIM will enable the 3-V output drivers in the pads. When configured as digital I/O, the BIM will enable the 5-V output drivers in the pads.

The Port A/B registers are accessible via the IMB unless the BIM is in emulation mode.

Table 3-12 Port A/B Supported Pin Functions

PIN	Master	Single Chip	MFTM	SLAM	Emulation[1]
A[15:0] / PA[7:0], PB[7:0]	A[15:0]	PA[7:0], PB[7:0]	A[15:0]	A[15:0] (input)	A[15:0]

NOTES:
1. Digital I/O pin function provided by port replacement unit

3.2.4.1 Port A/B Data Direction Register (DDRAB)

DDRAB — Port A/B Data Direction Register **0xYF FA14**

MSB 7	6	5	4	3	2	1	LSB 0
RESERVED						DDA	DDB
RESET:							
0	0	0	0	0	0	0	0

Table 3-13 DDRAB Bit Descriptions

Bit(s)	Name	Description
7:2	—	Reserved
1	DDA	Port A data direction. When port A and B are assigned to digital I/O, the pins are configured as either input or output on a whole port basis. 0 = All port A pins are configured as inputs 1 = All port A pins are configured as outputs
0	DDB	Port B data direction. When port A and B are assigned to digital I/O, the pins are configured as either input or output on a whole port basis. 0 = All port B pins are configured as inputs 1 = All port B pins are configured as outputs

3.2.4.2 Port A Output Data Register (PORTA)

PORTA — Port A Output Data Register **0xYF FA10**

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

Table 3-14 PORTA Bit Descriptions

Bit(s)	Name	Description
7:0	PA[15:8]	Port A output data. PORTA is used to store the data to be driven on the port A output pins, when the port is configured as output. When read, the current value of the PORTA register, not the port A pins, is returned.

3.2.4.3 Port A Pin Data Register (PORTAP)

PORTAP — Port A Pin Data Register **0xYF FA12**

MSB 7	6	5	4	3	2	1	LSB 0
PA[15]/ PA[7]	PA[14]/ PA[6]	PA[13]/ PA[5]	PA[12]/ PA[4]	PA[11]/ PA[3]	PA[10]/ PA[2]	PA[9]/ PA[1]	PA[8]/ PA[0]
RESET: Not affected by resets.							
0	0	0	0	0	0	0	0

Table 3-15 PORTAP Bit Descriptions

Bit(s)	Name	Description
7:0	PA[7:0]	Port A pin data. When read, PORTAP reflects the current state of the port A pins. Writes to PORTAP have no effect.

3.2.4.4 Port B Output Data Register (PORTB)

PORTB — Port B Output Data Register **0xYF FA10**

MSB 7	6	5	4	3	2	1	LSB 0
PB[7]	PB[6]	PB[5]	PB[4]	PB[3]	PB[2]	PB[1]	PB[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-16 PORTB Bit Descriptions

Bit(s)	Name	Description
7:0	PB[7:0]	Port B output data. PORTB is used to store the data to be driven on the port B output pins, if the port is configured as an output. When read, the current value of the PORTB register, not the port B pins, is returned.

3.2.4.5 Port B Pin Data Register (PORTBP)

PORTBP — Port B Pin Data Register **0xYF FA12**

MSB 7	6	5	4	3	2	1	LSB 0
A[7]/ PB[7]	A[6]/ PB[6]]	A[5]/ PB[5]	A[4]/ PB[4]	A[3]/ PB[3]	A[2]/ PB[2]	A[1]/ PB[1]	A[0]/ PB[0]
RESET: Not affected by resets.							
0	0	0	0	0	0	0	0

Table 3-17 PORTBP Bit Descriptions

Bit(s)	Name	Description
7:0	A/PB[7:0]	Port B pin data. When read, PORTBP reflects the current state of the port B pins. Writes to PORTBP have no effect.

3.2.5 Port C Operation

In master mode and MFTM, port C can be configured as external address bus pins A[23:16] or as digital I/O. In single chip mode, all port C pins can be configured as external address bus pins A[23:16] or as digital I/O, but only digital I/O is supported.

Table 3-18 indicates the port C pin functions supported for each operation mode. Even when a pin function is not listed as supported for a particular mode, that pin function can still be selected with the port C pin assignment register. However, the pin activity is undefined for that case.

When a port C pin is configured as A[23:16], the BIM will enable the 3-V output driver in the associated pad. When a port C pin is configured as digital I/O, the BIM will enable the 5-V output driver in the associated pad.

Table 3-18 Port C Supported Pin Functions

PIN	Master	Single Chip	MFTM	SLAM	Emulation[1]
A[23] / PC[7]	A[23] or PC[7]	PC[7]	A[23] or PC[7]	PC[7]	A[23]
A[22:16] / PC[6:0]	A[22:16] or PC[6:0]	PC[6:0]	A[22:16] or PC[6:0]	PC[6:0]	A[22:16]

NOTES:
1. Digital I/O pin function provided by port replacement unit.

3.2.5.1 Port C Pin Assignment Register (PCPAR)

PCPAR — Port C Assignment Register **0xYF FA1E**

MSB 7	6	5	4	3	2	1	LSB 0
A[23]/ PC[7]	RESERVED				A[22:16]/ PC[6:0]		
RESET: See **Table 3-20**.							
0	0	0	0	0	0	0	0

Table 3-19 PCPAR Bit Descriptions

Bit(s)	Name	Description
7	A[23]/PC[7]	Port C configuration. This bit configures the port C[7] pin as either external address bus pin A[23], or digital I/O. When this bit is cleared, port C[7] is configured as digital I/O; when set port C[7] is configured as A[23].
6:3	—	Reserved
2:0	A[22:16]/PC[6:0]	Port C configuration encoding. The bits in this field configure port C[6:0] pins as either external address lines A[22:16] or digital I/O. The value stored in PCPAR[2:0] selects the number of contiguous pins configured as address lines. The address lines selected are contiguous from the least significant pin, A[16], forming a contiguous address space with A[15:0]. The I/O pins are contiguous from the most significant pin, port C[6]. See **Table 3-20** for port C pin assignment encodings.

Table 3-20 Port C[2:0] Pin Assignment Encoding

PCPAR[2:0]	Port C I/O Pins	Address Pins
0 0 0	PC[6:0]	None
0 0 1	PC[6:1]	A[16]
0 1 0	PC[6:2]	A[17:16]
0 1 1	PC[6:3]	A[18:16]
1 0 0	PC[6:4]	A[19:16]
1 0 1	PC[6:5]	A[20:16]
1 1 0	PC[6]	A[21:16]
1 1 1	None	A[22:16]

3.2.5.2 Port C Data Direction Register (DDRC)

DDRC — Port Data Direction Register **0xYF FA1C**

MSB 7	6	5	4	3	2	1	LSB 0
PC[7]	PC[6]	PC[5]	PC[4]	PC[3]	PC[2]	PC[1]	PC[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-21 DDRC Bit Descriptions

Bit(s)	Name	Description
7:0	PC[7:0]	Port C data direction. Port C digital I/O pins can be individually configured as either input or output. The bits in this register control the direction of the port C pin drivers when the pins are configured as I/O. When any bit in this register is set to one, the corresponding pin is configured as an output. When any bit in this register is cleared to zero, the corresponding pin is configured as an input. When the port is not defined to be digital I/O, its corresponding data direction bit has no effect.

3.2.5.3 Port C Output Data Register (PORTC)

PORTC — Port C Output Data Register **0xYF FA18**

MSB 7	6	5	4	3	2	1	LSB 0
PC[7]	PC[6]	PC[5]	PC[4]	PC[3]	PC[2]	PC[1]	PC[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-22 PORTC Bit Descriptions

Bit(s)	Name	Description
7:0	PC[7:0]	Port C output data. PORTC is used to store the data to be driven on the port C output pins, if the port is configured as output. When read, the current value of the PORTC register, not the port C pins, is returned.

3.2.5.4 Port C Pin Data Register (PORTCP)

PORTCP — Port C Pin Data Register **0xYF FA1A**

MSB 7	6	5	4	3	2	1	LSB 0
A[23]/ PC[7]	A[22]/ PC[6]]	A[21]/ PC[5]	A[20]/ PC[4]	A[19]/ PC[3]	A[18]/ PC[2]	A[17]/ PC[1]	A[16]/ PC[0]
RESET: Not affected by resets.							
0	0	0	0	0	0	0	0

Table 3-23 PORTCP Bit Descriptions

Bit(s)	Name	Description
7:0	A[23:16]/ PC[7:0]	Port C pin data. When read, PORTCP reflects the current state of the port C pins. Writes to PORTCP have no effect.

3.2.6 Port D Operation

When not in emulation mode, port D pins can be selected individually to be either digital I/O, or their primary function as chip select signals and background debug signals. If the BIM is in emulation mode, any access to port D registers is ignored (IAACK not asserted). In emulation mode, port D pins are configured as their primary function.

Table 3-24 indicates the port D pin functions supported for each operation mode. Even when a pin function is not listed as supported for a particular mode, that pin function can still be selected with the port D pin assignment register. However, the pin activity is undefined for that case.

When a port D pin is configured as its primary function, the BIM will enable the 3-V output driver in the associated pad. When a port D pin is configured as digital I/O, the BIM will enable the 5-V output driver in the associated pad.

Table 3-24 Port D Supported Pin Function s

PIN	Master	Single Chip	MFTM	SLAM	Emulation[1]
	Background	Background	Background	Background	Background
$\overline{CS3}$-$\overline{CS1}$ / PD[7:5]	$\overline{CS3}$-$\overline{CS1}$ or PD[7:5]	PD[7:5]	$\overline{CS3}$-$\overline{CS1}$ or PD[7:5]	$\overline{CS3}$-$\overline{CS1}$	$\overline{CS3}$-$\overline{CS1}$
	$\overline{CS3}$-$\overline{CS1}$ PD[7:5]	PD[7:5]	$\overline{CS3}$-$\overline{CS1}$ PD[7:5]	NA	$\overline{CS3}$-$\overline{CS1}$
FREEZE / QUOT / CSE1 / PD[4]	FREEZE or PD[4]	FREEZE or PD[4]	FREEZE or QUOT[2] or PD[4]	FREEZE (input) or QUOT[2]	FREEZE / CSE1[3]
	FREEZE	FREEZE	FREEZE	NA	FREEZE / CSE1[3]
$\overline{BKPT}$ / DSCLK / PD[3]	$\overline{BKPT}$ / DSCLK[3] or PD[3]	$\overline{BKPT}$ / DSCLK[3] or PD[3]	$\overline{BKPT}$ / DSCLK[3] or PD[3]	PD[3]	$\overline{BKPT}$ / DSCLK[3]
	$\overline{BKPT}$ / DSCLK[3]	$\overline{BKPT}$ / DSCLK[3]	$\overline{BPT}$ / DSCLK[3]	NA	$\overline{BKPT}$ / DSCLK[3]
IPIPE2 / PD[2]	IPIPE2 or PD[2]	IPIPE2 or PD[2]	IPIPE2 or PD[2]	PD[2]	IPIPE2
	IPIPE2 or PD[2]	IPIPE2 or PD[2]	IPIPE2 or PD[2]	NA	IPIPE2 or PD[2]
$\overline{IPIPE1}$ / DSI / PD[1]	IPIPE1 or PD[1]	IPIPE1 or PD[1]	IPIPE1 or PD[1]	LATCH (input)	IPIPE1
	DSI	DSI	DSI	NA	DSI
$\overline{IPIPE0}$ / DSO / PD[0]	$\overline{IPIPE0}$ or $\overline{PD[0]}$	$\overline{IPIPE0}$ or $\overline{PD[0]}$	IPIPE0 or $\overline{PD[0]}$	$\overline{CYS}$ (input)	IPIPE0
	DSO	DSO	DSO	NA	DSO

NOTES:
1. Digital I/O pin function provided by port replacement unit.
2. Selected by bit in module test control register (CREG).
3. $\overline{BKPT}$ becomes DSCLK after FREEZE is asserted.

3.2.6.1 Port D Pin Assignment Register (PDPAR)

PDPAR — Port D Pin Assignment Register **0xYF FA1E**

MSB 7	6	5	4	3	2	1	LSB 0
$\overline{CS3}$-$\overline{CS1}$/PD[7:5]			FREEZE/ QOUT CSE1/PD[4]	$\overline{BKPT}$/ $\overline{DSCLK}$/PD[3]	$\overline{IPIPE2}$/ PD[2]	$\overline{PIPE1}$/ DSI/ LATCH/PD[1]	$\overline{PIPE0}$/ DSO/ $\overline{CYS}$/ PD[0]
RESET: See **Table 3-3**.							
0	0	0	0	0	0	0	0

Table 3-25 PDPAR Bit Descriptions

Bit(s)	Name	Description
7:5	—	Port D pin function. The bits in this register control the function of each pin of port D. 0 = Corresponding pin is configured as a digital I/O pin 1 = Corresponding pin is used for its primary function

3.2.6.2 Port D Data Direction Register (DDRD)

DDRD — Port D Data Direction Register **0xYF FA1C**

MSB 7	6	5	4	3	2	1	LSB 0
PD[7]	PD[6]	PD[5]	PD[4]	PD[3]	PD[2]	PD[1]	PD[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-26 DDRD Bit Descriptions

Bit(s)	Name	Description
7:0	PD[7:0]	Port D data direction. The bits in this register control the direction of the port D pin drivers when the pins are configured as I/O. When the port is not defined to be digital I/O, its corresponding data direction bit has no effect. 0 = Corresponding pin is configured as an input 1 = Corresponding pin is configured as an output

3.2.6.3 Port D Output Data Register (PORTD)

PORTD — Port D Output Data Register **0xYF FA1A**

MSB 7	6	5	4	3	2	1	LSB 0
PD[7]	PD[6]	PD[5]	PD[4]	PD[3]	PD[2]	PD[1]	PD[0]
RESET:							
1	1	1	1	1	1	1	1

Table 3-27 PORTD Bit Descriptions

Bit(s)	Name	Description
7:0	PD[7:0]	Port D output data. PORTD is used to latch the data to be driven on the port D output pins, if the port is configured as output. When read, the current value of the PORTD register, not the port D pins, is returned.

3.2.6.4 Port D Pin Data Register (PORTDP)

PORTDP — Port D Pin Data Register **0xYF FA1A**

MSB 7	6	5	4	3	2	1	LSB 0
$\overline{CS3}$/PD[7]	$\overline{CS2}$/PD[6]	$\overline{CS1}$/PD[5:]	FREEZE/ QOUT CSE1/PD[4]	$\overline{BKPT}$/ DSCLKPD[3]	$\overline{IPIPE2}$/ PD[2]	$\overline{PIPE1}$/ DSI/ LATCH/PD[1]	$\overline{PIPE0}$/ $\overline{DSO}$/ $\overline{CYS}$/
RESET: Not affected by RESET.							
0	0	0	0	0	0	0	0

Table 3-28 PORTDP Bit Descriptions

Bit(s)	Name	Description
7:0	—	Port D pin data. When read, PORTDP reflects the current state of the port D pins. Writes to PORTDP have no effect.

3.2.7 Port E Operation

In master mode and MFTM, Port E pins can be selected individually to be either digital I/O, or external bus control/handshake signals. In single chip mode, port E can function as either digital I/O, or external bus control/handshake signals, but only digital I/O is supported. In SLAM, port E pins [7:5] are configured as either $\overline{CS7}$-$\overline{CS5}$ or PC[7:5], since the function codes are supplied on the address pins. In SLAM, pin [4] is typically configured as a clock signal, although the pin can function as PE[4] once the tester has been synchronized. In SLAM, pins [2:0] are configured as cycle control signals. Since the function codes pins are not used in SLAM, these external pins can be selected as either $\overline{CS7}$-$\overline{CS5}$ or PE[7:5]. In emulation mode, port E pins are configured as their primary chip select, bus control, and clock functions.

Table 3-29 indicates the port E pin functions supported for each operation mode. Even when a pin function is not listed as supported for a particular mode, that pin function can still be selected with the port E pin assignment register. However, the pin activity is undefined for that case.

When a port E pin is configured as its primary function, the BIM will enable the 3-V output driver in the associated pad. When a port E pin is configured as digital I/O, the BIM will enable the 5-V output driver in the associated pad.

Table 3-29 Port E Supported Pin Functions

Pin	Master	Single Chip	MFTM	SLAM	Emulation[1]
FC[2:0] / $\overline{CS7\text{-}CS5}$ / PE[7:5]	FC[2:0] or $\overline{CS7\text{-}CS5}$ or PE[7:5]	PE[7:5]	FC[2:0] or $\overline{CS7\text{-}CS5}$ or PE[7:5]	CS7-CS5 or PE[7:5]	CS7-CS5
CLKOUT / PE[4]	CLKOUT or PE[4]	CLKOUT or PE[4]	CLKOUT or PE[4]	CLKOUT or PE[4]	CLKOUT
SIZE / BLOCK PE[3]	SIZE or PE[3]	PE[3]	SIZE or PE[3]	BLOCK (input)	SIZE
$\overline{AS}$ / PE[2]	AS or $\overline{PE[2]}$	PE[2]	AS or $\overline{PE[2]}$	AS (input)	AS
$\overline{DS}$ / PE[1]	DS or $\overline{PE[1]}$	PE[1]	DS or $\overline{PE[1]}$	DS (input)	DS
R/$\overline{W}$ / PE[0]	R/$\overline{W}$ or PE[0]	PE[0]	R/$\overline{W}$ or PE[0]	R/$\overline{W}$ (input)	R/$\overline{W}$

NOTES:
1. Digital I/O pin function provided by port replacement unit.

3.2.7.1 Port E Pin Assignment Register (PEPAR)

PEPAR — Port E Pin Assignment Register **0xYF FA26**

MSB 7	6	5	4	3	2	1	LSB 0
FC[2:0]/PD[7:5] CS7-5/PE[7:5]		Reserved	CLK-OUT/PE[4]	SIZE/BLOCK /PE[3]	$\overline{AS}$/PE[2]	$\overline{DS}$/PE[1]	R/$\overline{W}$ / PE[0]
RESET: See Table 3-3.							
0	0	0	0	0	0	0	0

Table 3-30 PEPAR Bit Descriptions

Bit(s)	Name	Description
7:6	FC[2:0]/ PD[7-5], CS7-5, PE[7:5]	Port E pin function. The bits in this field configure the port E[7:5] pins as either function codes, chip selects, or digital I/O. 00 = PE[7:5] 01 = Reserved 10 = FC[2:0] 11 = CS7-CS5
5	—	Reserved
4:0	—	Port E pin function. The bits in this field control the function of port E[4:0]. 0 = Corresponding pin is configured as an I/O pin 1 = Corresponding pin is configured as its primary function

3.2.7.2 Port E Data Direction Register (DDRE)

DDRE — Port E Data Direction Register **0xYF FA24**

MSB 7	6	5	4	3	2	1	LSB 0
PE[7]	PE[6]	PE[5]	PE[4]	PE[3]	PE[2]	PE[1]	PE[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-31 DDRE Bit Descriptions

Bit(s)	Name	Description
7:0	PE[7:0]	Port E data direction. The bits in this register control the direction of the port E pin drivers when the pins are configured as I/O. When the port is not defined to be digital I/O, its corresponding data direction bit has no effect. 0 = Corresponding pin is configured as an input 1 = Corresponding pin is configured as an output

3.2.7.3 Port E Output Data Register (PORTE)

PORTE — Port E Output Data Register **0xYF FA20**

MSB 7	6	5	4	3	2	1	LSB 0
PE[7]	PE[6]	PE[5]	PE[4]	PE[3]	PE[2]	PE[1]	PE[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-32 PORTE Bit Descriptions

Bit(s)	Name	Description
7:0	PE[7:0]	Port E output direction. PORTE is used to latch the data to be driven on the port E output pins, if the port is configured as output. When read, the current value of the PORTE register, not the port E pins, is returned.

3.2.7.4 Port E Pin Data Register (PORTEP)

PORTEP — Port E Pin Data Register **0xYF FA22**

MSB 7	6	5	4	3	2	1	LSB 0
FC[2]/$\overline{CS7}$/ PE[7]	FC[1]/$\overline{CS6}$/ PE[6]	FC[0]/$\overline{CS5}$/ PE[5:]	CLKOUT/ PE[4]	SIZE/PE[3]	$\overline{AS}$/PE[2]	$\overline{DS}$/PE[1]	R/$\overline{W}$ / PE[0]
RESET: Not affected by RESET.							
0	0	0	0	0	0	0	0

Table 3-33 PORTEP Bit Descriptions

Bit(s)	Name	Description
7:0	—	Port E pin data. When read, PORTEP reflects the current state of the port E pins. Writes to PORTEP have no effect.

3.2.8 Port F Operation

This section Table 3-34 indicates the port F pin functions supported for each operation mode. Even when a pin function is not listed as supported for a particular mode, that pin function can still be selected with the port F pin assignment register. However, the pin activity is undefined for that case.

When a port F pin is configured as a level or edge sensitive interrupt, the BIM will enable the 3-V output driver in the associated pad. When a port F pin is configured as an edge detect digital I/O, the BIM will enable the 5-V output driver in the associated pad.

Unlike the other ports, port F registers do respond when the BIM is in emulation mode. The pin function can be changed in emulation mode by accessing the port F pin assignment register.

Table 3-34 Port F Supported Pin Functions

Pin	Master	Single Chip	MFTM	SLAM	Emulation
$\overline{\text{IRQX}}$ / $\overline{\text{BERR}}$ / PF[7]	IRQX or $\overline{\text{BERR}}$ or PF[7]	IRQX or PF[7]	$\overline{\text{IRQX}}$* or $\overline{\text{BERR}}$ or PF[7]	IIRQX (out) or PF[7]	IRQX or $\overline{\text{BERR}}$ or PF[7]
$\overline{\text{IRQ}}$[6:5,3:1] / PF[6:5,3:1]	IRQ[6:5,3:1] or PF[6:5,3:1]	IRQ[6:5,3:1] or PF[6:5,3:1]	IRQ[6:5,3:1] or PF[6:5,3:1]	IIRQ[6:5,3:1] (out) or PF[6:5,3:1]	IRQ[6:5,3:1] PF[6:5,3:1]
$\overline{\text{IRQ4}}$ / PF[4]	IRQ4 or PF[4]	IRQ4 or PF[4]	IRQ4 or PF[4]	IIRQ4 (out) or PF[4]	IRQ4 or PF[4]
$\overline{\text{CS4}}$ / PF[0]	CS4 or PF[0]	PF[0]	CS4 or PF[0]	CS4 or PF[0]	CS4 or PF[0]

In all operation modes, port F pins can be selected individually to be either digital I/O or their primary function ($\overline{\text{IRQ}}$ or $\overline{\text{CS4}}$). An additional feature of port F is that software may generate both $\overline{\text{IRQ}}$ service requests or digital edge detect interrupt service requests using the loopback logic.

If the SHIRQ bit is set in the BIMTR, port F must be programmed for $\overline{\text{IRQ}}$ operation to allow the internal IIRQ signals to be output. In SLAM, the $\overline{\text{IRQ}}$ pin function is limited to Show $\overline{\text{IRQ}}$ signals, regardless of the SHOW_IRQ bit setting in the BIMTR; the normal $\overline{\text{IRQ}}$ input pin function is not supported.

Table 3-35 Port F Master, Single Chip, MFTM, and Emulation Pin Operation

Pin(s)	Level-Sensitive Interrupt Pin PAR[i+1, i] = 11	Edge-Sensitive Interrupt Pin PAR[i+1, i] = 00	Digital I/O with Edge-Detect PAR[i+1, i] = 01 or 10
$\overline{IRQX}$ / PF[7] (IRQXL = 7)	If the $\overline{IRQX}$ pin is held low for two clocks, the BIM asserts the IMB IRQ7 line. The pin must remain low until the level 7 IACK cycle is initiated. The pin should be returned high before exception processing is completed. Additional IRQ7 level-sensitive interrupts are not recognized until a falling edge is detected to prevent redundant servicing.	If a high-to-low transition is detected on the $\overline{IRQX}$ pin, the BIM latches the condition and asserts the IMB IRQ7 line. The $\overline{IRQX}$ pin is not required to remain low until the level 7 IACK cycle is initiated. The latched condition is cleared when the BIM wins arbitration for the level 7 IACK cycle.	The IRQXL field has no affect when the PFPAR register selects PF[7]. If the edge specified in the PFPAR register for the PF[7] is detected, the BIM sets PORTFE[7]. If PFEER[7] is set, the BIM also asserts the IMB IRQ line corresponding to the priority level specified in PFEL field. The pin is not required to remain asserted. The latched condition is cleared when the user writes a zero to PORTFE[7].
$\overline{IRQX}$ / PF[7] (IRQXL < 7 and IRQXL > 0)	If the IRQXL field in the PFLVR register is less than seven and not zero, and the $\overline{IRQX}$ pin is held low for two clocks, the BIM asserts the IMB IRQ line corresponding to the priority level specified by IRQXL. The pin must remain low until an IACK cycle for that level is initiated and the BIM wins arbitration. The pin should be returned high before exception processing is completed.	If the IRQXL field in the PFLVR register is less than seven and not zero, and a high-to-low transition is detected on the $\overline{IRQX}$ pin, the BIM latches the condition and asserts the IMB IRQ line corresponding to the priority level specified by IRQXL. The $\overline{IRQX}$ pin is not required to remain low until an IACK cycle for that level is initiated.The latched condition is cleared when the BIM wins arbitration for the IACK cycle.	The IRQXL field has no affect when the PFPAR register selects PF[7]. If the edge specified in the PFPAR register for the PF[7] is detected, the BIM sets PORTFE[7]. If PFEER[7] is set, the BIM also asserts the IMB IRQ line corresponding to the priority level specified in PFEL field. The pin is not required to remain asserted. The latched condition is cleared when the user writes a zero to PORTFE[7].
BERR / PF[7] (IRQXL = 0)	If IRQXL = 0, $\overline{IRQX}$ is configured as the $\overline{BERR}$ input pin.	If IRQXL = 0, $\overline{IRQX}$ is configured as the $\overline{BERR}$ input pin.	The IRQXL field has no affect when the PFPAR register selects digital I/O (PF[7]). If the edge specified in the PFPAR register for PF[7] is detected, the BIM sets PORTFE[7]. If PFEER[7] is set, the BIM also asserts the IMB IRQ line corresponding to the priority level specified in PFEL field. The pin is not required to remain asserted. The latched condition is cleared when the user writes a zero to PORTFE[7].

Table 3-35 Port F Master, Single Chip, MFTM, and Emulation Pin Operation

Pin(s)	Level-Sensitive Interrupt Pin PAR[i+1, i] = 11	Edge-Sensitive Interrupt Pin PAR[i+1, i] = 00	Digital I/O with Edge-Detect PAR[i+1, i] = 01 or 10
IRQ[6:1]	The priority level of interrupt request inputs $\overline{\text{IRQ}}$[6:1] is fixed to correspond to the interrupt pin number (e.g., $\overline{\text{IRQ}}$[6] has a fixed priority level of 6). If held low for two clocks, the BIM asserts the IMB IRQ line corresponding to the Interrupt Input pin number.The pin must remain low until an IACK cycle for that level is initiated and the BIM wins arbitration.The pin should be returned high before exception processing is completed.	If a high-to-low transition is detected, the BIM latches the condition and initiates interrupt exception processing at the priority level corresponding to the interrupt input pin number. The pin is not required to remain low. The latched condition is cleared when the BIM recognizes an IACK cycle for the specified priority level and wins the arbitration.	If the edge specified in the PFPAR register for the specific pin is detected, the BIM will set the corresponding flag bit in the PORTFE register. If the corresponding enable bit in PFEER is set, the BIM also asserts the IMB IRQ line corresponding to the priority level specified in PFEL field. The pin is not required to remain asserted. The latched condition is cleared when the user writes a zero to the corresponding flag bit in PORTFE[6:1].
CS4 / PF[0]	If not programmed as a rising or falling edge-detect digital I/O pin, the CS4 pin operates as chip select. Cannot be programmed to operate as a level-sensitive interrupt request pin.	If not programmed as a rising or falling edge-detect digital I/O pin, the CS4 pin operates as chip select. Cannot be programmed to operate as a level-sensitive interrupt request pin.	If the edge specified in the PFPAR register for the PF[0] is detected, the BIM sets PORTFE[0]. If PFEER[0] is set, the BIM also asserts the IMB IRQ line corresponding to the priority level specified in PFEL field. The pin is not required to remain asserted. The latched condition is cleared when the user writes a zero to PORTFE[0].

All edge detect pin functions require the external pin to be asserted for a minimum of one clock plus the asynchronous set-up time. This allows port F circuitry to synchronize and latch the interrupt request to the BIM's internal clock.

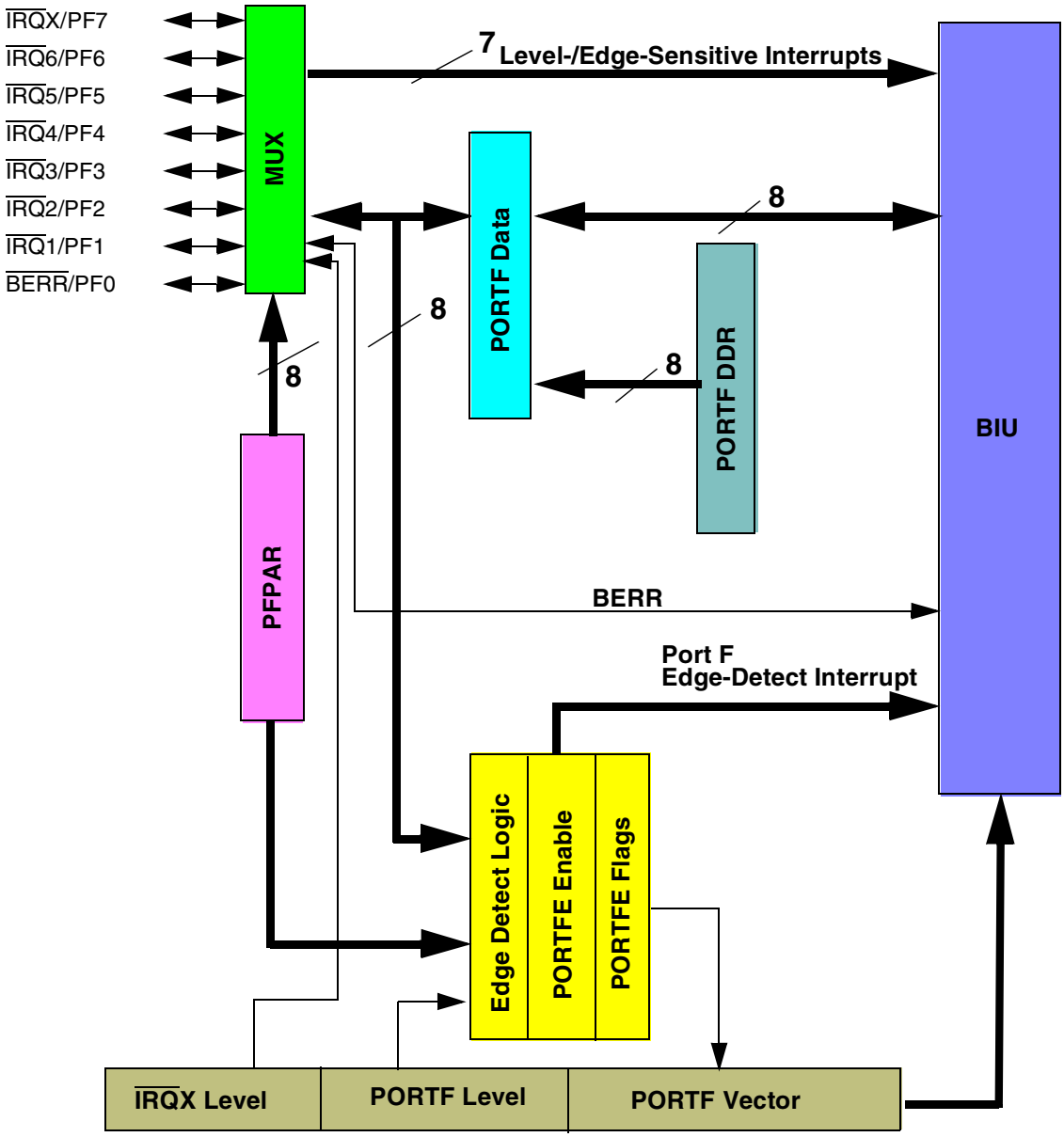


Figure 3-7 PORT F Register Block Diagram

3.2.8.1 Port F Pin Assignment Register (PFPAR)

PFPAR — Port F Pin Assignment Register **0xYF FA36**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
$\overline{IRQX}$ / $\overline{BERR}$ / PF[7]			$\overline{IRQ6}$ / PF[6]			$\overline{IRQ5}$ /PF[5]		$\overline{IRQ4}$ /PF[4]		$\overline{IRQ3}$ /PF[3]		$\overline{IRQ2}$ /PF[2]		$\overline{IRQ1}$ /PF[1]	$\overline{CS4}$ / PF[0]
RESET: See **Table 3-3**. Note that PCON[6] = 0 selects the 01 (rising edge detect I/O pin) encoding and PCON[6] = 1 selects the 11 ($\overline{CS4}$) encoding for PFPAR[1:0]. The Level-Sensitive Interrupt Request pin encoding (11) is selected at reset for PFPAR[15:2].															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3-36 PFPAR Bit Descriptions

Bit(s)	Name	Description
15:0	—	Port F pin assignment. The eight fields in this register each control the function of a pin in port F. In all operation modes, the field encoding 01 and 10 configure the associated pin as a digital I/O pin with rising or falling edge-detect. In master, single chip, MFTM, and emulation modes, the field encoding 11, configures the corresponding pin as a level-sensitive interrupt request for pins [7:1] and $\overline{CS4}$ on pin [0]. The field encoding 00, configures the corresponding pin as a falling edge-sensitive interrupt request for pins [7:1] and $\overline{CS4}$ on pin [0]. To select $\overline{BERR}$, the field must be set to either 00 or 11 and the IRQXL field in PFLVR must be set to zero. As an alternate pin function for $\overline{IRQX}$, and $\overline{IRQ[6:1]}$, the internal IMB $\overline{IRQ}$ lines can be output on these external pins and the value of the internal $\overline{IRQ}$ lines can also be read in the port F pin data register (PORTFP). In SLAM, the encodings 00 or 11 always select this alternate pin function. In MFTM and in background debug mode, this alternate pin function can be selected by setting the SHIRQ bit in the BIMTR. 00 = Falling edge sensitive interrupt request (pins 7 to 1), $\overline{CS4}$ (pin 0) 01 = Digital I/O with rising edge detect 10 = Digital I/O with falling edge detect 11 = Level-sensitive interrupt request (pins 7 to 1), $\overline{CS4}$ (pin 0)

3.2.8.2 Port F Data Direction Register (DDRF)

DDRF — Port F Data Direction Register **0xYF FA34**

MSB 7	6	5	4	3	2	1	LSB 0
PF[7]	PF[6]	PF[5]	PF[4]	PF[3]	PF[2]	PF[1]	PF[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-37 DDRF Bit Descriptions

Bit(s)	Name	Description
7:0	PF[7:0]	Port F data direction. The bits in this register control the direction of the port F pin drivers when the pins are configured as digital I/O pins. When any bit in this register is set to one, the corresponding pin is configured as an output. When any bit in this register is cleared to zero, the corresponding pin is configured as an input. Unlike the other ports, DDRF also affects port F operation when the PFPAR selects the primary pin function, $\overline{IRQ}$. When $\overline{IRQ}$ is selected and interrupt requests are generated by the $\overline{IRQ}$ pin, the DDRF bit must select input. This is the common programming choice for customer applications. When $\overline{IRQ}$ is selected and interrupt requests are generated by programming the PORTF output data register, the DDRF must select output. When $\overline{IRQ}$ is selected and SHIRQ is programmed in the BIMTR, the DDRF must select output to allow optional Port F pins to be read in PORTFP. Then the loopback logic allows testing of the $\overline{IRQ}$ functionality regardless of whether the pin is present. This is the usual programming for functional test patterns. It is also useful for customer applications that need to generate software interrupts at various interrupt levels. In SLAM, or when the SHIRQ bit is set during MFTM or background debug mode, the DDRF has no affect when PFPAR selects the primary pin function,$\overline{IRQ}$. In this case, the internal IMB $\overline{IRQ}$ lines are reflected on the external pin and can be read in PORTFP. The DDRF[0] has no affect when the CS4 pin function is selected in PFPAR.t

3.2.8.3 Port F Output Data Register (PORTF)

PORTF — Port F Output Data Register **0xYF FA30**

MSB 7	6	5	4	3	2	1	LSB 0
PF[7]	PF[6]	PF[5]	PF[4]	PF[3]	PF[2]	PF[1]	PF[0]
RESET:							
1	1	1	1	1	1	1	1

Table 3-38 PORTF Bit Descriptions

Bit(s)	Name	Description
7:0	PF[7:0]	Port F output data. The PORTF register is used to store the data to be driven on the port F output pins. When read, the current value of the PORTF register, not the PORTF pins, is returned.

3.2.8.4 Port F Pin Data Register (PORTFP)

PORTFP — Port F Pin Data Register **0xYF FA32**

MSB 7	6	5	4	3	2	1	LSB 0
$\overline{IRQ}$X/$\overline{BERR}$/ PF[7]	$\overline{IRQ}$6/PF[6]	$\overline{IRQ}$5/PF[5:]	$\overline{IRQ}$4/PF[4]	$\overline{IRQ}$3/PF[3]	$\overline{IRQ}$2/PF[2]	$\overline{IRQ}$1/PF[1]	CS4/PFE[0]
RESET: Not affected by RESET.							
0	0	0	0	0	0	0	0

Table 3-39 PORTFP Bit Descriptions

Bit(s)	Name	Description
7:0	—	Port F data input. When read, PORTFP reflects the current state of the port F pins. Writes to PORTFP have no effect.

3.2.8.5 Port F Edge-Detect Flag Register (PORTFE)

PORTFE — Port F Edge-Detect Flag Register **0xYF FA38**

MSB 7	6	5	4	3	2	1	LSB 0
EF7	EF6	EF5	EF4	EF3	EF2	EF1	EF0
RESET:							
0	0	0	0	0	0	0	0

Table 3-40 PORTFE Bit Descriptions

Bit(s)	Name	Description
7:0	EF[7:0]	Port F edge detect flag. The port F edge-detect flag register (PORTFE) indicates that the programmed transition has occurred on Port F input or output pins. The edge-detect flag bits in PORTFE are set if the specified edge is detected on the corresponding port F pin, when the corresponding port F pin is configured as an input or output edge-detect pin. If a pin is configured as an $\overline{IRQ}$, $\overline{BERR}$, or $\overline{CS4}$, the flag for that pin is not set on a transition. If a pin transition occurs as a result of changing the PAR encoding from the primary pin function to digital I/O, an edge will be detected. Also, an interrupt can optionally be generated after the proper transition has been detected. An interrupt request is generated whenever the enable bit in the port F edge-detect I/O Interrupt enable register (PFEER) for the corresponding pin is a one, and the port F edge-detect level field in the port F interrupt level register (PFLVR) is non-zero. The bit in the PORTFE, PFEER, and the PFLVR field can be set in any order to generate the interrupt request. When the interrupt is serviced, the interrupt service routine may read PORTFE and logically AND it with PFEER to determine which pin(s) caused the interrupt. Once set, the edge-detect flag bits remains set, regardless of the subsequent state of the corresponding pin or changes in PAR programming, until the bit is cleared by software, or reset. To clear an edge-detect flag, the bit must be first read as a one and then the bit can be written to zero. Flags which are zero when the register is read are unaffected by the write operation. Also, if the edge detect logic detects another edge after the flag was read as a one and before a zero is written to clear it, the flag cannot be cleared until the flag is read as a one again and written to a zero. Bits in this register are set by the BIM and cannot be written to a one by software.

3.2.8.6 Port F Edge-Detect I/O Interrupt Enable (PFEER)

PFEER — Port F Edge-Detect I/O Register **0xYF FA3A**

MSB 7	6	5	4	3	2	1	LSB 0
PFEE7	PFEE6	PFEE5	PFEE4	PFEE3	PFEE2	PFEE1	PFEE0
RESET:							
0	0	0	0	0	0	0	0

Table 3-41 PFEER Bit Descriptions

Bit(s)	Name	Description
7:0	PFEER [7:0]	Port F I/O interrupt enable. The bits in this register individually enable interrupts by the corresponding edge-detect I/O pins. If a PFEER bit is written to one and the corresponding bit in the PORTFE register is set (or later becomes set when the specified edge occurs on the corresponding pin), an interrupt request is generated at the interrupt priority level specified by the PFEL[2:0] field of the port F interrupt level register. When the interrupt is serviced, the interrupt service routine may read PORTFE and logically AND it with PFEER to determine which pin(s) caused the interrupt. If the PFEER bit is written to zero, any pending I/O interrupt request is disabled. Also when specified edge occurs on the corresponding pin, an interrupt request is not generated. However, the PORTFE bit is set to record the edge event.

3.2.8.7 Port F Interrupt Level Register (PFLVR)

PFLVR — Port F Interrupt Level Register **0xYF FA3C**

MSB 7	6	5	4	3	2	1	LSB 0
0	IRQXL[2:0]			0	PFEL[2:0]		
RESET:							
0	1	1	1	0	0	0	0

Table 3-42 PFLVR Bit Descriptions

Bit(s)	Name	Description
7	—	Reserved
6:4	IRQXL	Port F priority level assignment. When the $\overline{\text{IRQX}}$ pin is configured as a level or edge sensitive IRQ pin (PFPAR[15:14] = 00 or 11), the IRQXL field specifies the interrupt request level of the IRQX pin, or defines the pin to be the $\overline{\text{BERR}}$ input. If IRQXL is not equal to 000, the pin functions as a level-sensitive or edge-sensitive interrupt request, generating the programmed level interrupt request when asserted. If IRQXL = 000, the pin functions as the $\overline{\text{BERR}}$ pin. The IRQXL field may be changed at any time. However, if the programmed edge or level change occurs on $\overline{\text{IRQX}}$ when the IRQXL field is being changed, the priority level of the new interrupt request is unpredictable. If an interrupt request is pending at the current level programmed in IRQXL, changing the value of IRQXL has no affect on this pending interrupt. However, future interrupt events on the IRQXL pin are reported at the new interrupt request level. The IRQXL field comes out of reset containing the value 111, which configures the pin as a level 7 interrupt pin ($\overline{\text{IRQ7}}$). This field has no affect if the pin function is programmed to be PF[7] (PFPAR[15:14] = 10 or 01).
3	—	Reserved
2:0	PFEL	Port F edge-detect I/O interrupt level. The interrupt priority level for all port F edge-detect I/O interrupts is determined by the priority level specified in the PFEL field. PFEL is reset to 0x0, which disables interrupt requests for the I/O edge detect logic. PFEL can be programmed to a value 0x0 (interrupt disabled) through 0x7 (highest priority). The port F edge-detect interrupt has the lowest arbitration priority in the BIM, when other BIM sources of interrupts are arbitrating at the same priority level. The PFEL field controls the interrupt request level of any pending digital I/O interrupts, even if the PFPAR is later programmed to select the primary pin function. The interrupt request level of pending digital I/O interrupts can be changed at any time.

3.2.9 Port G/H Operation

In master mode, emulation mode, MFTM, and SLAM, Port G/H functions as external data bus pins D[15:0]. In single chip mode, port G/H functions as digital I/O; each pin can be individually configured as either input or output. These pin functions cannot be changed, as there is no pin assignment register.

When a port G/H pin is configured as D[15:0], the BIM will enable the 3-V output driver in the associated pad. When a port G/H pin is configured as digital I/O, the BIM will enable the 5-V output driver in the associated pad.

The port G/H registers are accessible via the IMB unless the BIM is in emulation mode.

Table 3-43 Port G/H Supported Pin Functions

Pin	Master	Single Chip	MFTM	SLAM	Emulation[1]
D[15:0] / PG[7:0], PH[7:0]	D[15:0]	PG[7:0], PH[7:0]	D[15:0]	D[15:0]	D[15:0]

NOTES:
1. Digital I/O pin function provided by port replacement unit.

3.2.9.1 Port G Data Direction Register (DDRG)

DDRG — Port G Data Direction Register **0xYF FA2C**

MSB 7	6	5	4	3	2	1	LSB 0
PG[7]	PG[6]	PG[5]	PG[4]	PG[3]	PG[2]	PG[1]	PG[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-44 DDRG Bit Descriptions

Bit(s)	Name	Description
7:0	PG[7:0]	Port G data direction. The bits in this register control the direction of the port G pin drivers when the pins are configured as digital I/O. When the port is not defined to be digital I/O, its corresponding data direction bit has no effect. 0 = Corresponding pin is configured as an input 1 = Corresponding pin is configured as an output

3.2.9.2 Port G Output Data Register (PORTG)

PORTG — Port G Output Data Register **0xYF FA28**

MSB 7	6	5	4	3	2	1	LSB 0
PG[7]	PG[6]	PG[5]	PG[4]	PG[3]	PG[2]	PG[1]	PG[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-45 PORTG Bit Descriptions

Bit(s)	Name	Description
7:0	PG[7:0]	Port G output data. The PORTG register is used to store the data to be driven on the port G output pins, if the port is configured as output. When read, the current value of the PORTG register, not the port G pins, is returned.

3.2.9.3 Port G Pin Data Register (PORTGP)

PORTGP — Port G Pin Data Register **0xYF FA2A**

MSB 7	6	5	4	3	2	1	LSB 0
D[15]/PG[7]	D[14]/PG[6]	D[13]/PG[5:]	D[12]/ PG[4]	D[11]/PG[3]	D[10]/PG[2]	D[9]/PG[1]	D[8]/ PG[0]
RESET: Not affected by RESET.							
0	0	0	0	0	0	0	0

Table 3-46 PORTGP Bit Descriptions

Bit(s)	Name	Description
7:0	—	Port G data input. When read, the PORTGP register reflects the current state of the port G pins. Writes to PORTGP have no effect.

3.2.9.4 Port H Data Direction Register (DDRH)

DDRH — Port H Data Direction Register **0xYF FA2C**

MSB 7	6	5	4	3	2	1	LSB 0
PH[7]	PH[6]	PH[5]	PH[4]	PH[3]	PH[2]	PH[1]	PH[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-47 DDRH Bit Descriptions

Bit(s)	Name	Description
7:0	PH[7:0]	Port H data direction. The bits in this register control the direction of the port H pin drivers when the pins are configured as I/O. When the port is not defined to be digital I/O, its corresponding data direction bit has no effect. 0 = Corresponding pin is configured as an input 1 = Corresponding pin is configured as an output

3.2.9.5 Port H Output Data Register (PORTH)

PORTH — Port H Output Data Register **0xYF FA28**

MSB 7	6	5	4	3	2	1	LSB 0
PH[7]	PH[6]	PH[5]	PH[4]	PH[3]	PH[2]	PH[1]	PH[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-48 PORTH Bit Descriptions

Bit(s)	Name	Description
7:0	PH[7:0]	Port H output data. PORTH is used to store the data to be driven on the port H output pins, if the port is configured as output. When read, the current value of the PORTH register, not the port H pins, is returned.

3.2.9.6 Port H Pin Data Register (PORTHP)

PORTHP — Port H Pin Data Register **0xYF FA2A**

MSB 7	6	5	4	3	2	1	LSB 0
D[7]/PH[7]	D[6]/PH[6]	D[5]/PH[5:]	D[4]/ PH[4]	D[3]/PH[3]	D[2]/PH[2]	D[1]/PH[1]	D[0]/ PH[0]
RESET: Not affected by RESET.							
0	0	0	0	0	0	0	0

Table 3-49 PORTHP Bit Descriptions

Bit(s)	Name	Description
7:0	—	Port H data input. When read, PORTHP reflects the current state of the port H pins. Writes to PORTHP have no effect.

3.2.10 Port K Operation

In master mode and MFTM, port K can be configured as burst control/handshake signals or as digital I/O. In single chip mode, all port K pins can be configured as burst control/handshake signals or digital I/O, but only digital I/O is supported. In SLAM, port K pins DTACK, BREQ, and BTACK are used as cycle control signals. The other pins can be configured as burst chip select signals, $\overline{\text{DTACK}}$, or digital I/O. In emulation mode the pins are configured as burst control/handshake signals and BCS functions.

Table 3-50 indicates the port K pin functions supported for each operation mode. Even when a pin function is not listed as supported for a particular mode, that pin function can still be selected with the port K pin assignment register. However, the pin activity is undefined for that case.

When a port K pin is configured as its primary function, the BIM will enable the 3-V output driver in the associated pad. When a port K pin is configured as digital I/O, the BIM will enable the 5-V output driver in the associated pad.

Table 3-50 Port K Supported Pin Function s

Pin	Master	Single Chip	MFTM	SLAM	Emulation[1]
BCLK / PK[7]	BCLK or PK[7]	PK[7]	BCLK or PK[7]	BCLK or PK[7]	BCLK
$\overline{\text{BWE}}$ / PK[6]	BWE or PK[6]	PK[6]	BWE or PK[6]	PK[6]	BWE
$\overline{\text{BOE}}$ / PK[5]	BOE or PK[5]	PK[5]	BOE or PK[5]	PK[5]	BOE
$\overline{\text{BAA}}$ / PK[4]	BAA or PK[4]	PK[4]	BAA or PK[4]	PK[4]	BAA
$\overline{\text{LBA}}$ / PK[3]	LBA or PK[3]	PK[3]	LBA or PK[3]	PK[3]	LBA
$\overline{\text{BREQ}}$ / PK[2]	BREQ or PK[2]	PK[2]	BREQ or PK[2]	$\overline{\text{BREQ}}$ (in)	BREQ
$\overline{\text{BTACK}}$ / PK[1]	BTACK or PK[1]	PK[1]	BTACK or PK[1]	BTACK (out) or PK[1]	BTACK
$\overline{\text{DTACK}}$ / PK[0]	DTACK or PK[0]	PK[0]	DTACK or PK[0]	$\overline{\text{DTACK}}$ (out) or PK[0]	DTACK

NOTES:
1. Digital I/O pin function provided by port replacement unit

3.2.10.1 Port K Pin Assignment Register (PKPAR)

PKPAR — Port K Pin Assignment Register **0xYF FA26**

MSB 7	6	5	4	3	2	1	LSB 0
RESERVED				Burst CS/PK[7:3]	$\overline{\text{BREQ}}$, PK[2]	$\overline{\text{BTACK}}$,/PK[1] PK[1]	$\overline{\text{DTACK}}$/ PK[0]
RESET: See **Table 3-3**.							
0	0	0	0	0	0	0	0

Table 3-51 PKPAR Bit Descriptions

Bit(s)	Name	Description
7:4	—	Reserved
3	Burst CS/PK[7:3]	Port K pin configuration. This bit configures the port K[7:3] pins as either burst chip select signals (BCLK, $\overline{\text{BWE}}$, $\overline{\text{BOE}}$, $\overline{\text{BAA}}$ and $\overline{\text{LBA}}$) or digital I/O. 0 = Port K[7:3] is configured as digital I/O 1 = Port K[7:3] is configured as the burst chip select signals
2:1	—	Port K pin configuration. These bits configure the port K[2:1] pins as either burst mode hand-shake signals $\overline{\text{BREQ}}$ and $\overline{\text{BTACK}}$, or digital I/O. 0 = Port K[2:1] is configured as digital I/O 1 = port K[2:1] is configured as $\overline{\text{BREQ}}$ and $\overline{\text{BTACK}}$
0	$\overline{\text{DTACK}}$/PK[0]	Port K pin configuration. This bit configures the port K[0] pin as either external bus handshake signal $\overline{\text{DTACK}}$ or digital I/O. When this bit is cleared, port K[0] is configured as digital I/O. When set, port K[0] is configured as $\overline{\text{DTACK}}$. 0 = Port K[0] is configured as digital I/O 1 = port K[0] is configured as $\overline{\text{DTACK}}$

3.2.10.2 Port K Data Direction Register (DDRK)

DDRK — Port K Data Direction Register **0xYF FA24**

MSB 7	6	5	4	3	2	1	LSB 0
PK[7]	PK[6]	PK[5]	PK[4]	PK[3]	PK[2]	PK[1]	PK[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-52 DDRK Bit Descriptions

Bit(s)	Name	Description
7:0	PK[7:0]	Port K data direction. The bits in this register control the direction of the port K pin drivers when the pins are configured as I/O. When the port is not defined to be digital I/O, its corresponding data direction bit has no effect. 0 = Corresponding pin is configured as an input 1 = Corresponding pin is configured as an output

3.2.10.3 Port K Output Data Register (PORTK)

PORTK — Port K Output Data Register **0xYF FA20**

MSB 7	6	5	4	3	2	1	LSB 0
PK[7]	PK[6]	PK[5]	PK[4]	PK[3]	PK[2]	PK[1]	PK[0]
RESET:							
0	0	0	0	0	0	0	0

Table 3-53 PORTK Bit Descriptions

Bit(s)	Name	Description
7:0	PK[7:0]	Port K I/O data. PORTK is used to store the data to be driven on the port K output pins, when the port is configured as outputs. When read, the current value of the PORTK register, not the port K pins, is returned.

3.2.10.4 Port K Pin Data Register (PORTKP)

PORTKP — Port K Pin Data Register **0xYF FA22**

MSB 7	6	5	4	3	2	1	LSB 0
BCLK/ PK[7]	$\overline{\text{BWE}}$/ PK[6]	$\overline{\text{BOE}}$/ PH[5:]	$\overline{\text{BAA}}$/ PH[4]	$\overline{\text{LBA}}$/ PH[3]	$\overline{\text{BREQ}}$/PH[2]	$\overline{\text{BTACK}}$/PH[1]	$\overline{\text{DTACK}}$/ PH[0]
RESET: Not affected by RESET.							
0	0	0	0	0	0	0	0

Table 3-54 PORTKP Bit Descriptions

Bit(s)	Name	Description
7:0	—	Port K data input. When read, PORTKP reflects the current state of the port K pins. Writes to PORTKP have no effect.

3.3 Signal Definitions

This section contains a brief description of the input and output signals by their functional groups. Each signal is explained in a brief paragraph with reference (if applicable) to other sections that contain more detail about the function being performed. Specific timing information for each signal can be found in **APPENDIX E ELECTRICAL AND AC CHARACTERISTICS**.

3.3.1 Port A, B, and C Signals

In addition to the functions stated below, the following pins can also be configured to be discrete I/O pins. Refer to **3.2.4 Port A/B Operation** and **3.2.5 Port C Operation** for port control information.

3.3.1.1 Address Bus (A[23:0])

These three-state outputs provide the address for a bus transfer during all currently defined cycles, except CPU-space references. During CPU-space references, the address bus provides CPU related information. The address bus is capable of addressing 16 Mbytes (2^{24}) of data.

3.3.2 Port D Signals

In addition to the functions stated below, the following pins can also be configured to be discrete I/O pins. Refer to **3.2.6 Port D Operation** for port control information.

3.3.2.1 Chip Select ($\overline{CS}$[3:1])

These output signals select external devices at programmed addresses. Refer to **3.6 Asynchronous Chip Select (ACS)** for more information.

3.3.2.2 Freeze (FREEZE)

This output signal indicates that the CPU has acknowledged a breakpoint and has initiated background mode operation.

3.3.2.3 Quotient Out (QUOT)

This output furnishes either the quotient bit of the MSRA or the modulo counter clock. Refer to **3.4.6 Clock Control Registers** for more details on QUOT.

3.3.2.4 Emulation Mode Chip Selects (CSE1)

This output signal, along with CSE0, provides development support hooks. Refer to **3.1.9 Emulator Support** for more details.

3.3.2.5 Breakpoint ($\overline{BKPT}$)

This input signal indicates a hardware breakpoint to the CPU.

3.3.2.6 Development Serial In, Out, Clock (DSI,DSO, DSCLK)

These signals provide serial communications for background debug modes.

3.3.2.7 Instruction Pipe (IPIPE[2:0])

These output signals track the movement of words through the CPU instruction pipeline.

3.3.3 Port E Signals

In addition to the functions stated below, the following pins can also be configured to be discrete I/O pins. Refer to **3.2.7.1 Port E Pin Assignment Register (PEPAR)** for port control information.

3.3.3.1 Function Codes

These three-state outputs identify the processor state and the address space of the current bus cycle, as defined in **Table 3-55**.

Table 3-55 Function Code Assignments

FC2	FC1	FC0	Cycle Type
0	0	0	Reserved (Motorola)
0	0	1	User data space
0	1	0	User program space
0	1	1	Reserved (Motorola)
1	0	0	Reserved (Motorola)
1	0	1	Supervisor data space
1	1	0	Supervisor program space
1	1	1	Supervisor CPU-space operation

3.3.3.2 Chip Selects ($\overline{CS}$[7:5])

These output signals select external devices at programmed addresses. Refer to **3.6 Asynchronous Chip Select (ACS)** for information.

3.3.3.3 System Clock (CLKOUT)

This output signal is the MCU internal system clock. CLKOUT should be used as the bus cycle timing reference by external devices. This clock normally runs at the system clock speed but can be turned off in low power stop mode.

3.3.3.4 Transfer Size (SIZE)

This three-state output signal indicates the number of operand bytes to be transferred on the current bus cycle.

3.3.3.5 Address Strobe ($\overline{AS}$)

This three-state output is driven by the bus master to indicate that valid address, function codes, size, and R/$\overline{W}$ state information is on the bus.

3.3.3.6 Data Strobe ($\overline{DS}$)

In a read cycle, this three-state output signal is driven by the bus master to indicate that an external device should place valid data on the data bus. In a write cycle, data strobe indicates that the master device has placed valid data on the data bus.

3.3.3.7 Read/Write (R/$\overline{W}$)

This three-state output is driven by the bus master to indicate the direction of the data transfer on the bus. A logic one indicates a read from a slave device and a logic zero indicates a write to a slave device.

3.3.4 PORT F SIGNALS

In addition to the functions stated below, the following pins can also be configured to be discrete I/O pins. Refer to **3.2.8 Port F Operation** for port control information.

3.3.4.1 Interrupt Request ($\overline{IRQ}$[6:1], $\overline{IRQX}$)

These three-state inputs are prioritized input lines where $\overline{IRQ7}$ is the highest priority. These signals can be configured as level sensitive or edge sensitive.

$\overline{IRQ}$[6:1] and $\overline{IRQX}$ are internally maskable interrupts. $\overline{IRQX}$ can be configured to any interrupt request level. At reset, $\overline{IRQX}$ is configured as $\overline{IRQ7}$. Refer to **3.2.8.7 Port F Interrupt Level Register (PFLVR)** for more details.

3.3.4.2 Bus Error ($\overline{BERR}$)

This input signal indicates that an invalid bus operation is being attempted.

3.3.4.3 Chip Select ([$\overline{CS4}$])

This output signal selects external devices at programmable addresses. Refer to **3.6 Asynchronous Chip Select (ACS)** for information.

3.3.4.4 PORT G/H SIGNALS

In addition to the functions stated in this section, the following pins can also be configured to be discrete I/O pins. Refer to **3.2.9 Port G/H Operation** for port control information.

3.3.4.5 Data Bus (D[15:0])

These three-state bi-directional signals provide the general purpose data path between the MCU and all other devices.

3.3.5 Port K Signals

In addition to the functions stated below, the following pins can also be configured to be discrete I/O pins. Refer to **3.2.10 Port K Operation** for port control information.

3.3.5.1 Burst Chip Select Control (BCLK, BWE, BOE, BAA, LBA)

These signals implement the burst protocol. Refer to **3.6 Asynchronous Chip Select (ACS)** for more information.

3.3.5.2 Burst Request (BREQ)

This output signal indicates that a burst data transfer is requested by the CPU.

3.3.5.3 Burst Transfer Acknowledge (BTACK)

This input signal indicates that a word burst data transfer is complete. During a read cycle, data is latched when the processor recognizes $\overline{\text{BTACK}}$. Then another burst cycle is run, if needed; during a write cycle, when the processor recognizes $\overline{\text{BTACK}}$, the CPU continues with another burst cycle, if needed.

3.3.5.4 Data Transfer Acknowledge ($\overline{\text{DTACK}}$)

This input signal indicates that a data transfer is complete. During a read cycle, data is unlatched when the processor recognizes $\overline{\text{DTACK}}$ and the bus cycle terminates; during a write cycle the bus cycle is terminated when the processor recognizes $\overline{\text{DTACK}}$.

3.3.6 RESET OUT ($\overline{\text{RSTOUT}}$)

This output signal is an indication that the internal reset controller has reset the chip. When $\overline{\text{RESET}}$ and $\overline{\text{RSTOUT}}$ are both active, the user may drive his override options on the data bus.

3.3.6.1 Emulation Mode Chip Selects (CSE0)

This output signal, along with CSE1, provides development support signals. Refer to **3.1.9 Emulator Support** for more details.

3.3.7 RESET ($\overline{\text{RESET}}$)

This bi-directional open-drain signal is the system reset signal. The signal may be asserted internally or externally. Causes of assertion are power on reset, external reset, system protection reset, CPU reset instruction, test module reset, or loss of clock reset. In all cases, the $\overline{\text{RESET}}$ pin is guaranteed to be asserted by internal logic for a minimum of 512 clock cycles. Refer to **3.4 Clocks** for a complete description.

3.3.8 Crystal Oscillator (EXTAL, XTAL)

These two pins are the connections for an external crystal to the internal oscillator circuit. The EXTAL pin is used as the input in external clock mode. The clock source is configured during reset. Refer to **3.4 Clocks** for more details.

3.3.9 External Filter Capacitor (XFCN, XFCP)

These pins are used to add an external filter circuit for the phase locked loop. Refer to **3.4 Clocks** for more details.

3.3.10 Synthesizer Power (V_{DDSYN})

This pin provides separate power to the frequency synthesizer circuit. It is also used to configure system clock options during reset.

3.3.11 Synthesizer Ground (V_{SSSYN})

This pin is used to provide separate ground to the frequency synthesizer circuit.

3.3.12 External Bus Request ($\overline{EBR}$)

This input signal is used to prevent the MCU from using the external bus.

3.3.12.1 Three-State Control ($\overline{TSC}$)

When $\overline{EBR}$ is asserted simultaneously with the $\overline{RESET}$ pin, the MCU places all the MCU output drivers in a high-impedance state, except $\overline{RSTOUT}$ and $\overline{EBR}$.

3.3.12.2 Floating External Pins

All BIM I/O pins which are configured as inputs, must always be driven by external logic or pulled-up (or down) to avoid unnecessary current drain. The external bus pins, when they are configured for their primary function, will float under certain conditions as summarized in **Table 3-56**. Port G and H pins when configured as data bus pins may also float.

Table 3-56 Driven State of BIM External Bus Pins

Mode	A[23:16], A[15:0], FC[2:0], $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, SIZE	$\overline{DTACK}$, $\overline{BERR}$	D[15:0]
Master mode — $\overline{EBR}$ negated	Driven by BIM	Driven by external logic	Driven by BIM in write cycles. Driven by external device in read cycles. Will float otherwise.
Master mode — $\overline{EBR}$ asserted	Pins not driven by BIM. Should be driven by external master.	Driven by external logic	Pins will float if not driven by external master.
Slave access mode	Driven by external logic	$\overline{DTACK}$ driven by BIM (only if $\overline{AS}$ asserted). $\overline{BERR}$ driven by external logic.	Driven by BIM in read cycles. Driven by external device in write cycles. Will float otherwise.
Single chip	Pin function not supported. Only function available is digital I/O.	Pin function not supported. Only function available is digital I/O.	Pin function not supported. Only function available is digital I/O.
In the cases where the pins will float, the user should drive the pins to either logic 0 or 1, or pull the pins up or down with external resistors for minimum power consumption.			

3.4 Clocks

The clock submodule consists of the crystal oscillator reference (OSC), the phase lock loop (PLL), the reduced frequency divider (RFD), the synthesizer control register

(SYNCR), the synthesizer status register (SYNST), and the clock control logic. shows the overall block diagram for the BIM/IMB clock generation.

To improve noise immunity, the PLL has its own set of power supply pins: V_{DDSYN} and V_{SSSYN}. All other circuits are powered by the normal internal supply pins, which are V_{DDI} and V_{SSI}.

The clock submodule also contains the reset control logic and the reset status register. See **3.4.8 RESET Operation** for details.

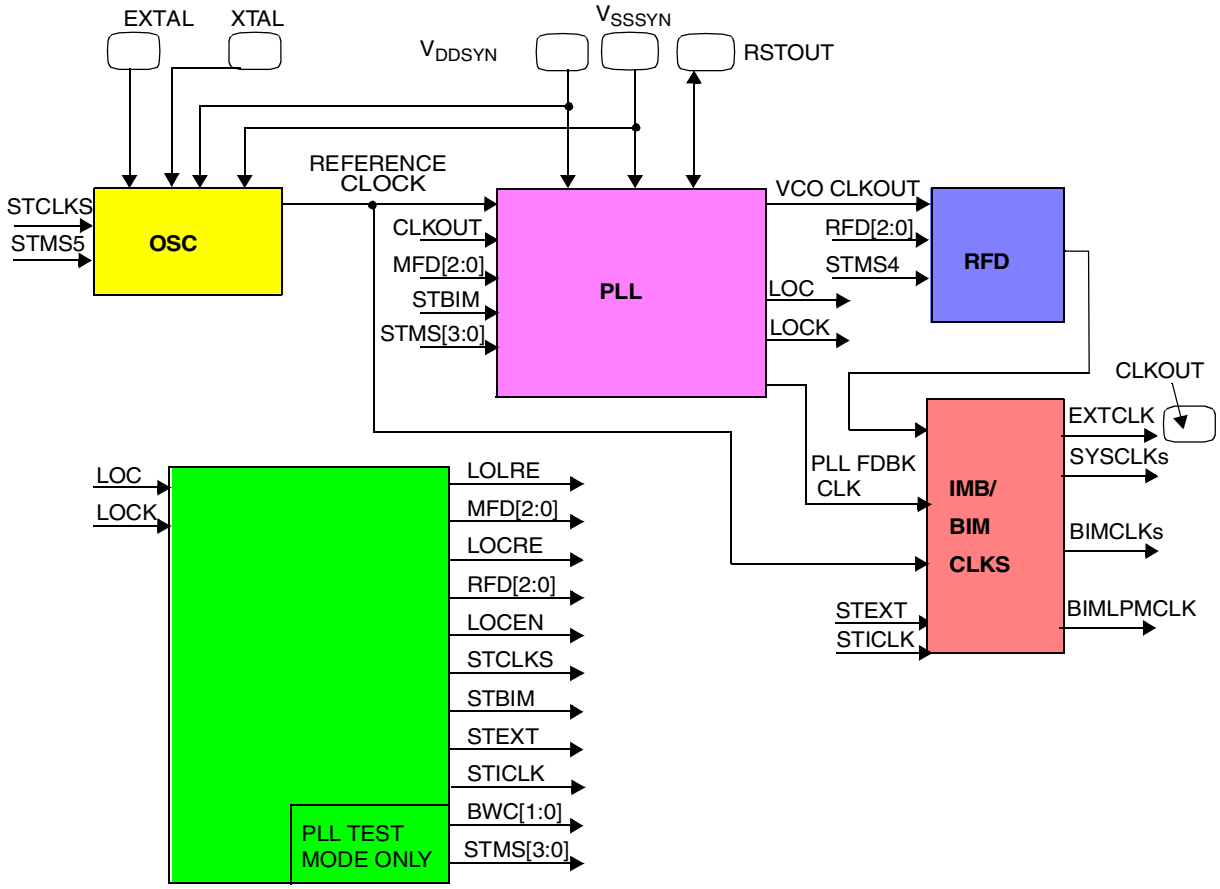


Figure 3-8 BIM Clocks Block Diagram

3.4.1 BIM Clock Modes

As shown in **Table 3-57**, the BIM clock mode is determined during reset by the state of the V_{DDSYN} pin and PCON[4] bit. The $\overline{LBA}$ pin is the override for PCON[4].

The value of V_{DDSYN} is latched during reset, but it is advisable to keep V_{DDSYN} at its reset level after reset is negated. If the V_{DDSYN} is changed during any reset other than power-on reset, the internal clocks may glitch as the clock source is changed between external clock mode and a PLL clock mode. This will affect the accuracy of the system

protection submodule timers. Whenever V_{DDSYN} is changed in reset, an immediate loss of lock condition occurs.

Table 3-57 Clock Source Configuration

V_{DDSYN}	$\overline{LBA}$ (PCON[4])	PLL Options	PLL Enabled
1	1	Normal PLL mode: $Fsys = Fref \cdot \frac{MFD + 2}{2^{RFD}}$	Yes
1	0	1:1 PLL mode: $Fsys = Fref$	Yes
0	X	External clock mode: $Fsys = \frac{Fref}{2}$	No

NOTES:
F_{ref} = input reference frequency
F_{sys} = CLKOUT frequency
MFD ranges from 0 to 7, see **Table 3-58**.
RFD ranges from 0 to 7, see **Table 3-58**.

When the normal PLL clock mode is selected, the PLL is fully programmable. The multiplication factor divider (MFD) provides frequency multiplication of 2X to 9X, and the RFD can be used to reduce the PLL clock frequency without disturbing the PLL.

The MFD and RFD bits have no affect. The external clock is divided by two internally to produce the various system clocks. Refer to **APPENDIX E ELECTRICAL AND AC CHARACTERISTICS** for external clock input requirements.

3.4.2 System Clocks Generation

The BIM uses the reference or the PLL to generate various system clocks. The reference is the crystal oscillator in the normal PLL clock mode. The reference is the external clock in the external and 1:1 PLL clock modes.

System clocks include the EXTCLK, SYSCLKs, BIMCLKs, and the BIMLPMCLK. The clock for the external system, EXTCLK, is provided on the CLKOUT pin when the CLKOUT function is selected by the PEPAR. See **3.2.7.1 Port E Pin Assignment Register (PEPAR)** for more information. SYSCLKs consisting of ICLOCK, ICLOCKE, and ICLK2XE are generated, routed on the IMB3 and are used by all internal IMB3 modules, except the BIM.

NOTE

ICLK2XE operates at two times the ICLOCK frequency. The primary BIM clocks are BIMCLKs. However the system protection submodule's timers, and the reset and port F pin synchronizers are clocked by BIMLPMCLK.

3.4.2.1 Programming System Clocks Frequency

In normal PLL clock mode, the default PLL frequency is two times the reference frequency after reset. The PLL frequency multiples that can be selected in the normal PLL clock mode are shown in **Table 3-58**. The frequency multiple is determined by the RFD and MFD bits in the SYNCR.

When programming the PLL, the system designer must be sure not to violate the maximum system clocks frequency. See **APPENDIX E ELECTRICAL AND AC CHARACTERISTICS** for the maximum allowable frequency The following steps are recommended to accommodate the frequency overshoot that occurs when the MFD bits are changed.

1. Determine the appropriate value for the MFD and RFD fields in the synthesizer control register (SYNCR). Note that the amount of jitter in the system clocks can be minimized by selecting the maximum MFD factor that can be paired with an RFD factor to provide the desired frequency.
2. Write a value of RFD = RFD (from step one) + 1 to the RFD field of the SYNCR.
3. Write the value determined in step one for the MFD field to the SYNCR.
4. Monitor the synthesizer lock bit (LOCK) in the synthesizer status register (SYNST). When the PLL achieves lock, write the RFD value determined in step one to the RFD field of the SYNCR. This changes the system clocks frequency to the desired frequency.

Table 3-58 PLL Frequency Multiples of The Reference — Normal PLL Mode[1]

RFD[2:0]	MFD[2:0] =							
	000 (2X)	001 (3X)	010 (4X)*	011 (5X)	100 (6X)	101 (7X)	110 (8X)	111 (9X)
0 = 000 (÷1)	2	3	4	5	6	7	8	9
1 = 001 (÷2)[2]	1	1.5	2	2.5	3	3.5	4	4.5
2 = 010 (÷4)	0.5	0.75	1	1.25	1.5	1.75	2	2.25
3 = 011 (÷8)	0.25	0.375	0.5	0.625	0.75	0.875	1	1.125
4 = 100 (÷16)	0.125	0.188	0.25	0.313	0.375	0.4385	0.5	0.563
5 = 101 (÷32)	0.0625	0.094	0.125	0.156	0.188	0.218	0.25	0.281
6 = 110 (÷64)	0.031	0.047	0.063	0.078	0.094	0.109	0.125	0.141
7 = 111 (÷128)	0.016	0.023	0.031	0.039	0.047	0.055	0.062	0.070

NOTES:
1. Keep maximum system clock frequency below specified limit given in **APPENDIX E ELECTRICAL AND AC CHARACTERISTICS**.
2. Denotes default value out of reset.

3.4.3 PLL Lock Detection

The lock detect logic monitors the reference frequency and the PLL feedback frequency to determine when frequency lock has been achieved. Phase lock is inferred by the frequency relationship, but is not guaranteed. The PLL lock status is reflected in the LOCK status bit in the SYNST. A sticky lock status indication, LOCKS, is also provided.

The lock detect function utilizes two counters which are clocked by the reference and PLL feedback respectively. When the reference counter has counted N cycles, the feedback counter's count is compared. If the feedback counter has also counted N cycles, the process is repeated for N + K counts. If the two counters' counts still match, the lock criteria is relaxed by 1/2 and the system is alerted that the PLL has achieved frequency lock.

After lock has been detected, the lock circuitry continues to monitor the reference and feedback frequencies using the alternate count and compare process. If the counters do not match at any comparison time, then the LOCK status bit is cleared to indicate that the PLL has lost lock. At this point, the lock criteria is tightened and the lock detect process is repeated.

The alternate count sequences prevent false lock detects due to frequency aliasing while the PLL tries to lock. Alternating between a tight and relaxed lock criteria prevents the lock detect function from randomly toggling between locked and not locked status due to phase sensitivities. **Figure 3-9** illustrates the sequence for detecting locked and not locked conditions.

In external clock mode, the PLL cannot lock since the PLL is disabled.

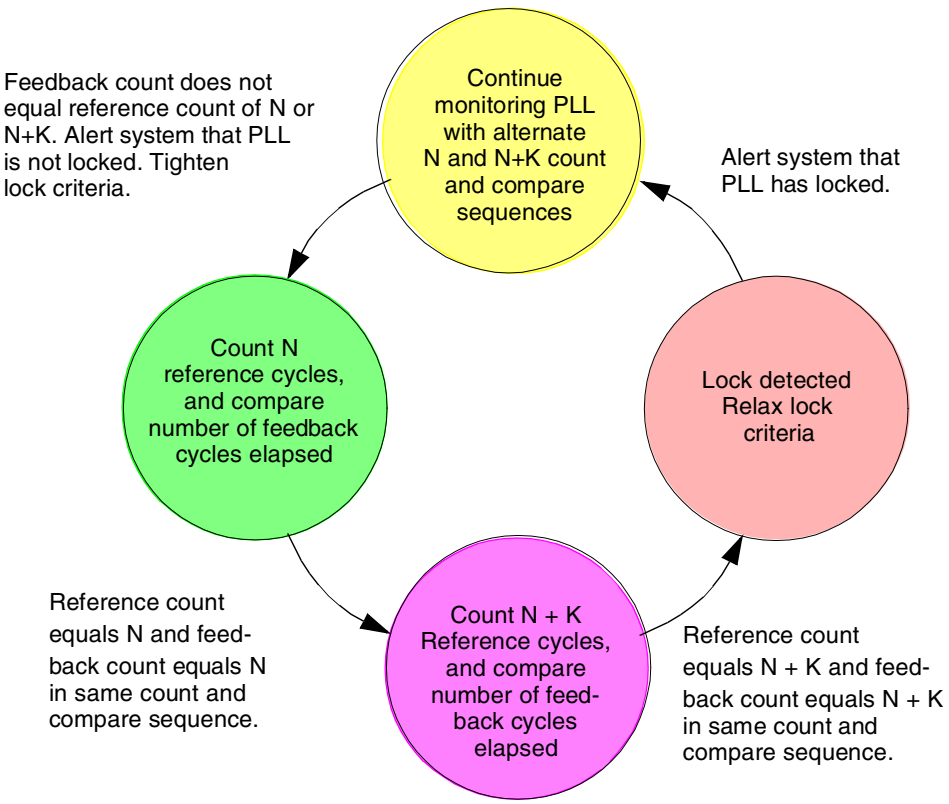


Figure 3-9 Lock Detect Sequence

3.4.3.1 PLL Loss of Lock Conditions

Once the PLL locks after reset, the LOCK and LOCKS status bits are set. If the MFD is changed in normal PLL mode or if an unexpected loss of lock condition occurs in normal or 1:1 PLL clock mode, the LOCK and LOCKS status bits are negated. The system clocks continue to be sourced from the PLL while the PLL attempts to relock.

NOTE

The PLL may overshoot when attempting to relock, causing the system clocks to exceed the maximum system frequency and possibly violating system timing characteristics.

Once locked, the LOCK bit is set. However the LOCKS bit remains cleared if the loss of lock was unexpected. The LOCKS bit is set to one when the loss of lock was caused by changing MFD.

3.4.3.2 PLL Loss of Lock RESET

The BIM provides the ability to assert reset when a loss of lock condition occurs by programming the LOLRE bit in the SYNCR. RESET is asserted if LOLRE is set. Since the LOCK and LOCKS bits in the SYNST are reinitialized after reset, the LOL bit must be read in the RSR to determine that a loss of lock condition occurred.

To exit reset in normal and 1:1 PLL modes, the reference must be present and the PLL must lock.

In external clock mode, the PLL cannot lock. Therefore a loss of lock condition cannot occur, and LOLRE has no affect.

3.4.3.3 Loss of Lock RESET during LPSTOP

The SYNCR can be programmed to disable the PLL during LPSTOP, causing a loss of lock. When the PLL is intentionally disabled in this manner, the loss of lock circuitry does not assert reset upon entering LPSTOP. See **Table 3-61** for when the PLL is disabled in LPSTOP mode.

3.4.4 Loss of Clock Detection

When enabled by the LOCEN bit in the SYNCR, the loss of clock (LOC) detection circuit monitors whether the system clocks are operating in normal or 1:1 PLL modes. When these clocks fail, this logic determines whether the failure is due to a PLL failure or a reference failure. In normal or 1:1 PLL modes, both the crystal oscillator and the PLL are monitored. In external clock mode, the loss of clock circuitry is disabled.

The LOC circuitry monitors both the reference and the PLL operation by monitoring the input clocks to the phase frequency detector shown in **Figure 3-10**. If the reference/feedback clock frequency falls below a minimum frequency, the LOC circuitry considers the clock to have failed and loss of clock status is reported by the LOCS bit in the SYNST. See **APPENDIX E ELECTRICAL AND AC CHARACTERISTICS** for the minimum clock frequency.

3.4.4.1 Alternate Clock Selection

Depending upon which clock source has failed, the LOC circuitry switches the system clocks source to the remaining operational clock. The system clocks are derived from the alternate clock source until reset is asserted. As shown in **Table 3-59**, if the reference fails in normal or 1:1 PLL modes, the PLL goes out of lock and into self-clocked mode (SCM). The PLL remains in SCM until the next reset. If the loss of clock condition is due to a PLL failure, the reference becomes the system clocks source until the next reset.

Table 3-59 Loss Of Clock Summary for Normal Operation

Clock Mode	System Clock Source Before Failure	Clock(s) Monitored by LOC Circuitry When LOCEN = 1	Reference Failure (Alternate Clock Selected by LOC Circuitry until RESET)	PLL Failure (Alternate Clock Selected by LOC Circuitry until RESET)
Normal PLL	PLL	Crystal oscillator, PLL	PLL self-clocked mode	Crystal oscillator
1:1 PLL	PLL	External clock, PLL	PLL self-clocked mode	External clock
External	External clock	None	None	NA

A special loss of clock condition occurs when both the reference and the PLL fail. The failures may be simultaneous or the PLL may fail first. In either case, the reference clock failure takes priority. The PLL attempts to operate in SCM. If successful, the PLL remains in SCM until the next reset. If the PLL cannot operate in SCM, the system remains static until the next reset. Both the reference and the PLL must be functioning properly to exit reset.

If the reference fails in external clock mode, there is no alternate clock to use as the system clocks source. The system remains static until the external clock begins operating again. The LOC condition is not reported, since the loss of clock circuitry is disabled.

3.4.4.2 Loss of Clock RESET

When a loss of clock condition is recognized, reset is asserted if the LOCRE bit in the SYNCR is set. The LOCS bit in the SYNST is cleared after reset, so the LOC bit must be read in the RSR to determine that a loss of clock condition occurred. LOCRE has no affect in external clock mode.

To exit reset in normal and 1:1 PLL modes, the reference must be present and the PLL must lock.

3.4.4.3 Loss of Clock During LPSTOP

SYNCR bits allow the PLL and/or reference clocks to be disabled during LPSTOP. When clocks are intentionally disabled in this manner, a loss of clock condition is not reported. RESET is not asserted upon entering LPSTOP when LOCRE is set. Instead, the loss of clock circuitry is adjusted to only monitor the clock(s) which remain(s) enabled. RESET is asserted if a monitored clock fails during LPSTOP when LOCRE

is set. The BIMCLK and SYSCLK will always match when the part is in LPSTOP except when the pll is disabled. For this case, the BIMCLK operates twice as fast as the SYSCLK. See **Table 3-60** for more information.

Table 3-60 Loss of Clock Monitor Summary During LPSTOP

Clock Mode	STCLKS	STBIM	BIMLPMCLK /CLKOUT Source	Clock(s) Monitored by LOC Circuitry When LOCEN = 1	Reference Failure (Alternate Clock Used:)	PLL Failure (Alternate Clock Used:)
Normal PLL	0	0	Crystal oscillator	Crystal oscillator	PLL self-clocked mode	NA
1:1 PLL			External clock	External clock	PLL self-clocked mode	NA
External			External olock	None	None	NA
Normal PLL	0	1	PLL	Crystal oscillator, PLL	PLL self-clocked mode	Crystal oscillator
1:1 PLL			PLL	External clock, PLL	PLL self-clocked mode	External clock
External			External olock	None	None	NA
Normal PLL	1	x	Disabled	None	NA	NA
1:1 PLL			Disabled	None	NA	NA
External			Disabled	None	NA	NA

So long as the alternate clock remains functioning, LPSTOP can be exited via an interrupt. In normal and 1:1 PLL clock modes, the CPU is released from LPSTOP and execution continues with the alternate clock. Software should monitor LOCS to determine that a loss of clock condition occurred. RESET must be asserted to attempt to relock the PLL.

For external clock mode, the external clock must begin to operate before an interrupt or reset can be recognized to exit LPSTOP mode.

3.4.5 LPSTOP Operation

Low power stop (LPSTOP) mode is entered when the CPU performs a LPSTOP broadcast cycle after executing an LPSTOP instruction. In LPSTOP, some or all system clocks may be disabled to save power.

The STICLK, STCLKS, STBIM, STEXT bits in the SYNCR determine the affect of LPSTOP on the system clocks as follows:

- The BIMCLKs and SYSCLKs can be disabled via STICLK.
- The EXTCLK can be disabled via STEXT
- All system clocks can be disabled via STCLKS.
- The PLL is disabled via STBIM.

These LPSTOP options are summarized in **Table 3-61**.

RESET or any interrupt request that is higher than the interrupt mask priority level will exit LPSTOP mode. During LPSTOP the input synchronizers for interrupts and reset are clocked off the BIMLPMCLK clock, unless all clocks have been disabled. However when both the crystal oscillator and the PLL are disabled, reset operation and interrupt inputs are un-synchronized since no clock is active in the BIM. In this case, the only way to restart the system is by asserting reset or by asserting an external level interrupt request with higher priority than the LPSTOP interrupt priority mask level.

Table 3-61 LPSTOP Summary

STICLK	STCLKS	STBIM	STEXT	SYSCLKs and BIMCLKs Disabled	BIMLPMCLK Disabled	PLL Disabled	EXTOUT Disabled
0	0	0	0	No	No	Yes	No
0	0	0	1	No	No	Yes	Yes
0	0	1	0	No	No	No	No
0	0	1	1	No	No	No	Yes
1	0	0	0	Yes	No	Yes	No
1	0	0	1	Yes	No	Yes	Yes
1	0	1	0	Yes	No	No	No
1	0	1	1	Yes	No	No	Yes
X	1	X	X	Yes	Yes	Yes	Yes

3.4.5.1 Incurred Clock Start-Up

When the PLL is disabled in LPSTOP mode, exiting LPSTOP via an interrupt requires the PLL to re-lock for normal or 1:1 PLL clock modes. System clocks are derived from the reference until the PLL has locked. Then the CPU is released from LPSTOP mode and the system clocks are sourced from the PLL.

NOTE

When EXTCLK is driven during LPSTOP, the source of EXTCLK is switched on-the-fly from the reference to the PLL after it locks.

If the crystal oscillator is disabled during LPSTOP, exiting LPSTOP via an interrupt requires the normal crystal start up time plus PLL lock time for normal or 1:1 PLL clock modes. Until the crystal oscillator starts, the PLL operates in self-clocked mode. When a clock is recognized on the reference input, the PLL starts the locking process. After the PLL locks, the CPU is released from LPSTOP mode and the system clocks are sourced from the PLL.

NOTE

EXTCLK is not driven until the PLL locks.

There is no start-up time when the PLL remains active during LPSTOP.

Whenever LPSTOP is exited via reset, the reference must be present and the PLL must lock before execution can continue in normal and 1:1 PLL modes.

In external clock mode, the reference must be present to exit LPSTOP via reset or via an interrupt.

3.4.6 Clock Control Registers

The clock operation is controlled by the synthesizer control register (SYNCR) and status is reported in the synthesizer status register (SYNST). The following sections describe these registers in detail.

3.4.6.1 Synthesizer Control Register (SYNCR)

The synthesizer control register (SYNCR) is readable and writable in supervisor mode. It contains bits for defining the clock operation for the system. Reserved bits are not affected by writes and always return zero on a read. SYNCR register shows the definition of this register. Following the register definition is an expanded description of each bit field.

SYNCR — Synthesizer Control Register **0xYF FA08**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
LOLRE	MFD[2:0]			LOCRE	RFD[2:0]			STI-CLK	ST-CLKS	STBIM	STEXT	LO-CEN	RESERVED		
RESET:															
1	0	1	0	1	0	0	1	0		0	0	1	0	0	0

Table 3-62 SYNCR Bit Descriptions

Bit(s)	Name	Description
15	LOLRE	Loss of lock reset enable. When the PLL is operating in normal or 1:1 PLL mode, the PLL must be locked before setting the LOLRE bit. Otherwise reset is immediately asserted. The PLL is disabled in external clock mode, so the PLL does not lock. Since LOLRE is intended to handle a loss of lock, the LOLRE bit has no affect in external clock mode. LOCEN does not affect the LOLRE operation. See **3.4.3 PLL Lock Detection** for additional information. 0 = RESET is not asserted when a loss of lock indication occurs 1 = RESET is asserted when a loss of lock indication occurs
14:12	MFD	Multiplication factor divider. The MFD bits control the value of the divider in the PLL feedback loop. The value specified by the MFD bits establish the multiplication factor applied to the reference frequency. The bit field encoding is shown in row one of **Table 3-58**. When the MFD bits are changed, the PLL loses lock. To prevent an immediate reset, the LOLRE bit must be cleared before writing the MFD bits. In 1:1 PLL mode, the MFD bits are ignored and the multiplication factor is set to 1X. In external clock mode the MFD bits have no affect.
11	LOCRE	Loss of clock reset enable. The LOCRE bit determines how the BIM handles a loss of clock condition. If the LOCS bit in the SYNST indicates a loss of clock condition, setting the LOCRE bit causes an immediate reset. Additionally, the LOC bit in the reset status register (RSR) should be checked before setting the LOCRE bit to avoid a hardware/software loop disabling the system. Since reset must be asserted to allow the PLL to relock after entering SCM, both LOCEN and LOCRE are enabled after reset. In external clock mode LOCRE has no affect, since the loss of clock circuitry is disabled. 0 = RESET is not asserted when a loss of lock indication occurs. Has no affect when LOCEN = 0 1 = RESET is asserted when a loss of lock indication occurs

Table 3-62 SYNCR Bit Descriptions (Continued)

Bit(s)	Name	Description
10:8	RFD	Reduced frequency divider. The RFD bits control a prescaler at the output of the PLL. The value specified by the RFD bits establish the divisor applied to the PLL frequency. The RFD bit field encoding is shown in column one in **Table 3-58**. Changing the RFD bits does not affect the PLL's VCO, (i.e., no re-lock delay is incurred.) Resulting changes in clock frequency are synchronized to the next falling edge of the current system clock. However these bits should only be written when the lock bit (LOCK) is set, to avoid surpassing the allowable system operating frequency. In external clock mode the RFD bits have no affect.
7	STICLK	Stop internal clocks. The STICLK bit determines whether the BIMCLKs and SYSCLKs continue to operate in LPSTOP mode. 0 = BIMCLKs and SYSCLKs are enabled 1 = BIMCLKs and SYSCLKs are disabled (normal LPSTOP operation)
6	STCLKS	Stop clocks. The STCLKS bit determines whether any clocks are operational in LPSTOP mode. See **Table 3-61**. 0 = Reference clock continues to operate normally 1 = Reference and PLL are disabled
5	STBIM	Stop BIM clocks. The STBIM bit determines the clock source for BIMLPMCLK in LPSTOP mode. If STCLKS = 1, the STBIM bit has no affect since all clocks are stopped. See **Table 3-61**. In external clock mode, this bit has no affect; the BIMLPMCLK clock source is the external clock. 0 = BIMLPMCLK clock source is the reference and the PLL is disabled 1 = BIMLPMCLK clock source is the PLL
4	STEXT	Stop BIM clocks. The STEXT bit determines whether EXTCLK is driven during LPSTOP mode. The clock source for EXTCLK is determined by STBIM. If STCLKS = 1, STEXT has no affect on EXTCLK since all clocks are disabled. See **Table 3-61**. EXTCLK may also be disabled at the CLKOUT pin by selecting digital I/O as the pin function. Refer to **3.2.7.1 Port E Pin Assignment Register (PEPAR)** for additional information. 0 = EXTCLK is driven 1 = EXTCLK is not driven
3	LOCEN	Loss of clock enable. The LOCEN bit determines whether the loss of clock function is operational. In external clock mode, this bit has no affect; the loss of clock circuitry is disabled. LOCEN does not affect the loss of lock circuitry. When a loss of lock occurs, both LOCK and LOCKS status bits are cleared. See **3.4.4 Loss of Clock Detection** for more information. EXTCLK may also be disabled at the CLKOUT pin by selecting digital I/O as the pin function. Refer to **3.2.7.1 Port E Pin Assignment Register (PEPAR)** for additional information. 0 = Loss of clock function is disabled 1 = Loss of clock function is enabled
2:0	—	Reserved

3.4.6.2 Synthesizer Status Register (SYNST)

The synthesizer status register (SYNST) is an 8-bit read-only register. Reserved bits return zero when read. The synthesizer status register shows the definition of this register. Following the register definition is an expanded description of each bit field.

SYNST — Synthesizer Status Register **0xYF FA0A**

MSB 15	14	13	12	11	10	9	LSB 8
MODE[1:0]		RESERVED			LOCKS	LOCK	LOCS
RESET:							
X		0	0	0	X	X	0

X indicates the bits are decided at reset based on the clock mode.

Table 3-63 SYNST Bit Descriptions

Bit(s)	Name	Description
15	MODE	Clock mode. The MODE bits are determined at reset. 00 = External clock 01 = External clock 10 = 1:1 PLL 11 = Normal PLL
13:11	—	Reserved
10	LOCKS	PLL lock status bit. The LOCKS bit is a sticky indication of PLL lock status. LOCKS is set by the lock detect circuitry when the PLL acquires lock after: 1) a system reset; 2) exiting LPSTOP mode; or 3) a write to the SYNCR which modifies the MFD bits in normal PLL mode. Whenever the PLL loses lock, LOCKS is cleared. LOCKS remains cleared even after the PLL relocks, until one of the three previously-stated conditions occurs. Furthermore, if the LOCKS bit is read when the PLL simultaneously loses lock, the bit does not reflect the current loss of lock condition. If operating in external clock mode, LOCKS remains cleared after reset. In normal and 1:1 PLL modes, LOCKS is set after reset.
9	LOCK	PLL lock status. The LOCK bit indicates whether the PLL has acquired lock. The frequency is specified by the MFD and RFD bits in the SYNCR register for normal PLL clock mode. The frequency is 1X the reference in 1:1: PLL clock mode. The PLL is disabled in external clock mode. The PLL loses lock when a frequency deviation of greater than approximately 1.5% occurs. PLL lock occurs when the synthesized frequency matches to within approximately 0.75% of the desired frequency. If the LOCK bit is read when the PLL simultaneously loses lock, the bit does not reflect the current loss of lock condition. The LOCK bit is used by the power-on-reset circuit as a condition for releasing reset. See **3.3.7 RESET (RESET)** for additional information. If operating in external clock mode, LOCK remains cleared after reset. In normal and 1:1 PLL modes, LOCK is set after reset. 0 = PLL is not yet locked or the PLL has lost lock 1 = PLL is locked
8	LOCS	Sticky loss of clock status. The LOCS bit is a sticky indication of whether a loss of clock condition has occurred at any time since exiting reset in normal and 1:1 PLL modes. If the read of the LOCS bit and the loss of clock condition occur simultaneously, the bit does not reflect the current loss of clock condition. A loss of clock condition can only be detected when LOCEN = 1. See **3.4.4 Loss of Clock Detection**. LOCS is always zero in external clock mode. 0 = System clocks are operating normally 1 = System clocks have failed due to a reference failure or a PLL failure

3.4.7 PLL Operation

In normal PLL mode, the system clocks are synthesized by a PLL that is capable of multiplying the reference clock frequency by 2X to 9X, provided the system clock (CLKOUT) frequency remains within the range listed in **APPENDIX E ELECTRICAL AND AC CHARACTERISTICS**. For example, if the reference frequency is one MHz, the PLL can synthesize frequencies of two MHz to nine MHz. In addition, the output of the PLL can be divided down to reduce the system frequency with the RFD. The RFD is not contained in the feedback loop of the PLL, so changing the RFD bits does not affect PLL operation. See **Table 3-62**.

NOTE

The system frequency programmed should not exceed the maximum system clock frequency allowed, refer to **APPENDIX E ELECTRICAL AND AC CHARACTERISTICS** for the PLL.

Figure 3-10 shows the overall block diagram for the PLL. Each of the major blocks shown is discussed briefly below.

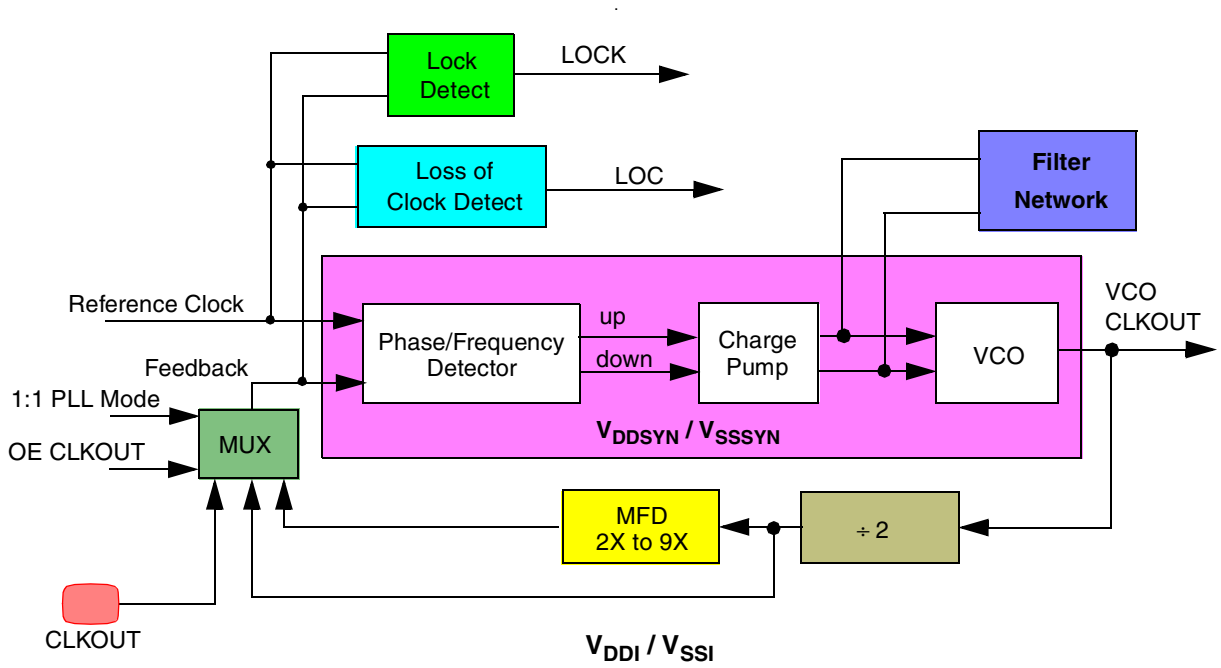
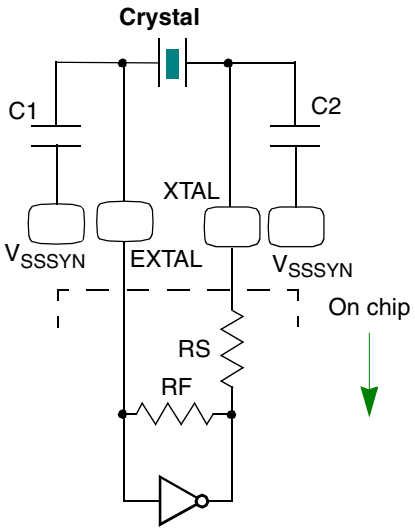


Figure 3-10 PLL Block Diagram Crystal Oscillator

The external support circuitry for the crystal oscillator is shown in **Figure 3-11**. Suggested component values are shown as well.

NOTE

The actual circuit should be reviewed with the crystal manufacturer.



8-MHz Crystal Configuration
C1 = C2 = 16 PF
RF = 1 MW
RS = 470 W

Figure 3-11 Crystal Oscillator

3.4.7.1 Phase/Frequency Detector (PFD)

The PFD is a quad-latch type IV phase-frequency detector. It compares both the phase and frequency of the reference clock and the feedback clock. The reference clock comes from either the crystal oscillator or an external clock source. The feedback clock comes from either CLKOUT in 1:1 PLL mode, two divide by two blocks if CLKOUT is disabled in 1:1 PLL mode, or from the VCO output divided down by the MFD in normal PLL mode.

When the frequency of the feedback clock is less than half the reference clock, the PFD asserts the UP signal continuously. When the feedback frequency is less than the reference clock frequency but greater than half its frequency, than the UP signal will be pulsed for a fraction of each reference clock cycle. When the frequency of the feedback clock equals the frequency of the reference clock (i.e., the PLL is frequency locked), the PFD will pulse the UP or DOWN signals depending on the relative phase of the two clocks. If the falling edge of the feedback clock lags the falling edge of the reference clock, then the UP signal is pulsed. If the falling edge of the feedback clock leads the falling edge of the reference clock, then the DOWN signal is pulsed. The width of these pulses relative to the reference clock is dependent on how much the two clocks lead or lag each other. Once phase lock is achieved, the PFD continues to pulse the UP and DOWN signals for a very short duration during each reference clock cycle. These short pulses force the PLL to continually update and prevent a frequency drift phenomenon referred to as “dead-banding.”

3.4.7.2 Charge Pump/Loop Filter

The charge pump operates in two modes which are automatically selected during the phase locking process. Initially, the charge pump operates in its high current mode (PLL wide bandwidth mode) which enables the PLL frequency to ramp up quickly.

Then as frequency/phase lock is achieved, the charge pump switches to its low current mode (PLL narrow band mode) minimizing the level of systematic jitter in the PLL output clock.

Actual operation of the charge pump is controlled by the UP and DOWN signals from the PFD. They control whether the charge pump applies or removes charge, respectively, from the loop filter.

3.4.7.3 VCO

The voltage formed across the loop filter controls the frequency of the VCO output. The frequency to voltage relationship (or VCO gain) is positive and the output frequency is four times the target system frequency.

3.4.7.4 MFD

The MFD divides down the output of the VCO and feeds it back (when not in 1:1 PLL mode) to the PFD. The PFD controls the VCO frequency (via the charge pump and loop filter) such that the reference and feedback clocks have the same frequency and phase. Thus, the input to the MFD, which is also the output of the VCO, is at a frequency that is the reference frequency multiplied by the same amount that the MFD divides by. For example, if the MFD divides the VCO frequency by six, then the PLL will be frequency locked when the VCO frequency is six times the reference frequency. The presence of the MFD in the loop allows the PLL to perform frequency multiplication, or synthesis. **Table 3-58** shows the possible MF values. When the PLL is operating in 1:1 PLL mode, the MFD is bypassed and the effective multiplication factor is 1X. See **Table 3-57**.

3.4.7.5 Clock Delay

In 1:1 PLL mode, however, the RFD is disabled. The feedback clock comes directly from the CLKOUT pin and true phase lock is achieved. However, if the CLKOUT pin is disabled while operating in 1:1 PLL mode the feedback input for the PLL will be derived from two divide by two blocks. The transition will cause a momentary phase error between the reference and the internal system clock. The reverse situation of enabling CLKOUT while operating in 1:1 PLL mode with CLKOUT disabled will also cause a momentary phase error.

3.4.8 RESET Operation

The BIM has several sources of reset as shown in **Figure 3-12**. The purpose of the reset control logic is to determine the cause of reset, synchronize reset if necessary to the completion of the current bus cycle, and assert the appropriate reset lines. Each reset source has a status bit associated with it and is explained in detail in **3.4.8.13 RESET Status Register (RSR)**.

The pin configuration during reset is described in **3.1.5 RESET Configuration**. To avoid conflicts on the configuration pins, the reset values of the configuration pins should be driven only while the $\overline{\text{RESET}}$ pin and the $\overline{\text{RSTOUT}}$ pin are low. Note that the $\overline{\text{RSTOUT}}$ pin is multi-functional. When the BIM is in emulation mode, $\overline{\text{RSTOUT}}$ is mul-

tiplexed with the CSE0 pin function. While the RESET pin is low, the state of the BIM internal reset signal, SRESET, is driven by the BIM on the RSTOUT pin. Refer to the **APPENDIX E ELECTRICAL AND AC CHARACTERISTICS** for timing information.

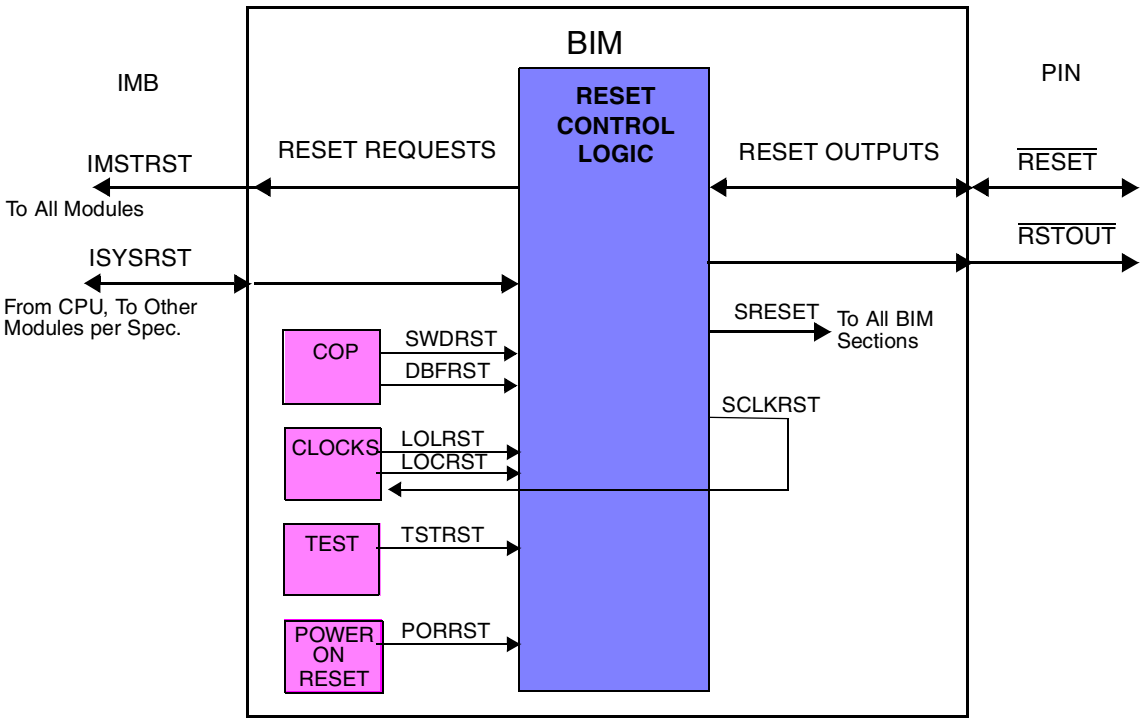


Figure 3-12 RESET Block Diagram

3.4.8.1 Sources of RESET

Table 3-64 defines the sources of reset and the signals driven by the BIM reset controller. External reset, test submodule reset, and loss-of-clock reset are synchronous master reset sources. Asynchronous master reset sources are power-on reset, software watchdog, double bus fault, and loss of lock. The RESET instruction is a system reset source only.

Synchronous master reset sources are not acted upon by the reset control logic until the end of the current bus cycle to protect data integrity. Whenever the reset control logic must synchronize reset to the end of the bus cycle, the internal bus monitor is automatically enabled regardless of the BME bit setting in the SYPCR. Then if the current bus cycle is not terminated normally, the bus monitor eventually terminates the cycle based on the length of time programmed in the BMT field of the system protection control register.

Single-byte or aligned-word writes on the IMB are guaranteed to complete without data corruption when a synchronous reset occurs. External writes are also guaranteed to complete, provided the external configuration logic on the configuration pins is condi-

tioned by the RSTOUT pin being low. A long word write, a misaligned operand write, or a read cycle are not guaranteed to complete.

Asynchronous master reset sources usually indicate a catastrophic failure. The reset control logic does not wait for the current bus cycle to complete. RESET is asserted immediately to the system. **Table 3-64** gives a summary of the action that occurs for each reset source.

Table 3-64 RESET Source Summary

Source	Timing	RESET Lines Asserted by RESET Controller				
External RESET pin	synch	IMSTRST[1]	SRESET[2]	SCLKRST[3]	SMCRSTB[4]	ISYSRST[5]
Power-on	async	IMSTRST	SRESET	SCLKRST	SMCRSTB	—
Sys prot WDOG	async	IMSTRST	SRESET	SCLKRST	SMCRSTB	ISYSRST
Sys prot DBF	async	IMSTRST	SRESET	SCLKRST	SMCRSTB	ISYSRST
Loss of clock	synch	IMSTRST	SRESET	SCLKRST	SMCRSTB	ISYSRST
Loss of lock	async	IMSTRST	SRESET	SCLKRST	SMCRSTB	ISYSRST
Test	synch	IMSTRST	SRESET	—	SMCRSTB	ISYSRST
CPU RESET instruction	synch	—	—	—	SMCRSTB	—

NOTES:
1. IMSTRSTB is the master reset line that goes to all modules on the IMB.
2. SRESET is the reset line that goes to all other internal circuits in the BIM.
3. SCLKRST is the reset line that resets the clock sub-module.
4. SMCRSTB is the reset line that drives the external reset pin.
5. ISYSRSTB is the system reset line that goes to all modules on the IMB. The reset controller asserts ISYSRSTB for one clock tic before asserting IMSTRSTB, except for POR.

3.4.8.2 External RESET

Since the RESET pin is a bi-directional line, a conflict exists when the CPU executes a RESET instruction and when an external device asserts the RESET pin. To guarantee that an external reset will be recognized by the EBI, the RESET pin must be held for at least 750 CLKOUT cycles so that it overlaps the 512 cycles of an internal reset.

If the CPU RESET instruction is not used, an external RESET assertion as short as four cycles will be recognized and latched by the BIM. The BIM then asserts RESET for approximately 512 cycles.

3.4.8.3 Power-On RESET

Upon power up, the reset controller asserts RESET and maintains the assertion of RESET for approximately 512 cycles after V_{DD} has reached a minimum acceptable level. Refer to **APPENDIX E ELECTRICAL AND AC CHARACTERISTICS**. Additionally, when 1:1 PLL clock mode or normal PLL clock mode is selected, RESET remains asserted until the PLL achieves phase-lock. The power on reset signal is now driven down the IMB.

3.4.8.4 Software Watchdog RESET

This reset condition occurs when the IRSEL bit in the SYPCR is cleared and the software watchdog counts down to zero. The reset controller asserts RESET for approximately 512 cycles, and then exits reset and resumes operation.

3.4.8.5 Double Bus Fault RESET

This reset condition occurs when the DBE bit in the SYPCR is set and HALT is asserted after a double bus fault. The reset controller asserts RESET for approximately 512 cycles, and then exits reset and resumes operation.

3.4.8.6 Loss of Clock RESET

This reset condition occurs when the reference or PLL fails in 1:1 or normal PLL mode and the LOCRE bit in the SYNCR register is set. The reset controller asserts RESET for approximately 512 cycles, and then exits reset and resumes operation.

3.4.8.7 Loss of Lock RESET

This reset condition occurs when the PLL loses lock and the LOLRE bit in the SYNCR register is set. The reset controller holds RESET for approximately 512 cycles, and then exits reset and resumes operation.

3.4.8.8 Test RESET

This reset condition occurs when the BIM is in test mode and the test module requests this reset as a result of its completion of an automatic scan operation. The reset controller asserts RESET for approximately 512 cycles, and then exits reset and resumes operation.

3.4.8.9 System RESET

A system reset occurs when the CPU executes a RESET instruction. The RESET instruction allows software to reset the system to a known state and then continue processing with the next instruction. The RESET instruction does not cause loading of the reset vector or affect any internal CPU registers or BIM configuration registers. The RESET instruction does assert the external RESET pin to reset external devices. The CPU drives ISYSRST, which may be used by modules other than the BIM. It is the responsibility of the CPU to assert reset at the proper time in the bus cycle, and to maintain reset in the asserted state for approximately 512 cycles (C32 X 256 cycles). Not all CPUs support driving ISYSRST.

The reset controller uses the ISYSRST input only as an indication to drive the SMCRSTB signal. RSTOUT will not be asserted for a system reset since the SRESET signal is not asserted, (i.e., the BIM is not reset).

After ISYSRST is negated by the CPU, the reset controller waits approximately 10 clocks for the external pullup on the RESET pin to negate the pin, and then samples the pin. If the RESET pin is high, the reset controller releases the MCU to continue bus cycles. If the pin is still low, it waits approximately 180 further clocks and samples the

pin again. If the RESET pin is negated at this point, the EBI is released to run bus cycles. If the pin is still low, an external reset sequence is initiated.

3.4.8.10 RESET Assertion by an External Device

If the RESET pin is driven low by an external device for at least four CLKOUT cycles, the reset control logic will latch the reset request internally, but the MCU is not yet in reset (13). When the current bus cycle is completed, SCLKRST and IMSTRSTB are asserted. The MCU is now in reset(19). When the reset control logic detects that the RESET pin is not being driven low externally(10), the reset control logic will then drive the RESET pin low for an additional 512 cycles to guarantee that a minimum 512 cycle reset is generated to the entire system(3). At the end of 512 cycles, the reset control logic releases the RESET pin (high impedance)(5), and the external configuration on the appropriate pins is latched(4) (see **3.1.5 RESET Configuration**).

Now the reset control logic begins waiting for 10 additional clock cycles, to allow the external pullup device on RESET to negate the pin(5). At the end of this 10 cycle period, the reset logic samples the RESET pin(6). If the RESET pin is high, the control logic releases the EBI to begin running bus cycles(9). If the RESET pin is still asserted low, the control logic waits for 180 further clock cycles and samples the pin again(7). If the pin is high, the control logic releases the MCU to begin running bus cycles(9). If RESET is still low, the reset control logic will wait for the pin to go high(10), and then drive the RESET pin low again for 512 cycles(3). At the end of this 512 clock period, the reset configuration is re-latched(4), and the pin-sampling sequence is repeated.

3.4.8.11 Internal RESET Request

If reset is asserted due to a synchronous internal reset source, such as test module reset(14), the reset control logic will wait for a bus cycle termination(18), and then drive internal and external reset for 512 system clock cycles(18,3). If reset is asserted by an asynchronous reset source, such as double bus fault or software watchdog(11), the reset control logic drives internal and external reset for 512 system clock cycles without waiting for a bus cycle termination(3). Thus, any internal source of reset is guaranteed to cause at least a 512 cycle assertion of reset. After the 512 cycles, the reset control logic will re-latch the configuration and continue the previously described sequence.

3.4.8.12 Power-On RESET

When the reset sequence is initiated by power-on reset(1), the same reset sequence is followed as other asynchronous reset sources.

3.4.8.13 RESET Status Register (RSR)

The reset status register (RSR) contains a status bit for every reset source in the BIM. When the BIM enters reset, the cause(s) of the reset condition is/are latched along with a value of zero for the other reset sources that did not cause the reset condition. These values are then reported via the RSR; one or more status bits may be set. The cause of any subsequent reset is also recorded in the register, overwriting status from

the previous reset condition. If a system reset occurs simultaneously with any other type of reset, the RSR only records the master reset source.

This register can be read at any time. This register cannot be written.

RSR — RESET Status Register **0xYF FA0A**

MSB 7	6	5	4	3	2	1	LSB 0
EXT	POW	SW	DBF	LOL	LOC	SYS	—
RESET:							
0	0	0	0	0	0	0	0

Table 3-65 RSR Bit Descriptions

Bit(s)	Name	Description
7	EXT	External reset. This bit indicates that the last reset state was caused by an external device asserting the RESET pin.
6	POW	Power up reset. This bit indicates that the last reset state was caused by the power-up reset circuit, which is part of the reset controller.
5	SW	Software watchdog reset. This bit indicates that the last reset state was caused by the software watchdog circuit in the system protection module.
4	DBF	Double bus fault. This bit indicates that the last reset state was caused by a double bus fault as detected by the system protection module.
3	LOL	Loss of lock reset. This bit indicates that the last reset state was caused by a loss of lock as detected by the PLL circuits.
2	LOC	Loss of clock reset. This bit indicates that the last reset state was caused by a loss of clock as detected by the loss of clock circuitry.
1	SYS	System reset. This bit indicates that the last reset state was caused by the CPU executing a RESET instruction.
0	—	Reserved

3.5 System Protection

The burst integration module (BIM) provides features to safeguard MCU operations within different operating environments. Many functions normally implemented externally, whether by glue logic, or in software, have been integrated on-chip. The BIM includes bus cycle timeout monitors, a real-time clock which can operate in low power mode, and a software watchdog timer for runaway software protection. The three monitors in the system protection sub-module are: the bus monitor, the spurious interrupt monitor and the double bus fault monitor.

Details on the programming model for system protection registers can be found in **3.5.5 Programming Model**.

Figure 3-13 represents a graphical overview of the system protection section.

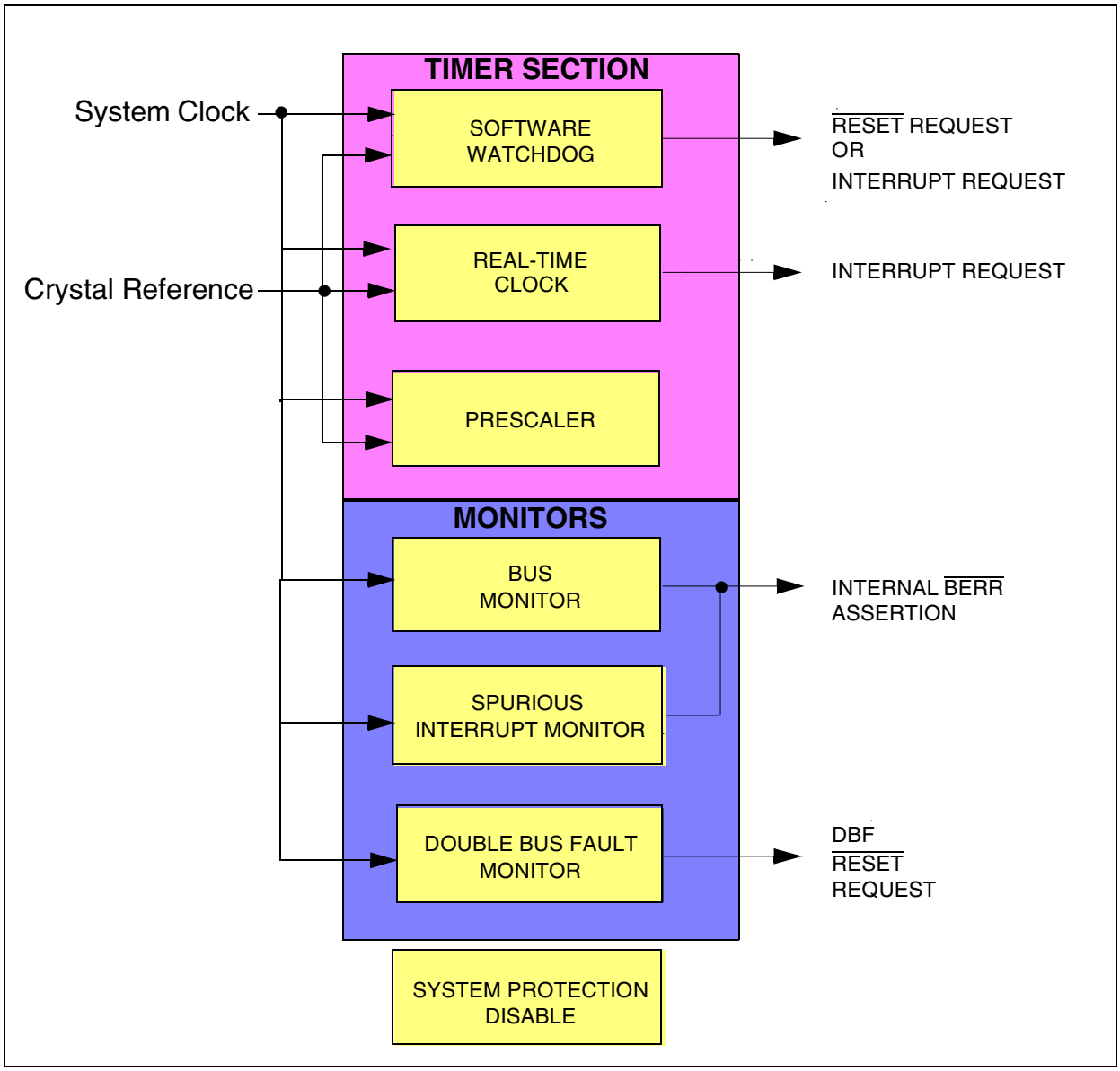


Figure 3-13 System Protection Block Diagram

3.5.1 Timer Section Functions

This section is an explanation of the features represented in **Figure 3-13**. The timer section includes the software watchdog, the real-time clock and the prescaler.

3.5.1.1 Software Watchdog Interval Timer (SWDOG)

The software watchdog monitors system software by requiring the software to periodically reload the software watchdog interval timer to prevent a time-out alarm from being issued and to provide a controlled exit from runaway software loops. The software watchdog is prevented from issuing a time-out alarm by requiring application

software to execute a special programming sequence to reload the SWDOG interval timer on a regular basis to inhibit time-out. If this real-time service does not take place, the software watchdog will issue either a reset or an interrupt request.

Features of the software watchdog include:

- Ability to select between an interrupt request or a reset if a time-out occurs.
- The input clock to the software watchdog interval timer may be the system clock, the crystal frequency, or one of four prescaler taps.
- User-specified time-out intervals which range from one µs to 67 seconds (with a 2-MHz crystal reference frequency input) or from 60.6 ns to 4.1 seconds (with a 33-MHz system clock input). See **Table 3-76** for a complete listing of time-out interval ranges for the software watchdog (SWDOG) at selected frequencies.
- The software watchdog interval timer is readable at any time.
- The software watchdog and the real-time interval timers can be configured to create a 32-bit interval timer for extended time-out.

3.5.1.2 Real-Time Clock (RTC)

The real-time clock provides a periodic interrupt timer (PIT). The features of the real-time clock include:

- Periodic interrupt capability if programmed to generate an interrupt request at time-out.
- The input clock to the real-time interval timer may be the system clock, the crystal frequency, or one of four prescaler taps.
- User-specified time-out intervals which range from 1 µs to 67 seconds (with a 2-MHz crystal reference frequency input) or from 60.6 ns to 4.06 seconds (with a 33-MHz system clock input). See **Table 3-76** for a complete listing of time-out interval ranges for the real-time clock at selected frequencies.
- The real-time clock interval timer is readable at any time.
- The RTC timer is user programmable to load a new interval value in one of two ways: load a new time interval immediately after a write to the interval register or load the new interval value after current interval completes.

3.5.2 SWDOG and RTC Configured as 32-Bit Interval Timer

The real-time clock and the SWDOG may be chained together to form a 32-bit interval timer. The most significant word of the timer is the software watchdog, and the least significant word of the timer is the real-time clock interval timer. In this configuration, the SWDOG interval timer acts as a 16-bit extension to the real-time clock interval timer and normal software watchdog functionality is disabled. See **Figure 3-19**.

The 32-bit interval timer features include:

- Exceptionally long time-out intervals by chaining the two 16-bit interval timers.
- The interval timers are readable at any time.
- The interval timers are unaffected by master reset so that timer configuration is insensitive to resets while the MCU is operating.
- A selectable clock source: system clock, crystal reference frequency or one of

four prescaler taps.
- In chained mode, the 32-bit timer can only assert an interrupt request as a result of time-out.
- The 32-bit interval timer can be user programmed in one of two ways: load a new time interval immediately after write to interval register or load register after current interval completes.

3.5.2.1 Prescaler

The SWDOG and the RTC both share a 10-bit synchronous timer prescaler.

The system protection prescaler features include:

- Clock source which is selectable between the PLL reference crystal frequency and the system clock.
- Four prescaler taps which add to the clock choices that can be applied to the SWDOG or the RTC interval timers.
- The prescaler may be read at any time.
- The prescaler is reset to zero under software control so that a complete interval time-out value will occur when the SWDOG and/or RTC interval timers begin counting.

3.5.3 Monitor Functions

This section documents the functionality of the bus monitor, the double bus fault monitor, and the spurious interrupt monitor.

3.5.3.1 Bus Cycle Timeout Monitor

If a bus cycle does not terminate with the assertion of DTACK or BTACK, the bus monitor will assert an internal BERR to terminate the bus cycle. The bus monitor may monitor internal and/or external bus accesses. Four selectable time-out intervals range from eight system clock cycles to 64 system clock cycles.

Typical asynchronous bus systems require the implementation of some kind of watchdog to monitor excessive DTACK or BTACK signal response times during a bus access. If an unimplemented address is accessed or a peripheral is inoperative, the DTACK or BTACK will not be issued and the bus will be locked up while it waits for the required response.

The BIM provides a bus monitor to monitor all internal bus accesses and an option to monitor any internal to external bus accesses. The bus monitor watches the DTACK response time during a normal bus cycle (or BTACK during a burst cycle) or watches for the auto vector (AVEC) signal during an interrupt acknowledge (IACK) bus cycle. If the response time of the internal modules or external peripherals exceeds the programmed count, the bus monitor will assert an internal bus error (BERR). The bus monitor does not watch for DTACK or BTACK response on the external bus unless the MCU initiates the bus cycle.

The bus monitor monitors the DTACK, BTACK or AVEC response time (in clock cycles) by using a progammable maximum allowable response interval.

Since the bus monitor is concerned only with internally initiated bus cycles, it will be disabled when the external bus request (EBR) pin is asserted.

There are four selectable response time intervals for the bus monitor ranging from eight system clock cycles (~142 ns @ 33 MHZ) to 64 system clock cycles (~1.94 µs @ 33 MHz). The intervals are selectable with the BMT field in the system protection control register (see **Table 3-67**). The programmability of the time-out allows for varying external peripheral response times. The timing mechanism is derived from taps off a six stage divider chain clocked by the system clock. The taps are taken at the "÷8", "÷16", "÷32", and "÷64" stages. The divider chain is cleared and started on all internal cycles, and optionally on internal to external cycles, and is then clocked by the system clock. If the cycle is not terminated by DTACK or AVEC within the selected response time, or BTACK response is too slow, a time-out will occur. When a time-out occurs, the bus monitor asserts internal BERR to terminate the bus cycle.

The BME bit in the system protection control register (SYPCR), when set, will alert the internal bus monitor to watch DTACK and BTACK response time during an internal to external bus cycle. Refer to **Table 3-67**.

3.5.3.2 Double Bus Fault Monitor (DBF)

The double bus fault monitor asserts reset if a double bus fault occurs. The DBF monitor monitors the HALT line of the intermodule bus which some CPUs assert after a double bus fault occurs.

A double bus fault occurs if internal BERR is asserted by the monitor during the process of servicing particular exceptions which are specific to the CPU ***(see appropriate CPU Manual)***. The CPU is halted and all processes cease. The halted state is the double bus fault state. Entry into this state (CPU initiated HALT) triggers the double bus fault monitor which causes a master reset. A flag is set in the BIM module's reset status register (RSR) when reset is caused by a double bus fault. The operation of this function may be inhibited by the DBE bit in the system protection control register (refer to **Table 3-67**).

3.5.3.3 Spurious Interrupt Monitor (SPM)

The internal BERR signal is asserted if no interrupt arbitration takes place during an interrupt acknowledge (IACK) bus cycle (spurious interrupt request). This feature is always operating and cannot be disabled.

The internal spurious interrupt monitor will assert internal BERR if no activity is detected on the internal bus lines IARB0-IARB1 during an IACK cycle. During an IACK cycle, one or more internal modules will recognize the interrupt request level generated by the CPU and will arbitrate for the bus. Arbitration is carried out using the two lines IARB0 and IARB1 on the internal bus if neither of those lines are asserted during an IACK cycle, a spurious interrupt request has been encountered.

This feature can never be inhibited; it is always enabled.

3.5.4 Disabling The System Protection Sub-Module

For power savings or development purposes, the system protection sub-module may have all or part of its subsections disabled as described in the following sections.

3.5.4.1 SYSPROT Disable Bit

The clocks to the entire system protection sub-module may be shut down at any time by writing a '1' to the SYSPROT bit in the module disable register (MDR).

3.5.4.2 Timer Section Disable

The timer section may be disabled at any time by asserting the internal FREEZE line while the BIM MCR FRZ[0] bit is a '1'. The clock to the software watchdog, the RTC and prescaler will be disabled on the low phase of the system clock. It will remain disabled until the next rising edge of the system clock after the FREEZE line is negated.

3.5.4.3 Bus Monitor Disable

When the internal FREEZE line is asserted and the BIM MCR FRZ[1] bit is a '1', the clock to the bus monitor will stop and therefore disable BERR from occurring if DTACK or BTACK is later than the programmed bus monitor time-out interval. The clock will be disabled on the low phase and will remain disabled until the next rising edge of the system clock after the FREEZE line is negated.

3.5.5 Programming Model

Table 3-66 shows the address map for this BIM submodule. These registers provide status information, and configuration and operation control.

Table 3-66 System Protection Submodule Programming Model

Address	15 8	7 0
0xYF FA50	SYSTEM PROTECTION CONTROL REGISTER (SYPCR)	
0xYF FA52	TIMER CONTROL REGISTER (TIC)	TIMER INTERRUPT VECTOR REGISTER (TIV)
0xYF FA54	RESERVED	SOFTWARE WATCHDOG SERVICE REGISTER (SWS)
0xYF FA56	PRESCALER (PRE) [READ-ONLY]	
0xYF FA58	SOFTWARE WATCHDOG INTERVAL REGISTGER (SWI)	
0xYF FA5A	REAL-TIME CLOCK INTERVAL REGISTER (RTI)	
0xYF FA5C	SOFTWARE WATCHDOG INTERVAL TIMER (SWIT) (READ-ONLY)	
0xYF FA5E	REAL-TIME INTERVAL COUNTER (RTIT) (READ-ONLY)	

3.5.6 System Protection Registers

The system protection registers consist of:

- System protection control register (SYPCR),
- Timer control register (TIC),
- Timer interrupt vector register (TIV),
- Software watchdog service register (SWS).

3.5.6.1 System Protection Control Register (SYPCR) (V4 and V5)

SYPCR — System Protection Control Register **0xYF FA50**

MSB 15	14	13	12	11	10	9	LSB 8
REN	SREN	PCLK	SLPC	RZ	SZ	IRSEL	TIEN
RESET:							
0	0	0	0	0	0	0	0

MSB 7	6	5	4	3	2	1	0
LCI[1]	TIQL[2]	TIQL[1]	TIQL[0]	DBE	BME	BMT[1]	BMT[0]
RESET:							
1	0	0	0	0	0	0	0

NOTES:
1. The LCI bit is enabled only on V5.

Bits [15:7] of the system protection control register are reset or asserted by power-on reset and bits [6:0] are reset or asserted by master reset.

NOTE

When FREEZE is asserted, bits [3:0] are writable at any time. Otherwise, these bits may be written only once after a power-on reset or master reset.

Table 3-67 SYPCR Bit Descriptions

Bit(s)	Name	Description
15	REN	RESET enable. This bit controls whether writes to the real-time interval register will reset the prescaler (see **3.5.9.1 Real-Time Interval Register (RTI)**). This control bit allows the user to control whether the Prescaler is started over at zero upon writing a new value to the real-time interval (RTI) register. Master reset will not affect this bit. 0 = Prescaler reset disabled 1 = Prescaler reset enabled
14	SREN	SWDOG reset enable. This bit controls whether writes to the Software Watchdog Interval (SWI) register will reset the prescaler. See **3.5.6.4 SWDOG Service Register (SWS)**. This control bit allows the user to control whether the prescaler is started over at zero upon writing a new value to the software watchdog interval (SWI) register. Master reset does not affect this bit. 0 = Prescaler reset disabled 1 = Prescaler reset enabled

Table 3-67 SYPCR Bit Descriptions (Continued)

Bit(s)	Name	Description
13	PCLK	Prescaler clock select. The PCLK bit selects the clock source to the prescaler. Master reset will not affect this bit. A clock source may be selected whether or not that source is active, (i.e., the PLL or crystal is turned off). In this case, the system protection prescaler will not run since it will have selected a channel which is not carrying a clock signal. If the SWDOG and the RTC interval timers are using the system protection prescaler taps, or the crystal frequency, they also will not run. **WARNING**: Because of the asynchronous relationship between the system clock and crystal reference clock inputs to the prescaler, switching these clock inputs while the prescaler is running (SWIT or RTIT being clocked from a prescaler tap) may cause erroneous counts. 0 = Selects system clock divided by 2 1 = Selects crystal reference frequency divided by 2
12	SLPC	Software watchdog LPSTOP count bit. This bit controls whether the software watchdog stops counting after entering LPSTOP mode. This bit is unaffected by master reset. 0 = SWDOG does not count in LPSTOP; counting will resume when exit from LPSTOP 1 = SWDOG counts in LPSTOP
11:10	RZ, SZ	Timer zero flags. The timer zero flags function as the interrupt pending bits for the interval timers. By reading the flag as a '1' and then writing the flag to a zero, the flag is cleared. When the flag is cleared, the pending interrupt request is cleared. Should another time-out event occur after the user has read the flag as a '1' and before the user writes a '0' to it, the flag will remain a '1' regardless of the attempt to clear it. This feature provides limited double buffering of time-out event. These bits are unaffected by master reset. RZ is asserted after the real-time interval timer reaches zero; SZ is asserted after the software watchdog interval timer reaches zero.
9	IRSEL	Interrupt/reset select bit. The IRSEL bit selects whether the software watchdog interval timer (SWIT) issues an interrupt request or a master reset when the SWDOG interval timer times out. This bit is unaffected by master reset. 0 = SWIT issues reset 1 = SWIT issues the interrupt request level programmed in the TIQL field of the system protection control register
8	TIEN	Timer interrupt enable. The TIEN bit controls whether the real-time interval timer (RTIT) issues an interrupt request after timing-out. The interrupt request priority is specified in bits [6:4] of the system protection control register. This bit is unaffected by master reset. 0 = No interrupt request is issued upon RTIT reaching zero 1 = Interrupt request will be issued upon RTIT reaching zero
7	LCI	Load count immediate. The LCI bit controls when a change in real-time interval register (or real-time interval register and software watchdog interval register in 32-bit chained mode) is loaded into the real-time interval timer (or real-time interval timer and SWDOG interval timer in 32-bit chained mode). This bit is unaffected by master reset. The LCI bit is enabled only on V5. 0 = Count is allowed to reach zero (time-out) before the new value is loaded 1 = new value is loaded immediately after the writing a new value to the RTI register
6:4	TIQL	Timer interrupt request level field. Bits [6:4] of the system protection control register define the priority of interrupt request that is generated when the real-time interval timer or the software watchdog interval timer time-out. A value of '000' disables interrupt requests, but does not disable the interval timers from running. After master reset, the timer interrupt request level field defaults to *interrupt disabled*. 000 = Disable interrupt requests 001 = Interrupt request level 1 010 = Interrupt request level 2 011 = Interrupt request level 3 100 = Interrupt request level 4 101 = Interrupt request level 5 110 = Interrupt request level 6 111 = Interrupt request level 7

Table 3-67 SYPCR Bit Descriptions (Continued)

Bit(s)	Name	Description
3	DBE	Double bus fault monitor enable. This bit controls the monitoring for double bus fault on the inter-module bus (IMB) by the double bus fault monitor (see **3.5.3.2 Double Bus Fault Monitor (DBF)** for a complete explanation of a double bus fault). When test mode is enabled or FREEZE is asserted, this bit is writable at any time. Otherwise, this bit may be written only once after a power-on reset or master reset, after which no more writes will be allowed. 0 = Disable double bus fault monitor 1 = Enable double bus fault monitor
2	BME	Bus monitor external enable. This bit controls whether the internal bus monitor will operate during an external bus cycle. When enabled, the bus monitor monitors DTACK or BTACK response time for all bus cycles. When test mode is enabled or FREEZE is asserted, this bit is writable at any time. Otherwise, this bit may be written only once after a power-on reset or master reset. 0 = Disable bus monitor function for external bus cycle 1 = Enable bus monitor function for external bus cycle
1:0	BMT	Bus monitor timing. These bits select the time-out interval, in system clocks, for the bus monitor. After master reset, the bus monitor time-out value defaults to *64 system clocks*. For external accesses, this time-out value should be longer than the chip select access time. When FREEZE is asserted, these bits are writable at any time. Otherwise, these bits may be written only once after a power-on reset or master reset. 00 = 64 system clocks 01 = 32 system clocks 10 = 16 system clocks 11 = 8 system clocks

3.5.6.2 Timer Control Register (TIC)

TIC — Timer Control Register **0xYF FA52**

MSB 7	6	5	4	3	2	1	0
PRR	SWC[2]	SWC[1]	SWC[0]	D2R	RTC[2]	RTC[1]	RTC[0]
RESET:							
0	0	0	0	0	0	0	0

The timer control register contains the prescaler reset bit (PRR), the real-time clock control field (RTC[2:0]), the divide-by-2 reset bit (D2R), and the software watchdog clock control field (SWC[2:0]). The SWC and RTC fields select the clock input to the software watchdog interval timer and the real-time interval timer, respectively. Master reset will have no effect on this register. **Figure 5-2** shows the prescaler block diagram. When the user selects the 32-bit chained mode, normal software watchdog functionality is disabled. For more details, see **3.5.9.9 RTC and SWDOG in 32-Bit Mode**.

The system clock and the crystal reference settings allow clocking of the SWDOG or the RTC interval timers by the system clock divided by two, or the crystal reference divided by two, respectively. The prescaler settings, on the other hand, allow one of four prescaler taps to clock the interval timers. When neither the RTC nor SWDOG select a prescaler tap, the prescaler is disabled from counting.

Asserting reset to exit LPSTOP has no affect on either the SWC or the RTC fields. Thus, the SWDOG and the RTC can each continue to run with the same clock configuration in and out of LPSTOP. These bits are unaffected by master reset.

Table 3-68 TIC Bit Descriptions

Bit(s)	Name	Description
7	PRR	Prescaler reset. This bit resets the prescaler when it is written to a '1'. This capability is provided so that the prescaler may be reset at the moment the user is changing the configuration of the timer control register. The user may not wish to reset the prescaler in cases where the SWDOG and the RTC are both using prescaler taps as clock inputs. A prescaler reset for the sake of one interval timer would affect the count in the other. In this case, a '1' should NOT be written to the PRR bit. The PRR bit always reads '0'.
6:4	SWC[2:0]	SWDOG clock control. Provides seven possible settings for choosing the input clock for the software watchdog interval timer (SWIT). 000 = System clock/2 001 = Crystal reference/2 010 = 2^{10} prescaler tap 011 = 2^{8} prescaler tap 100 = 2^{6} prescaler tap 101 = 2^{4} prescaler tap 110 = RTC extension 111 = SWDOG off
3	D2R	Divide-by-2 RESET. This bit resets the input clock divide-by-2 timer when it is written to a '1'. This capability is provided so that the divide-by-2 timer may be reset at the moment the user is changing the configuration of the timer control register. The user may not wish to reset the divide-by-2 timer in cases where the SWDOG and the RTC are both in use. A divide-by-2 timer reset for the sake of one interval timer would affect the count in the other. In this case, a '1' should NOT be written to the D2R bit. The D2R bit always reads '0'.
2:0	RTC{2:0]	RTC clock control. Provides six possible settings for choosing the input clock for the real-time interval timer (RTIT). 000 = System clock/2 001 = Crystal reference/2 010 = 2^{10} prescaler tap 011 = 2^{8} prescaler tap 100 = 2^{6} prescaler tap 101 = 2^{4} prescaler tap 110 = RTC off 111 = RTC off

3.5.6.3 Timer Interrupt Vector Register (TIV)

TIV — Timer Interrupt Vector Register **0xYF FA53**

MSB 7	6	5	4	3	2	1	0
TIV[7]	TIV[6]	TIV[5]	TIV[4]	TIV[3]	TIV[2]	TIV[1]	TIV[0]
RESET:							
0	0	0	0	1	1	1	1

Table 3-69 TIV Bit Descriptions

Bit(s)	Name	Description
7:0	TIV[7:0]	Timer interrupt vector. The timer interrupt vector register (8 bits) contains the vector number that is fetched during an interrupt acknowledge (IACK) cycle in response to a SWDOG or RTC interrupt request. This register is readable and writable at any time. Master reset will have no effect on this register.

3.5.6.4 SWDOG Service Register (SWS)

SWS — SWDOG Service Register **0xYF FA55**

MSB 7	6	5	4	3	2	1	0
SWS[7]	SWS[6]	SWS[5]	SWS[4]	SWS[3]	SWS[2]	SWS[1]	SWS[0]
RESET:							
0	0	0	0	0	0	0	0

The software service register is the location to which the SWDOG servicing sequence is written. See **3.5.9.2 Software Watchdog Interval Timer Operation (SWIT)** for instructions on the use of this register. This register is unaffected by master or power-on reset Reads of this register will always return a zero.

Table 3-70 SWS Bit Descriptions

Bit(s)	Name	Description
7:0	SWS[7:0]	SWDOG service. The software service register is the location to which the SWDOG servicing sequence is written. See **3.5.9.2 Software Watchdog Interval Timer Operation (SWIT)** for instructions on the use of this register. This register is unaffected by master or power-on reset Reads of this register will always return a zero.

3.5.7 Timer Section

The timer section consists of the prescaler (PRE), the SWDOG and RTC clock muxes (SMUX and RMUX), the SWDOG and RTC interval timers (SWIT and RTIT), and the SWDOG and RTC interval registers (SWI and RTI).

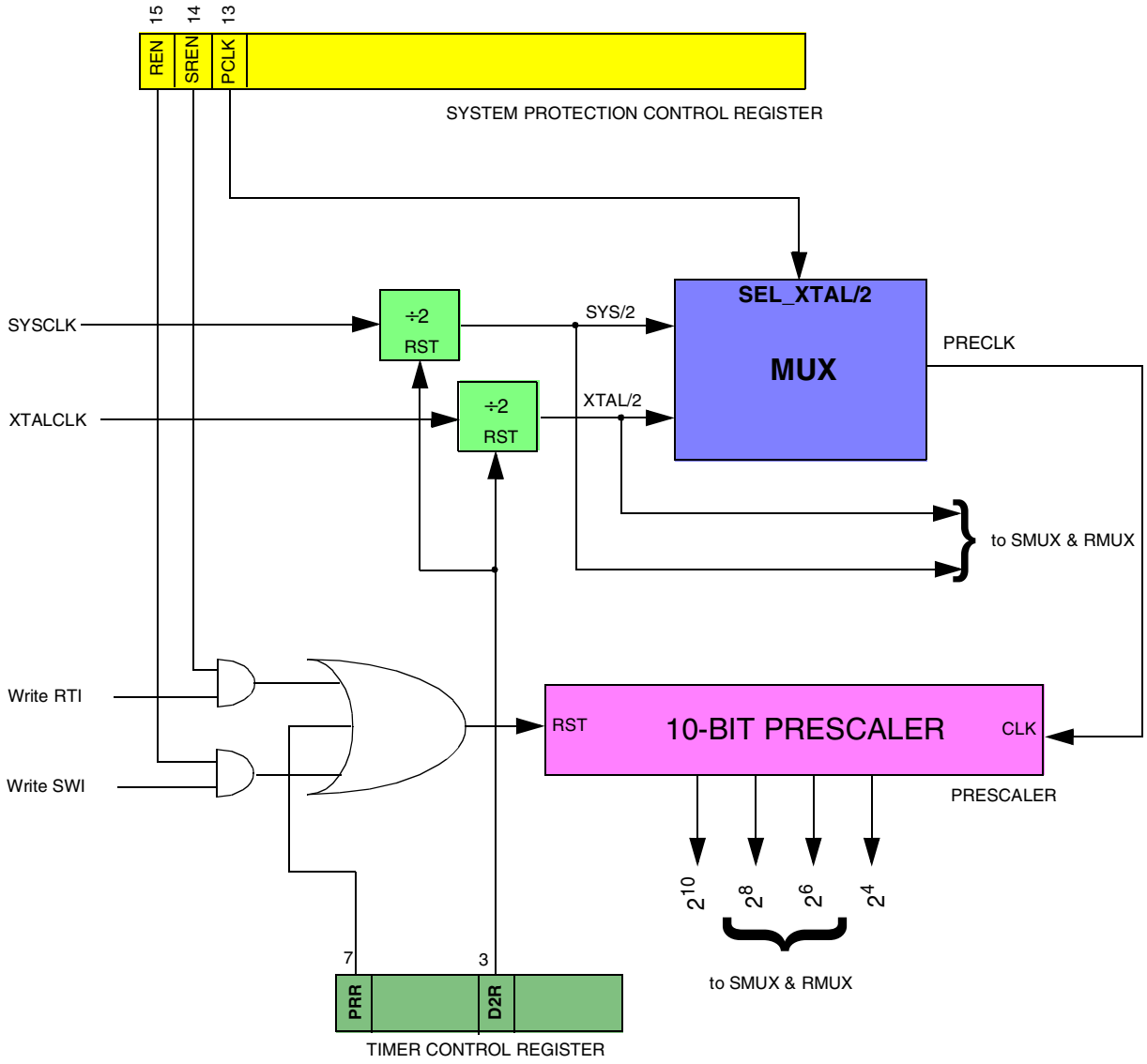


Figure 3-14 Software Watchdog, Real-Time Clock and Prescaler

3.5.7.1 System Protection Prescaler (PRE)

PRE — System Protection Prescaler Register **0xYF FA56**

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

Table 3-71 PRE Bit Descriptions

Bit(s)	Name	Description
15:0	0	SWDOG service. The SWDOG and the RTC both share a 10-bit synchronous timer which is known as the prescaler. Four different prescaler taps can be used as clocks to the SWDOG or RTC interval timers. Regardless of which clock is selected to feed the prescaler (see **3.5.2.1 Prescaler**), it is reiterated here that the clock will have been divided by two before reaching the prescaler (see **Figure 3-14**). The prescaler register is readable at any time. Since the prescaler is 10 bits, the unused bits [15:10] in the prescaler register will always be zero. If either of the reset enable bits (REN or SREN) are set, writes to the corresponding interval register will reset the prescaler. The prescaler is also reset whenever the PRR bit in the timer control register (TIC) is written to a '1'. The prescaler is unaffected by master reset and cannot be written.

3.5.7.2 Timer Section Disable

The clocks to the entire system protection sub-module may be disabled at any time by writing a '1' to the SYSPROT bit in the module disable register. See **3.1.11 Module Disable Register (MDR)** for more information on this control bit.

The timer section alone may be disabled at any time by asserting the FREEZE line while the BIM MCR FRZ[0] bit is a '1'. The clock to the software watchdog, the RTC and the prescaler will be disabled on the low phase of the system clock. It will remain disabled until the next rising edge of the system clock after the FREEZE line is negated.

3.5.8 Software Watchdog and Real-Time Clock Interrupts

The timer section has the highest priority of all the interrupt sources in the BIM. If the BIM wins the interrupt acknowledge cycle and the SWDOG or RTC has a pending interrupt request at that level, the timer interrupt vector (TIV) is placed on the intermodule bus (IMB).

3.5.9 SWDOG Interval Register (SWI)

SWI — SWDOG Interval Register **0xYF FA58**

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

Table 3-72 SWI Bit Descriptions

Bit(s)	Name	Description
15:0	0	Software watchdog interval. The software watchdog interval register (SWI) contains the value to be loaded into the software watchdog interval timer (SWIT) immediately upon the lower byte of the SWI being written. If the SWDOG is configured to interrupt at time-out (IRSEL = 1), the SWI value will be loaded into the SWIT every time it counts down to zero. A write to either the lower byte or the 16-bit word of the SWI activates the SWIT. This register is readable and writable at any time. Writes to this register will reset the prescaler if the SREN bit in the system protection control register (SYPCR) is set. A zero written to this register will turn off and clear the software watchdog interval timer without issuing a reset or interrupt request. The SWI is unaffected by master reset. This register will therefore contain the value programmed into it prior to master reset so that counting may continue during reset or freeze operation. The SWP is unaffected by master reset. This register will therefore contain the value programmed into it prior to master reset. Power-on reset will reset this register to zero.

3.5.9.1 Real-Time Interval Register (RTI)

RTI — Real-Time Interval Register **0xYF FA5A**

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

Table 3-73 RTI Bit Descriptions

Bit(s)	Name	Description
15:0	0	Real-time interval. The real-time interval register (RTI) contains the value to be loaded into the real-time interval timer (RTIT) when the lower byte of the RTI is written, if the LCI bit is set, and every time the RTIT counts down to zero. A byte write to the lower byte of this register or a 16-bit write of the RTI activates both the SWDOG and RTC timers in 32-bit interval timer mode or the RTIT when not in 32-bit interval timer mode. This register is readable and writable at any time. Writes to this register will reset the prescaler if the REN bit in the system protection control register (SYPCR) is set. A zero written to this register will turn off and clear the real-time interval timer without issuing an interrupt request. (If LCI = 0, the timer will not stop until next time-out). The RTI is unaffected by master reset. This register will therefore contain the value programmed into it prior to master reset so that counting may continue during reset or freeze operation.

3.5.9.2 Software Watchdog Interval Timer Operation (SWIT)

SWIT — Software Watchdog Interval Timer Operation Register **0xYF FA5C**

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

Table 3-74 SWIT Bit Descriptions

Bit(s)	Name	Description
15:0	0	Software watchdog interval timer operation. The SWDOG consists of a 16-bit interval timer clocked by the output of the SWDOG clock mux (SMUX). The SMUX output is applied to the SWIT which will start counting down when loaded with a non-zero value from the SWI register. The SWIT cannot be written directly. When accessing the SWI register with byte operations, the value in the SWI register will not be loaded into the SWIT until the lower bits [7:0] of the SWI are written. When the software watchdog times out, depending on the state of the IRSEL bit in the system protection control register, it will assert a reset or an interrupt request. If it is programmed to generate an interrupt request, the SWDOG interval timer will then load the current value from the SWI register and begin counting down again. The user may change the time out value in the software watchdog interval register at any time. After time-out occurs, the software watchdog zero flag (SZ) will be asserted. The software watchdog interval timer is readable at any time and writes have no effect. When master reset is asserted, the software watchdog interval timer is unaffected. See **Figure 3-15**.

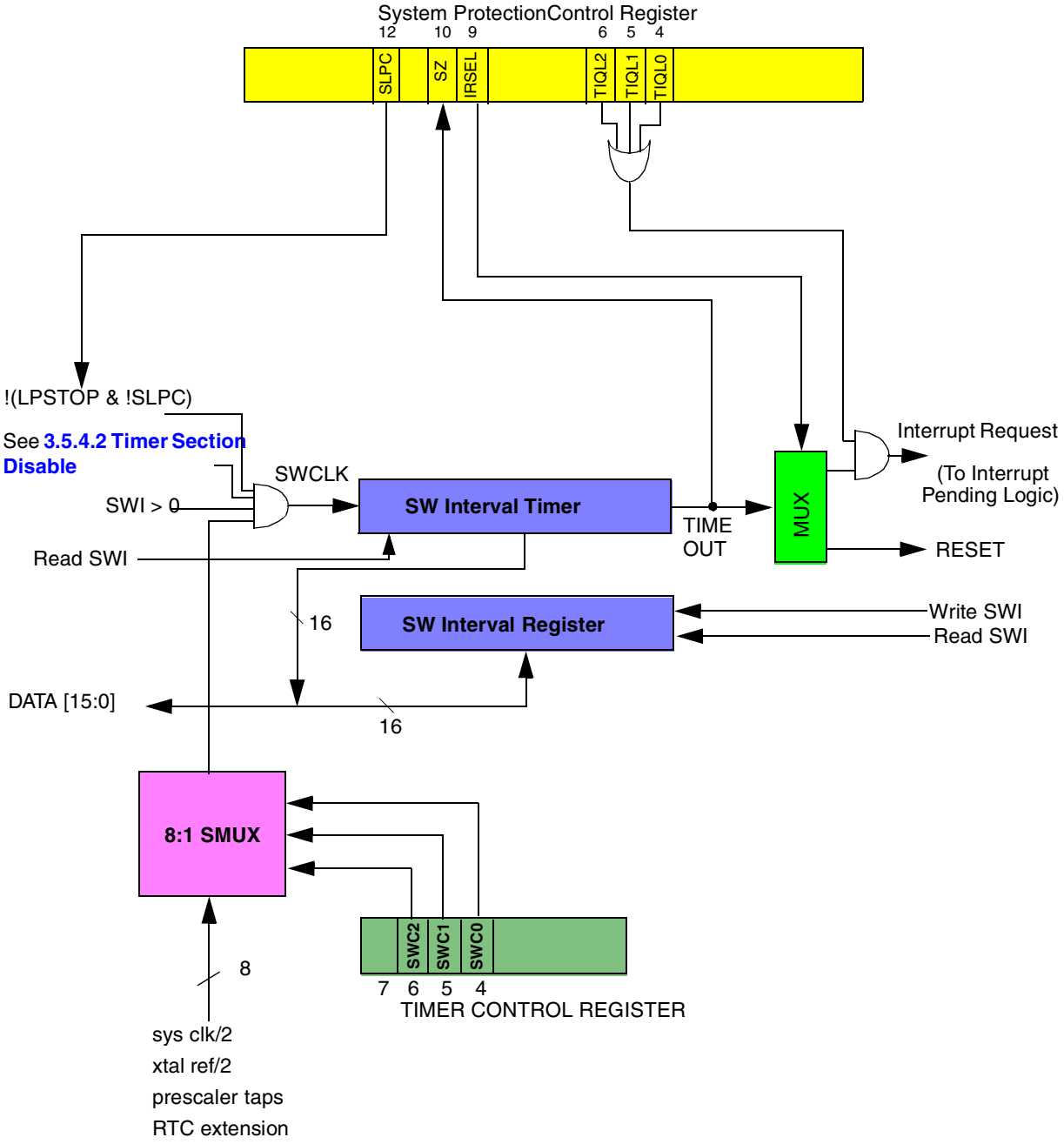


Figure 3-15 Software Watchdog Timer Block Diagram

3.5.9.3 Software Watchdog Time-Out Interval

Although most software disciplines encourage or permit the watchdog concept, all systems do not need the same time-out intervals.

The software watchdog time-out interval is set by the software watchdog interval register (SWI) and the speed of the clock applied to the SWDOG interval timer (SWIT). The clock applied to the SWIT, SWCLK, is derived from one of the following

- A clock operating at the crystal reference frequency divided by two.
- A clock operating at the system clock frequency divided by two.
- A prescaler tap — PRECLK10, PRECLK8, PRECLK6, or PRECLK4, which is system clock divided by two or the crystal frequency divided by two, and again divided by one of the prescaler tap values.

The selection of the clocks just mentioned, used to derive SWCLK, is controlled by the SWC field in the timer control register and by the PCLK bit in the system protection control register.

3.5.9.4 SWDOG Interval Calculation

The software watchdog provides a wide range of programmable time-out intervals depending on the system clock or crystal reference frequency and the input frequency chosen (refer to **Table 3-76**).

The SWDOG interval (not prescaled) is calculated as follows:

SWDOG interval = (SWI value) * 2 / (CLOCK)
where CLOCK = system clock or crystal reference frequency (whichever is chosen in the SWC field of the timer control register).

EXAMPLE

CLOCK = 2MHz (crystal ref freq) and SWI value = 256
SWDOG interval = (256 * 2) / 2 MHz = 256 uS

The SWDOG Interval (prescaled) is calculated as follows:

SWDOG interval = (SWI value) * 2 / (CLOCK / prescaler)
where CLOCK = system clock or crystal reference frequency to the prescaler (whichever is chosen in the PCLK bit of the system protection control register).

and prescaler = prescaler tap setting in the SWC field of the timer control register.

EXAMPLE

CLOCK = 33 MHz (system clock freq), SWI value = 1024,
and prescaler = 64
SWDOG interval = (1024 * 2) / (33 MHz / 64) = 4 ms

which gives the ranges in **Table 3-76**.

3.5.9.5 SWDOG Service Sequence

The special service sequence which inhibits the software watchdog time-out, and reloads the software watchdog interval timer with the value in the software watchdog interval register, consists of two steps:

- Write 0x55 to the SWDOG service register (SWS)
- Write 0xAA to the SWDOG service register (SWS)

Both writes must occur in the proper order and prior to time-out. Any number of instructions, up to the end of the time-out interval, may be executed in between each write.

When the proper service sequence occurs, the software watchdog interval timer reloads the value of the SWDOG Interval register and the process begins again.

If the time-out request is reached before a service action, the SWDOG will cause a master reset (or an interrupt request, depending upon the state of the IRSEL bit).

3.5.9.6 Real-Time Interval Counter and RTC Operation (RTIT)

RTIT — Real-Time Interval Counter and RTC Operation Register **0xYF FA5F**

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

Table 3-75 RTIT Bit Descriptions

Bit(s)	Name	Description
15:0	0	Real-time interval counter and RTC operation. The real-time clock consists of a 16-bit interval timer clocked by the RTCLK output of the RTC clock mux (RMUX). The RTCLK signal is applied to the RTIT which will start counting down when a non-zero value is loaded from the RTI. The RTIT cannot be written directly. If the LCI bit is set, the RTI will not be loaded into the RTIT until the lower byte (or 16-bit word) of the RTI is written. If the LCI bit is 0, the timer will be allowed to time out before the RTI value is loaded. When the interval timer reaches zero, an interrupt request will be generated if the TIEN bit is set and if the Interrupt Request Level, TIQL, is nonzero. The real-time zero flag always sets, regardless of the TIEN or TIQL value. At this time, the real-time interval timer loads the current real-time interval register value and begins timing again. The real-time interval timer is readable at any time and writes have no effect. When master reset is asserted, the real-time interval timer is unaffected. See **Figure 3-16** for more information.

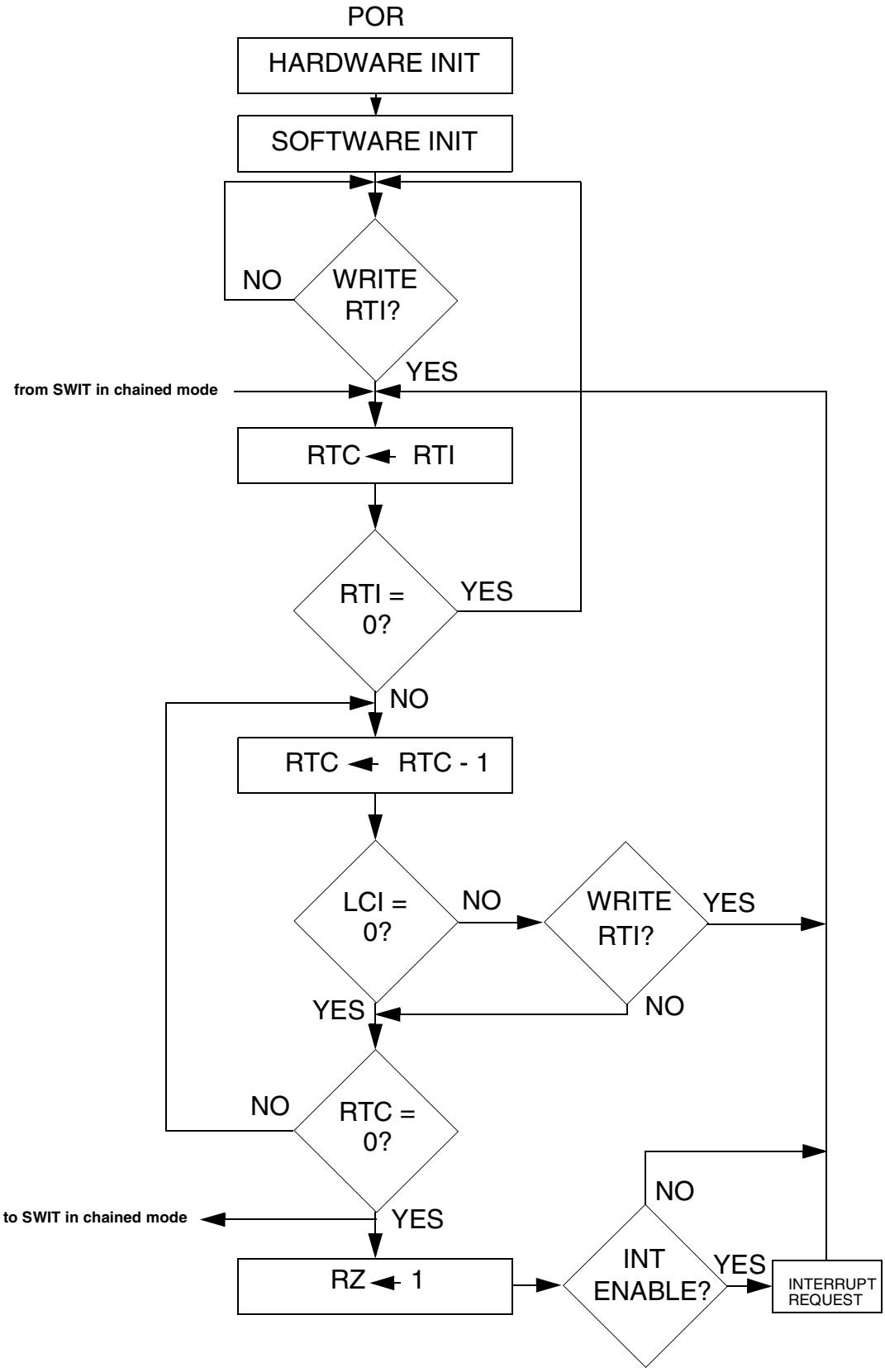


Figure 3-16 Real-Time Clock Flow

3.5.9.7 RTC Time-Out Interval

The real-time clock time-out is set by the real-time interval register (RTI) and the speed of the clock applied to the real-time interval timer (RTIT). The clock applied to the RTIT, RTCLK, is derived from one of the following

- A clock operating at the crystal reference frequency divided by two
- A clock operating at the system clock frequency divided by two
- A prescaler tap — PRECLK10, PRECLK8, PRECLK6, or PRECLK4, which is system clock divided by two or the crystal reference frequency divided by two, and again divided by one of the prescaler tap values.

The selection of the clocks is controlled by the RTC field in the timer control register and by the PCLK bit in the system protection control register.

3.5.9.8 RTC Interval Calculation

The real-time clock provides a wide range of programmable time-out intervals depending on the system clock or crystal reference frequency and the input frequency chosen (refer to **Table 3-76**).

The RTC interval (not prescaled) is calculated as follows:

RTC interval = (RTI value)x(2) / (CLOCK)
where CLOCK = system clock or crystal reference frequency (whichever is chosen in the RTC field of the timer control register).

The RTC interval (prescaled) is calculated as follows:

RTC interval = (RTI value)x(2) / (CLOCK / prescaler)
where CLOCK = system clock or crystal reference frequency to the prescaler (whichever is chosen in the PCLK bit of the system protection control register).

and prescaler = prescaler tap setting in the RTC field of the timer control register.

which gives the ranges in **Table 3-76**.

NOTE

The resolution of each range within a frequency category is its lower bound. For example, in the first row of the 2-MHz column, the resolution is one us.

A block diagram of the real-time clock is shown in **Figure 3-17**.

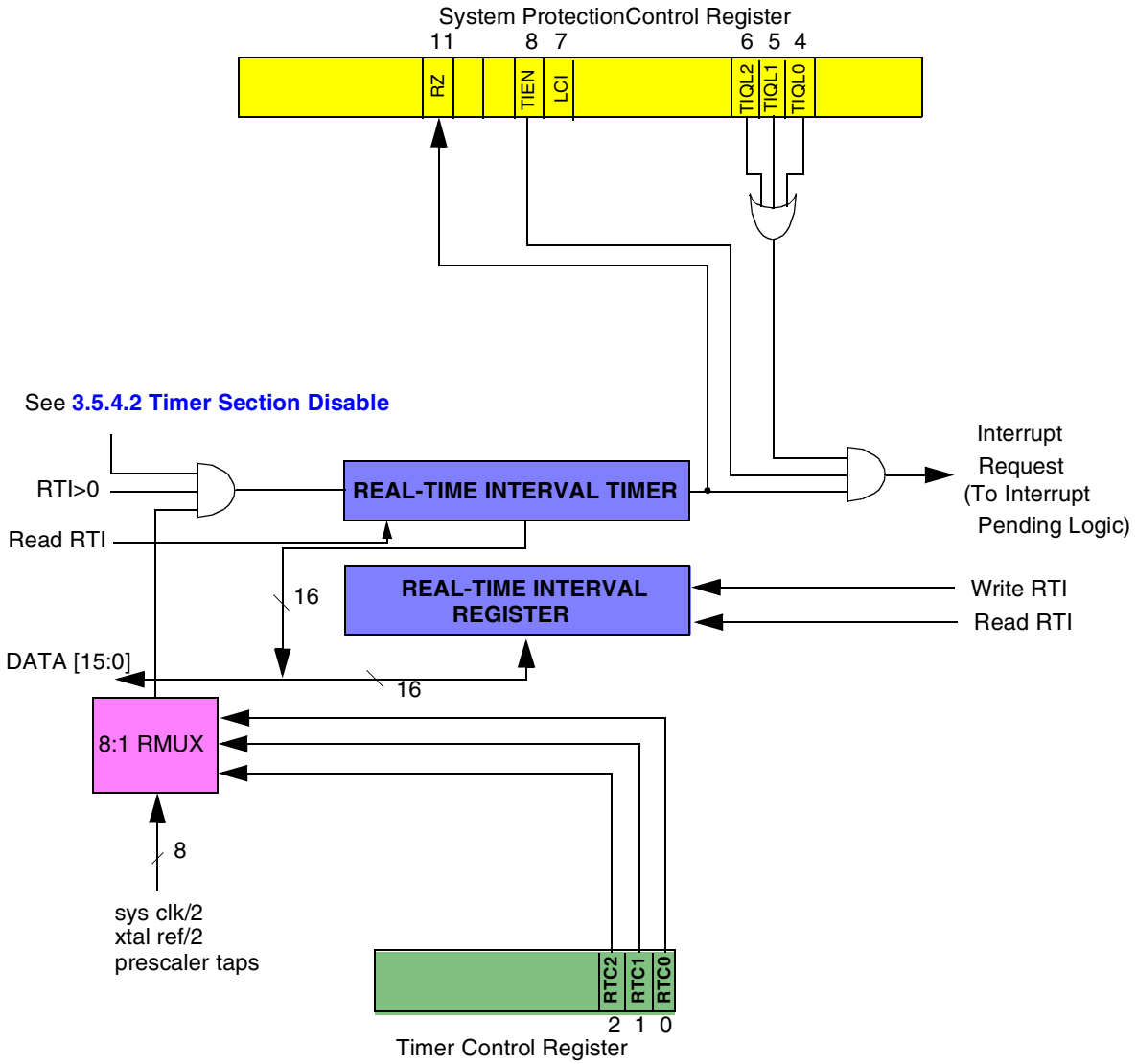


Figure 3-17 Real-Time Clock Block Diagram

Table 3-76 Examples of Timer Interval Ranges (Timers not Chained)

Prescaler Tap	Time-Out Ranges in Clock Frequency Categories			
	2 MHz	5 MHz	25 MHz	33 MHz
NONE	1 µs to 64.4 ms	400 ns to 26.2 ms	80 ns to 5.2 ms	60.6 ns to 3.96 ms
2^4	16 µs to 1 s	6.4 µs to 419.2 ms	1.28 µs to 83.2 ms	970 ns to 63.4 ms
2^6	64 µs to 4 s	25.6 µs to 1.68 s	5.12 µs to 332.8 ms	3.9 µs to 253.4 ms
2^8	256 µs to 17 s	102.4 µs to 6.71 s	20.5 µs to 1.33 s	15.5 µs to 1 s
2^{10}	1 ms to 67 s	409.6 µs to 26.83 s	81.92 µs to 5.32 s	62 µs to 4 s

3.5.9.9 RTC and SWDOG in 32-Bit Mode

The RTC and the SWDOG may be chained together to form a 32-bit interval timer. In this mode, the SWIT will act as the most significant word (MSW), and the RTIT will act as the least significant word (LSW). Upon reaching zero, rather than reloading the value in the interval register, the LSW will

- Count the SWIT down by one
- Roll over to 0xFFFF, and then
- Continue counting

One long word (32-bit) or two word (16-bit) values may be written to the SWI and RTI registers to trigger the interval timer chain.

The chained interval timer will not begin counting until the lower byte of the RTI (or 16-bit word) is written if the LCI bit is set. If the LCI bit is 0, the timer will be allowed to time out before a new RTI value is loaded.

Each time the real-time interval timer (RTIT) counts down through zero, a signal is sent to the software watchdog interval timer (SWIT) to count it down by one. When the SWIT and the RTIT have both reached zero, the real-time timer zero flag (RZ) will be set indicating that the interval timer chain has timed out. Upon timing out, provided that the TIEN bit is set and the TIQL bits are nonzero, an interrupt request will be asserted. The RTI value and the SWI value will then be loaded into each register's respective interval timer, and the chained timer will begin counting down again. The user may clear RZ, if necessary, by the standard clear mechanism.

Figure 3-19 is a flow diagram for the 32-bit chained mode operation.

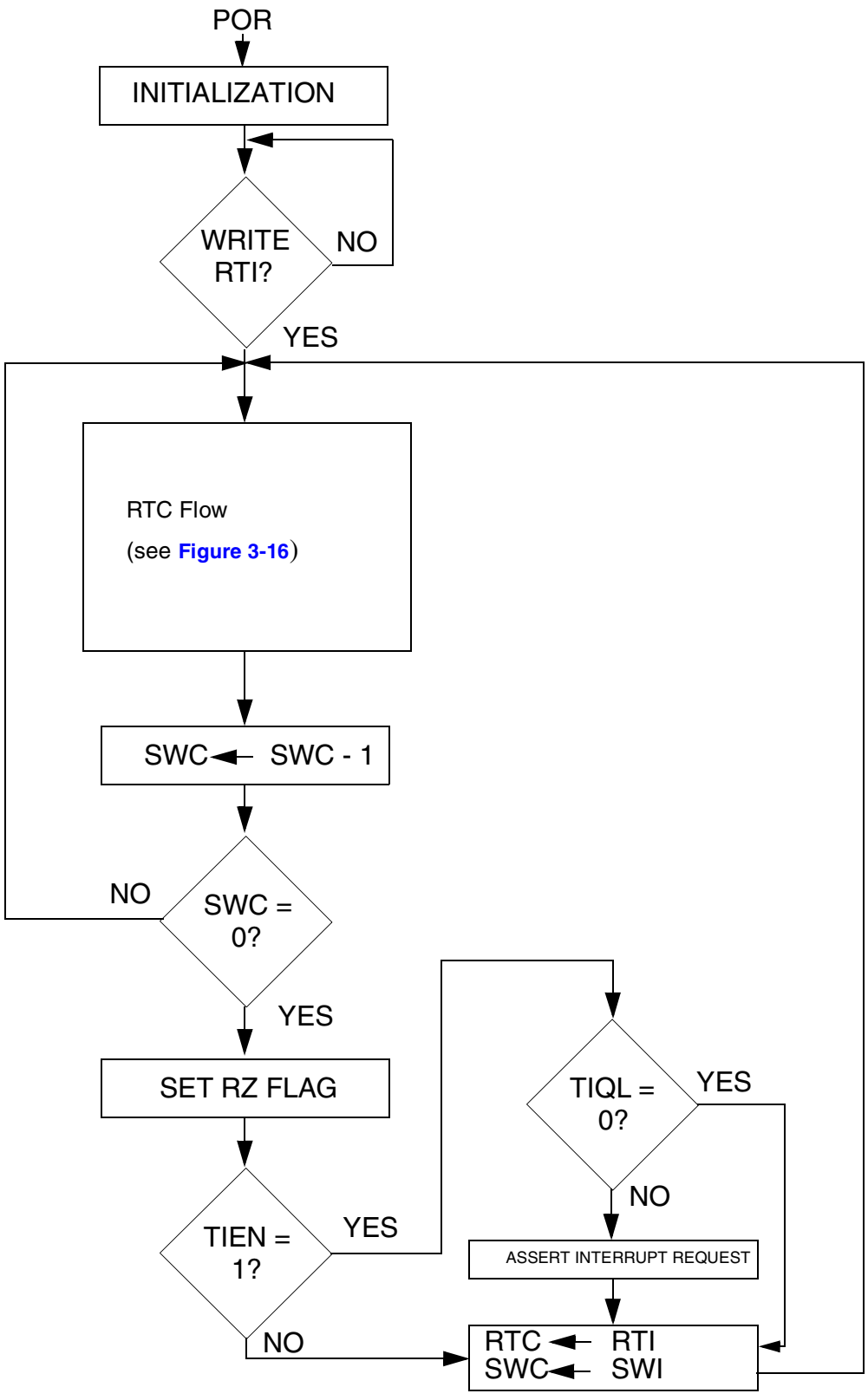


Figure 3-18 Flow Diagram of 32-Bit Chained Mode Operation

At any time during its operation, the timer chain may have a new interval value loaded into it by writing to the interval registers subject to the state of the LCI bit.

The user may read the timer at any time. However, separate, consecutive 16-bit reads may or may not be accurate, especially at times when the LSB is about to roll over and count the MSB down by one. In that particular case, consecutive 16-bit reads may result in as large as a 64-Kbyte error.

When master reset is asserted, the 32-bit timer's registers are unaffected.

3.5.9.10 32-Bit Timer Interval Calculation

The 32-bit timer provides time out intervals up to several days (refer to **Table 3-77**).

The 32-bit timer interval (not prescaled) is calculated as follows:

32-bit timer interval = (32 bit SWI/RTI value) x (2) / (CLOCK)
where CLOCK = system clock or crystal reference frequency (whichever is chosen in the RTC field of the timer control register).

The 32-bit timer interval (prescaled) is calculated as follows:

32-bit timer interval = (32 bit SWI/RTI) x (2) / (CLOCK / prescaler)
where CLOCK = system clock or crystal reference frequency to the prescaler (whichever is chosen in the PCLK bit of the system protection control register).

and prescaler = prescaler tap setting in the RTC field of the timer control register.

NOTE

The resolution of each range within a system frequency category is its lower bound. For example, in the first row of the 2-MHz column, the resolution is one us.

Table 3-77 Examples of Timer Interval Ranges (Timers Chained)

Prescaler Tap	Time-Out Ranges in System Clock Frequency Categories			
	2 MHz	5 MHz	25 MHz	33 MHz
NONE	1 µs to 1.2 hours	400 ns to 28.6 minutes	80 ns to 5.7 minutes	60.6 ns to 260.3 s
2^4	16 µs to 19 hours	6.4 µs to 7.64 hours	1.28 µs to 1.5 hours	969.6 µs to 1.2 hours
2^6	64 µs to 3.2 days	25.6 µs to 1.27 days	5.12 µs to 6.11 hours	3.9 µs to 4.6 hours
2^8	256 µs to 12.8 days	102.4 µs to 5.1 days	20.5 µs to 1 days	15.5 µs to 18.5 hours
2^{10}	1 ms to 51.2 days	409.6 µs to 20.36 days	89.92 µs to 4.1 days	62 µs to 3.1 days

3.5.9.11 Timer Application Example

This example shows the steps necessary to cause the timer to generate a level 5 interrupt request to exit LPSTOP after eight hours, in a system clocked by a 2-MHz crystal reference. A check of **Table 3-77** shows that at two MHz, we must use the 2^4 prescaler or higher. The timer control register (TIC) should be set to 0x101, thereby configuring

the RTIT to be clocked by the 2^4 prescaler and configuring the SWDOG timer to act as the RTC extension (see **3.5.6.2 Timer Control Register (TIC)**.

The SYPCR should then be set to 0xB15C in order to:

- Set the REN bit so that a write to the real-time interval register will reset the prescaler;
- Set PCLK = 1 to select the crystal frequency to clock the prescaler (see section **3.5.6.1 System Protection Control Register (SYPCR) (V4 and V5)**);
- Set the SLPC bit so that the SWDOG interval timer will continue to run in LPSTOP
- Set the TIEN bit to enable interrupt requests;
- Set an interrupt request priority level of five; and
- Leave the monitor settings in their default states (bits [3:0]).

NOTE

Writes to the IRSEL bit after selecting 32-bit timer mode will be ignored. Interrupt request generation is the default in 32-bit timer mode.

At this point, values must be calculated which will be loaded into the real-time interval register (RTI) and the SWDOG Interval register (SWI) to cause a time-out after eight hours.

Crystal frequency = 2 MHz

SWC & RTC interval	= 8 hours=	(SWI/RTI value*2)/(crystal freq/ 2^4)
	8 hours	= (SWI/RTI value*2) / (2e6 / 16)
SWI/RTI value		= 28800 s*(2e6 / 32)
		= 0x6B49D200 hex

The 32-bit interval timer will assert an interrupt request eight hours after the process began, and will count down to zero with resolution jumps of 16 us.

To make the interrupt request occur, the CPU interrupt mask must be dropped to a value below the value programmed into the TIQL bits (a 4 or lower). All that remains is to load the respective interval registers and the vector register and the timer will begin counting down as soon as the lower byte (or 16-bit word) of the RTI is written. When the RTI is loaded with the preset value, the prescaler will be reset and begin counting from precisely zero.

After eight hours, if the system was in LPSTOP mode, the 32-bit timer will issue an interrupt request which in turn will cause system wake-up from LPSTOP. The options at this point include using the interrupt handler to bring the SWDOG back on line, or writing the timer value to memory before reentering LPSTOP, or simply determining if the microcontroller's current state warrants a reset.

Figure 3-19 is a block diagram of the 32-bit mode operation.

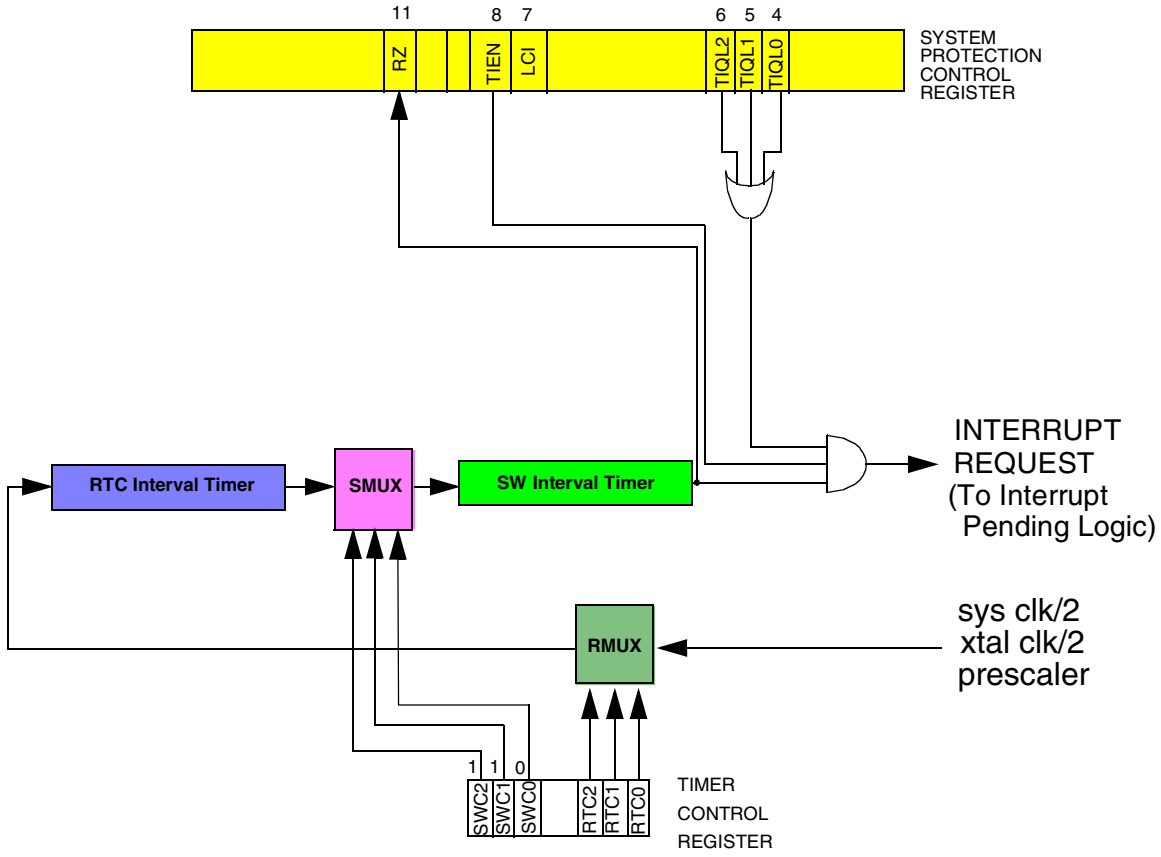


Figure 3-19 32-Bit Timer Operation Block Diagram

3.5.10 Low Power STOP Operation (LPSTOP)

When the burst integration module recognizes that the CPU has executed the “LPSTOP” instruction, the LPSTOP line is asserted.

NOTE

The LPSTOP line is *not asserted* when the CPU executes the “STOP” instruction and the RTC, SWDOG and monitors will continue to run.

3.5.10.1 RTC and LPSTOP Operation

The RTC always runs during LPSTOP unless disabled prior to executing the “LPSTOP” instruction. It will generate an interrupt request to bring the CPU out of LPSTOP if the interrupt request level (TIQL) is higher than the CPU mask level and if the TIEN bit is set. If the TIEN bit is not set, interrupt requests will not be asserted from the RTC. Refer to **3.5.6.1 System Protection Control Register (SYPCR) (V4 and V5)** for more information.

3.5.10.2 SWDOG and LPSTOP Operation

When LPSTOP is asserted, and the SLPC bit is not set, the software watchdog will be disabled on the low phase of the SWCLK, and will remain stopped until the LPSTOP line is negated. When LPSTOP negates, the timer resumes on the next rising edge of SWCLK. Otherwise, if the SLPC bit is set, the software watchdog will continue counting during LPSTOP.

The software watchdog may also be disabled during LPSTOP by writing a zero value to the SWI prior to executing the “LPSTOP” instruction.

If the system protection control register IRSEL bit is set, the software watchdog may generate an interrupt request to bring the CPU out of LPSTOP if the system protection control register interrupt request level (TIQL) is higher than the CPU mask level. If the IRSEL bit is low, however, the SWDOG may generate a reset to exit LPSTOP. Refer to **Table 3-67** for more information.

3.5.10.3 32-Bit Timer and LPSTOP Operation

Since the 32-bit timer is composed of the RTIT and SWIT, the SWIT must be configured to operate in LPSTOP, by setting the system protection control register (SYPCR) SLPC bit. Refer to **Table 3-67** for details.

As with the real-time clock, the 32-bit timer can be configured to interrupt out of LPSTOP if the system protection control register TIEN bit is set. See **Table 3-67** for more information.

3.5.10.4 Monitors and LPSTOP Operation

The bus monitor, double bus fault monitor and spurious interrupt monitor all respond to activity on the internal bus, and are effectively inactive as long as the system clock is disabled.

3.5.11 FREEZE Operation

When the FREEZE line is asserted, and the BIM MCR FRZ[0] bit is a ‘1’, the clock to the software watchdog, the real-time clock, and the prescaler will be disabled on the low phase of the system clock. The clock is only disabled within the system protection sub-module and its operation to the rest of the BIM is not affected. The clock will remain disabled until the FREEZE line is negated. It will be subsequently enabled to the system protection sub-module on the next rising edge of the system clock.

NOTE

The clock mentioned above can be either the system clock, or the crystal frequency.

When the FREEZE line is asserted and the BIM MCR FRZ[1] is a ‘1’, the bus monitor will stop counting and therefore disable BERR from occurring if DTACK or BTACK is later than the programmed bus monitor time-out interval.

3.5.12 RESET Determination

The system protection sub-module is reset when power-on reset (POR) is asserted.

Master reset will not reset bits [15:8] of the system protection control register, the timer control register, timer interrupt vector register, software watchdog interval register, real-time clock interval register, the prescaler, software watchdog interval timer, nor the real-time interval timer. The purpose here is to allow the configuration of these registers and bits to remain the same after reset so that the timer can continue to run in that configuration after a master reset.

The system protection sub-module provides two reset sources: the software watchdog and the double bus fault monitor. A reset from either source sets a flag in the BIM's reset status register (RSR).

3.6 Asynchronous Chip Select (ACS)

Typical systems require some external hardware to provide selects for external peripherals. Many of these devices do not have all the signals that are required for an asynchronous bus interface such as that of the M68000 family of parts. Additionally, higher levels of integration can provide cost, speed and reliability advantages over external logic. The asynchronous chip selects described here provide the means to minimize this external logic.

The primary function of the asynchronous chip select (ACS) submodule is to provide bus timing and bus cycle termination for external memory and peripheral devices. Each ACS can be programmed as read strobe, write strobe, output enable or interrupt acknowledge signals. The asynchronous chip selects may also be used to terminate external cycles, eliminating the external glue logic required to generate $\overline{\text{DTACK}}$.

There are seven ACS signals available. All ACS pins are optional and may not be included on a given MCU implementation incorporating the BIM. There is also a burst chip select channel which implements burst mode protocols for accessing burst memories.

3.6.1 Feature List

The following is a list of the features supported by the ACS module.

- Seven programmable asynchronous chip selects $\overline{\text{CS1}}$ — $\overline{\text{CS7}}$:
  - — Asynchronous chip selects may be independently programmed with the various selectable features.
- Various block sizes:
  - — The block size, which starts from the base address, can be programmed to be:
    64 Kbytes, 128 Kbytes, 256 Kbytes, 512 Kbytes, 1 Mbyte, 2 Mbytes, 4 Mbytes and 8 Mbytes.
- Support for 8-bit and 16-bit external devices:
  - — The port size can be programmed to be 8 or 16 bits. Eight-bit ports are accessible on both odd and even addresses when connected to D15-D8. Sixteen-bit ports can be accessed as odd-addressed bytes, even-addressed

bytes, or words. For byte transfers involving a 16-bit port, separate upper and lower byte asynchronous chip selects may be programmed.

- Read only, write only or read/write select:
  - — A block of memory can be designated as read only, write only or read/write.
- Flexible asynchronous chip select assertion options
  - — Asynchronous chip select signals assert in S0/S1($\overline{CE}$ timing) or the same bus state as $\overline{DS}$, so that control signals to memories or peripherals, such as $\overline{CE}$, $\overline{OE}$, and $\overline{WE}$, can be easily generated.
- Bus cycle termination with wait states:
  - — This option allows DTACK to be generated internally to terminate the bus cycle. The desired number of wait states can be programmed by the user to interface with various devices.
- Support for slow bus interface device:
  - — Additional timing flexibility allows asynchronous chip select signals to negate one clock before the DTACK generator ends the bus cycle.
- Address space checking:
  - — Supervisor, user, program, data, and CPU space accesses can be optionally checked. The CPU space type is checked by programming the desired type in the proper base address register bits.
- IACK cycle support:
  - — An asynchronous chip select can be programmed to match IACK cycles. The corresponding pin can be asserted for all IACK cycles or only for a specific interrupt priority level.

3.6.2 Asynchronous Chip Select Block Diagram

Figure 3-20 shows a programmable asynchronous chip select. All asynchronous chip selects have the same structure. All signals used to generate asynchronous chip select signals are taken from the module's internal bus, which originates in the bus interface unit (BIU) and in the external bus interface (EBI). Each asynchronous chip select has a base address register and option register which contain the programmable characteristics of a particular ACS pin.

There is only one $\overline{DTACK}$ generator in the asynchronous chip select submodule, and it is shared by all of the asynchronous chip selects. The active asynchronous chip select for a particular bus cycle determines the number of wait states produced by the $\overline{DTACK}$ generator before the IMB cycle is terminated.

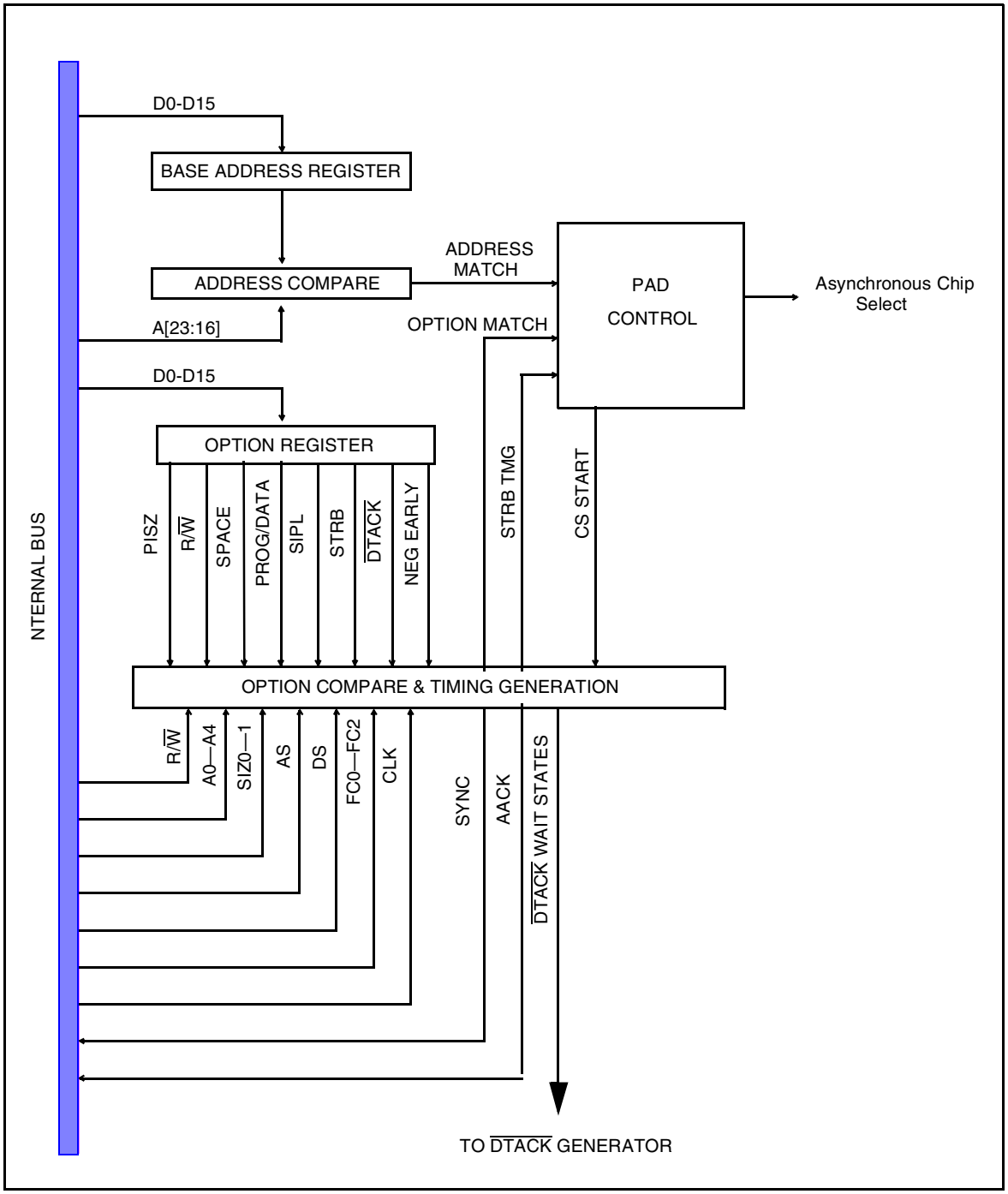


Figure 3-20 Block Diagram of Asynchronous Chip Select

3.6.3 Asynchronous Chip Select Default Attributes

After the MCU resets, the seven asynchronous chip select channels are disabled from participating in the address / option matching process, so that an external device cannot be accidentally selected until an initialization program sets up the base address registers and the option registers.

3.6.4 Asynchronous Chip Select Operations

Each ACS pin can provide a timing signal for an external device and internal bus cycle termination. These functions are enabled independently. To enable the ACS to provide an external timing signal, the ACS function is selected in the corresponding port's pin assignment register. To enable the internal generation of $\overline{\text{DTACK}}$ to terminate an IMB bus cycle, the DTACK field in the asynchronous chip select's option register is set to any value except "external $\overline{\text{DTACK}}$".

Both ACS pin assertion and bus cycle termination depend on an initial address / option match for activation. ACS attributes are defined by its base address register and option register. Each asynchronous chip select participates in the address/option matching process unless the R/$\overline{\text{W}}$ field in its option register is set to "chip select match disabled".

During the matching process, the programmed number of bits from the base address register for each ACS pin are compared against the corresponding internal address bits to determine whether an address match has occurred. This match is further qualified by a comparison of the internal read/write signal, space type, and access size with the programmed values in each asynchronous chip select's option register. When the address and option information match the current cycle, the ACS pin is asserted. If no ACS matches the bus cycle information for an external access, the access is assumed to be to a device with a 16-bit data interface.

More than one asynchronous chip select can be active for a bus cycle. However there is limited interaction between the DTACK field and the PSIZE field of all active ACS pins. If the same number of wait states are selected by each active ACS, the cycle terminates in the expected number of wait states. If the number of wait states differ amoung asynchronous chip selects, the cycle termination operation is undefined. If the size field for any active asynchronous chip select(s) specifies an 8-bit device, the data transfer will be an 8-bit transfer on D[15:8]. If the programmed PSIZE of any two active asynchronous chip selects are inconsistent (a mix of 8- and 16-bit programming), ACS operation is undefined.

The pin corresponding to the active asynchronous chip select may assert for external data transfers with various timing as shown in **Table 3-78**. The pin can be asserted in S0/S1 ($\overline{\text{CE}}$ timing) or in the same state that $\overline{\text{DS}}$ asserts, as selected by the STRB bit in its option register. By selecting appropriate options, an asynchronous chip select can produce a read strobe, write strobe, or output enable for an external device. For example, "Read/Write with CE timing" can be used for a chip enable ($\overline{\text{CE}}$) control, "Read only synchronized with CE" can be used for a read strobe, "Read only synchronized with $\overline{\text{DS}}$" can be used for an output enable ($\overline{\text{OE}}$), and "Write only synchronized with $\overline{\text{DS}}$" can be used for a write strobe.

NOTE

$\overline{CE}$ timing and $\overline{DS}$ timing is different on a read cycle. $\overline{CE}$ timing may assert the pin as early as S0 without regard to internal cycle indication (IAACK). While the $\overline{DS}$ timing option always asserts the pin in S1 with IAACK included in the assertion equation.

To support external devices with synchronous $\overline{OE}$ requirements or slow bus interface logic, the asynchronous chip select pin may be negated one clock before the bus cycle is terminated by the DTACK generator (see **Table 3-81**). This feature provides ample time for a slow bus interface device (during write cycles) to disconnect itself from the current bus cycle before another bus cycle starts.

When no ACS pin is available or when the digital I/O function is selected for the ACS pin, the active asynchronous chip select(s) can still terminate external cycles based on its DTACK value. After the programmed number of wait states expires, DTACK is asserted internally to terminate the IMB bus cycle. If the external $\overline{DTACK}$ pin is asserted before DTACK is asserted internally, the bus cycle is terminated by the external $\overline{DTACK}$.

Table 3-78 ACS Timing Summary

Address	CS Option Register Encodings				CS Timing	
	STRB	DTACK	Negate Early	R/$\overline{W}$	Assert	Negate
SLAM: internal or external	CE	X	X	X	B2	B4
SLAM: internal or external	DS	X	X	X	B2	B4
Internal	CE	X	X	X	B2	B4
External	CE	0	X	X	S1	S5
External	CE	1...14	0	X	S1	S5
External	CE	1...14	1	X	S1	Last S3
External	CE	**External**	X	X	S1	S5
Internal	DS	X	X	X	No assertion	
External	DS	0	X	R	S1	S5
External	DS	1...14	0	R	S1	S5
External	DS	1...14	1	R	S1	Last S3
External	DS	**External**	X	R	S1	S5
External	DS	0	X	W	No assertion	
External	DS	1...14	0	W	S3	S5
External	DS	1	1	W	NOT DEFINED	

Table 3-78 ACS Timing Summary (Continued)

Address	CS Option Register Encodings				CS Timing	
	STRB	DTACK	Negate Early	R/$\overline{W}$	Assert	Negate
External	DS	2...14	1	W	S3	Last S3
External	DS	External	X	W	S3	S5

3.6.4.1 CPU Space Cycle Selection

The asynchronous chip select can also be programmed to identify an IACK cycle, LPSTOP cycle, or breakpoint acknowledge cycle. **Table 3-79** shows the pertinent information for programming a ACS to match a CPU space cycle. Bits [15:8] in the base address field must be programmed to match the cycle address bits and the block size must be programmed to 64 Kbytes to allow the address comparator to check the CPU space type on address lines A[19:16]. When the SPACE field is set to “CPU space”, the PROG/DATA field is ignored by ACS logic. The R/$\overline{W}$ field must be set for the particular cycle. The correct setting of PSIZ for an LPSTOP or breakpoint cycle is system dependent. For an IACK cycle, PSIZ is set to “8-bit” to fetch a vector from D15-D8, and PSIZ is set to “16-bit, lower byte” or “16-bit, both bytes” to fetch a vector from D7-D0. For an interrupt acknowledge cycle, the ACS can also check the interrupt priority level, which is given on A[3:1]. A specific interrupt level or IPL (equal to the CS number) is preassigned to each asynchronous chip select. When IPL matching is enabled, the asynchronous chip select will only assert for IACK cycles that match its specific interrupt level. See **Table 3-79**.

Table 3-79 CPU Space Cycle Information

Cycle Type	FC[2:0]	A[23:20]	A[19:16]	R/$\overline{W}$
IACK	111	1111	1111	read
LPSTOP	111	0000	0011	write
Breakpoint	111	0000	0000	read

3.6.4.2 Chip Select Module Disable

When the STOPCS bit in the BIM module disable register (MDR) is set, the chip select clocks are disabled to save power.

3.6.5 ACS Register Definitions

The asynchronous chip selects are configured by individual base address registers and option registers. These registers are read/write from the IMB. Read operations on the unused bits return zeros and writes to these bits do not have any effect.

3.6.5.1 ACS Address Base Registers (CSBAR1 — CSBAR7)

The base address register consists of a base address field and a block size field. the base address is the starting address for a block enabled by the given asynchronous chip select. The block size determines the extent of the address space from its base

address. Each asynchronous chip select has its own individual base address register, so the address map can be efficiently assigned in any application.

CSBAR1 — Base Address Register **0xYF FA60**
CSBAR7 — Base Address Register **0xYF FA78**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
A23	A22	A21	A20	A19	A18	A17	A16	RESERVED					BLOCK SIZE		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3-80 CSBAR1—CSBAR7 Bit Descriptions

Bit(s)	Name	Description
15:8	A[23:16]	Base address fields. The base address field is contained in bits [15:8] of each base address register. Bits [15:8] are used to set the starting address of a particular address space. The address compare logic uses only the most significant address bits to cause an address match within its block size, thus the value of the base address may be selected from any multiple of the block size. **EXAMPLE**: Assuming a block size of 64 Kbytes, the comparator will compare address lines A[23:16] to bits [15:8] of the base registerTherefore, each 64-Kbyte block starts on a 64-Kbyte boundary in the address map.
6:3	—	Reserved
2:0	BLOCK SIZE	Block size fields. Block size is specified by bits [2:0] of each base address register. This field is used to determine the size of the address space enabled by the asynchronous chip select. For decoding a CPU space cycle, A[19:16] must be compared, so the block size must be programmed to 64 Kbytes. 000 = 64 Kbytes, address bits compared = A16 001 = 128 Kbytes, address bits compared = A17 010 = 256 Kbytes, address bits compared = A18 011 = 512 Kbytes, address bits compared = A19 100 = 1 Mbytes, address bits compared = A20 101 = 2 Mbytes, address bits compared = A21 110 = 4 Mbytes, address bits compared = A22 111 = 8 Mbytes, address bits compared = A23

3.6.5.2 ACS Option Registers (CSOR1 — CSOR7)

Each option register consists of individual fields, which determine the timing and the conditions for asserting the asynchronous chip select signals. Since the option register can be used to generate a number of different types of signals required to interface with external devices, care must be taken to program it properly for the desired access type.

CSOR1 — Asynchronous Chip Selection Option Register **0xYF FA62**
CSOR7 — Asynchronous Chip Selection Option Register **0xYF FA7A**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
PSIZ		R/$\overline{W}$		SPACE		PROG/DATA[1]		NEGATE EARLY	RESERVED	SIPL	STRB	DTACK (WAITS) (WAITS)			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NOTES:
1. The PROG/DATA field is ignored if the SPACE field is set to CPU space encoding.

Table 3-81 CSOR1 — CSOR7 Bit Descriptions

Bit(s)	Name	Description
15:14	PSIZ	Port size. The PSIZ option field defines whether the external device supported by the asynchronous chip select is an 8-bit or 16-bit. Additionally, if the device is 16 bits, PSIZ defines whether the asynchronous chip select decodes the upper byte, lower byte or both bytes of the data bus. When a asynchronous chip select is programmed for an 8-bit device, the external device must be connected to D[15:8]. The asynchronous chip select asserts for all addresses which produce an address/option match without regard to whether the transfer is to the upper/lower byte of the data bus. For 16-bit accesses, the BIM initiates multiple bus cycles and provides byte multiplexing of data as needed to satisfy the CPU data transfer. For an access of a 16-bit devicet, the PSIZ field controls the asynchronous chip select pin assertion to properly handle the bus transfer. The asynchronous chip select logic always recognizes which byte(s) is currently selected by the CPU by checking the internal SIZE0-SIZE1 and A0 signals. When a device is 16-bits, PSIZ is programmed to $3 to select the external device for both upper and lower byte transfers. However, when a 16-bit device is implemented by two 8- bit devices, separate enables are required for upper and lower bytes. In this case, two asynchronous chip selects can be programmed as controlling 16-bits. Then one asynchronous chip select is programmed for lower byte (0x1) and the other asynchronous chip select is programmed for upper byte ($2) to enable the correct external device for a byte access. If a MOVEP (Move Peripheral) instruction is used (CPU32 family) to access an 8-bit peripheral, the PSIZ bits in the option register must be set to 16-bit, and either the upper byte or lower byte must be selected according to the peripheral's connection to the data bus. 00 = 8-bit device 01 = 16-bit device, lower byte 10 = 16-bit device, upper byte 11 = 16-bit device, both bytes
13:12	R/$\overline{W}$	Read/write. This option determines whether a asynchronous chip select matches only read bus cycles, only write bus cycles, or both read and write bus cycles. This option field in combination with the STRB field allows the user to generate a variety of control signals for external devices. This option field must be programmed to a non-zero value to allow the corresponding asynchronous chip select to participate in the address/option matching process. When all zeroes are programmed, the asynchronous chip select logic is effectively disabled. The associated pin is driven high, and $\overline{DTACK}$ cannot be asserted internally by that asynchronous chip select. 00 = CS disabled 01 = Read only 10 = Write only 11 = Both

Table 3-81 CSOR1 — CSOR7 Bit Descriptions (Continued)

Bit(s)	Name	Description
11:10	SPACE	Address space. The SPACE option field is compared with the address space indicated by the function codes generated by the CPU during each bus cycle. There are four options which the address space field can select: CPU space, user space, supervisor space and supervisor/user space. If the user application needs to distinguish a program or data space access in supervisor or user space, the PROG/DATA bits can be used. Asynchronous chip select logic recognizes the function codes $0-3 as user spaces, function codes 0x4-6 as supervisor spaces, and function code $7 as CPU space. If the MOVES instruction generates an undefined function code, the space field match is determined by the preceding rule. 00 = CPU space 01 = User space 10 = Supv space 11 = S/U space
9:8	PROG/DATA	Program/data space. This field specifies how to decode program/data space accesses (FC[1] matching) when the SPACE field is not set to the CPU SPACE encoding. If the program/data field is set to 0x11, the operation of the CS match logic is undefined. This field is ignored when the SPACE field is programmed for CPU space. 00 = Data or program 01 = Data space 10 = Prog space 11 = Reserved
7	NEGATE EARLY	Negate CS one clock before cylce terminates. Devices with slow bus interfaces may require extra time to deselect from the current bus cycle due to the device's synchronous $\overline{OE}$ response time or the need for increased data hold time on write cycles. When NEGATE EARLY is cleared the CS pin will negate in S5. When NEGATE EARLY is set, the effect on the CS pin is specified by **Table 3-78**. If cycle termination via the DTACK generator is overridden by the assertion of any internal or external source of DTACK, the CS pin will negate in S5. EXAMPLE: If device access time requires one wait state and the device bus interface is slow, program the DTACK field for two wait states and set the NEGATE EARLY bit. 0 = Negate CS in S5 1 = Negate CS 1 clock before cycle ends
6	—	Reserved
5	SIPL	Specific interrupt level. When an asynchronous chip select is programmed to respond to IACK cycles based on its base address register and its option register encodings, the SIPL option bit determines whether the interrupt priority level (IPL) is compared with the asynchronous chip select's assigned IPL as part of the address/option matching process. Since each asynchronous chip select is assigned to a specific interrupt level, the user must choose the asynchronous chip select with the desired IPL to match IACK cycles. When SIPL is set to one, the assigned interrupt level is compared with the encoded level on A3-A1. Then the asynchronous chip select is only asserted if the IPL matches. If IPL checking is not desired, SIPL is set to zero. In this case the asynchronous chip select will assert on all IACK cycles. 0 = No IPL checking 1 = Only match assigned IPL

Table 3-81 CSOR1 — CSOR7 Bit Descriptions (Continued)

Bit(s)	Name	Description
4	STRB	Strobe timing. This option determines the asynchronous chip select pin assertion state if a match occurs. On a zero wait state write cycle (two clock) the external cycle does not include state S3, therefore neither $\overline{DS}$ nor any asynchronous chip select programmed with STRB = 1 will assert. 0 = Pin asserts as quickly as possible (may assert as early as mid S0) to control device $\overline{CE}$ pins 1= Pin asserts in the same external state as $\overline{DS}$ (read = S1, write = S3)
3:0	DTACK (WAITS)	The DTACK option field determines whether $\overline{DTACK}$ is internally generated. Since DTACK is asserted internally after the programmed number of wait states expires, the user can adjust the bus timing to accommodate the access speed of any external device. With up to 14 wait states selectable, even slow devices can be interfaced with the MCU. If a asynchronous chip select matches on an internal cycle, the DTACK generator is not enabled to assert IDTACK regardless of the field encoding. 0000 = No wait states 0001 = 1 wait states 0010 = 2 wait states 0011 = 3 wait states 0100 = 4 wait states 0101 = 5 wait states 0110 = 6 wait states 0111 = 7 wait states 1000 = 8 wait states 1001 = 9 wait states 1010 = 10 wait states 1011 = 11 wait states 1100 = 12 wait states 1101 = 13 wait states 1110 = 14 wait states 1111 = No internal $\overline{DTACK}$ generation

3.7 Burst Chip Select (BCS)

The burst chip select (BCS) facilitates interfacing industry standard NVRAM and SRAM memory to the BIM without the use of additional glue logic. The BCS is designed to provide improved system performance by the use of burst data transfers to pre-fetch instructions or to support an on-chip cache controller.

3.7.1 Feature List

The following is a list of the major features of the burst chip select.

- Boot chip select — The BCS is the active chip select coming out of reset and can boot from 8- or 16-bit memories.
- Supports industry standard 8- and 16-bit asynchronous memories, 8- and 16-bit synchronous memories, and 16-bit burst mode memories using three standard burst mode protocols.
- Provides better overall system performance with slower memories than previous integration modules.
- The setup and hold times have been improved over the SIM, SCIM, SCIM2, and SLIM which allows faster internal system clock frequencies to support slower external memories.
- High performance chip enable (HP$\overline{CE}$) circuitry provides programmability between high performance performance and low power memory control.
- Programmable burst memory boundary — to support both pre-fetch and cache system architectures, burst data cycles may be terminated at a 2^N – 1 address

boundary (3, 7, 15, or 31) or after 2^N data transfers (4, 8, 16, 32).

- Bus cycle termination with wait states — the desired number of wait states (to the first data transfer and between burst data transfers) can be programmed by the user. These options control when $\overline{\text{DTACK}}$ and $\overline{\text{BTACK}}$ will be generated internally.
- Programmable block sizes — the memory depth, starting from a programmable base address, can be set to 64 Kbytes, 128 Kbytes, 256 Kbytes, 512 Kbytes, 1 Mbyte, 2 Mbytes, 4 Mbytes, and 8 Mbytes.

3.7.2 BCS and External Memory Configurations

The BCS can be configured to provide the control signals for several different memory configurations. The primary pin set of the BCS consists of five control signals:

- Burst clock (BCLK)
- Load burst address ($\overline{\text{LBA}}$)
- Burst address advance ($\overline{\text{BAA}}$)
- Burst output enable ($\overline{\text{BOE}}$), and
- Burst write enable ($\overline{\text{BWE}}$).

Additionally, some burst RAM configurations require chip select 7 ($\overline{\text{CS7}}$) to provide a $\overline{\text{CE}}$ function.

The set of configurations the BCS supports can be divided up into three categories: one memory, an asynchronous memory, a burst capable memory, and two identical burst capable memories. These are illustrated in **Table 3-82**:

Table 3-82 External Memory Configurations

One Memory	An Asynchronous Memory and a Burst Capable Memory	Two Identical Burst Capable Memories
8-bit/16-bit asynchronous memory	One 8 bit or one 16 bit asynchronous memory and a 16 bit AMCU burst protocol memory	Two 16-bit AMCU burst protocol memories
8-bit/16-bit synchronous memory	One 8 bit or one 16 bit asynchronous memory and a 16 bit pipeline protocol memory	Two 16-bit pipeline protocol memories
16-bit AMCU burst protocol memory		
16-bit pipeline protocol memory		

Table 3-83 lists examples of the types of memories the BCS controls directly. Many other industry standard memories conform to the memory protocols listed below and are compatible with the BCS.

Table 3-83 Directly Supported Memory Memories

Memory	Description	Memory Type	Port K Pins Connect to Memory Pins				
			BCLK	LBA	BAA	BOE	BWE
AM29HL162B (AMD)	1 Mbyte x 16 Sync burst FLASH Burst length = 32	00	CLK	LBA sync 1 CLK	BAA sync	OE async	WE async
28F016XS (Intel)	1 Mbyte x 16 Sync pipeline FLASH Burst length = 32	10	CLK	ADV# sync each BA	CE# sync @S0	OE# async	WE# async 3 CLK
CY7C1032 (Cypress)	64 Kbytes x 16 Sync burst RAM Burst R/W 2-1-1 Burst length = 4	00	BCLK	ADSC sync 1 CLK	ADV sync	OE async	$\overline{WH}$[1] async
MCM67M618 (Motorola)	64 Kbytes x 18 Sync burst RAM Burst R/W 2-1-1 Burst length = 4	00	K	TSC&E sync 1 CLK	BAA sync	G async	$\overline{UW}$[2] async
NM27P6841 (National)	64 Kbytes x 16 Sync burst EPROM Burst length = 4	00	BCLK	TS sync 1 CLK	nc	CS0 async	nc
HN62W5016 (Hitachi)	32 Kbytes x 16 Sync burst MROM Burst length = 8	—	BCLK	—	—	—	—
Industry standard	Async FLASH / EEPROM	11	nc	nc	CE async	OE async	WE async
Industry standard	Async EPROM / ROM / OTP	11	nc	nc	CE async	OE async	nc
Industry standard	Sync RAM (≤20 ns @33 Mhz) Burst R/W 2-1-1	10	CLK	LBA	CE async	OE async	$\overline{WE}_{hi}$ async
Industry standard	Async RAM (>30 ns) Non-burst R/W	11	nc	CE	$\overline{WE}_{lo}$ async	OE async	$\overline{WE}_{hi}$ async

NOTES:
1. An asynchronous chip select (ACS) must be used to control WL for byte write access, $\overline{CS}$ must be controlled by an ACS, external logic, or grounded.
2. An asynchronous chip select (ACS) must be used to control LW for byte write access.

3.7.2.1 Specific Examples:

The NV memory configurations that the BCS supports are shown in **Figure 3-21**. The BCS provides full programmable control for a single NV burst memory. This memory configuration requires the use of asynchronous chip select (ACS) pins to control the SRAM.

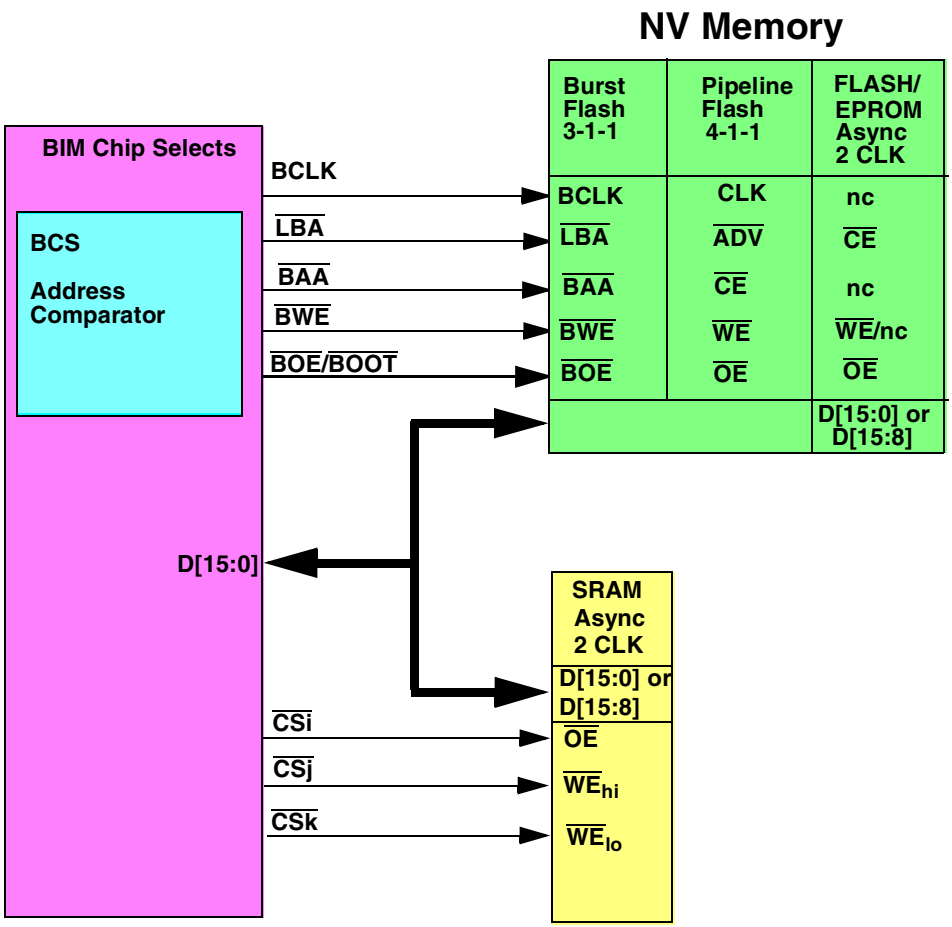


Figure 3-21 Supported NV Memory Configurations

The BCS can be configured to control SRAM memories as shown in **Figure 3-22** provided $\overline{CS7}$ is available to control the burst memory $\overline{CE}$. In this configuration, the BCS drives the $\overline{OE}$ of the asynchronous boot [NV] memory through the $\overline{BOE}$ pin during boot-up configuration, and while code is copied to the SRAM using non-burst accesses. After code is copied to the burst SRAM, and the BCS is programmed to burst on a $\overline{CS7}$ address comparator match. Once this is done, the user must guarantee that the code does not try to run in the BCS address range because the BIM will attempt to burst out of the asynchronous memory. By implementing the OEs in this manner, the asynchronous NV memory is used as the boot memory at reset and then the burst SRAM is used after boot configuration to improve system performance.

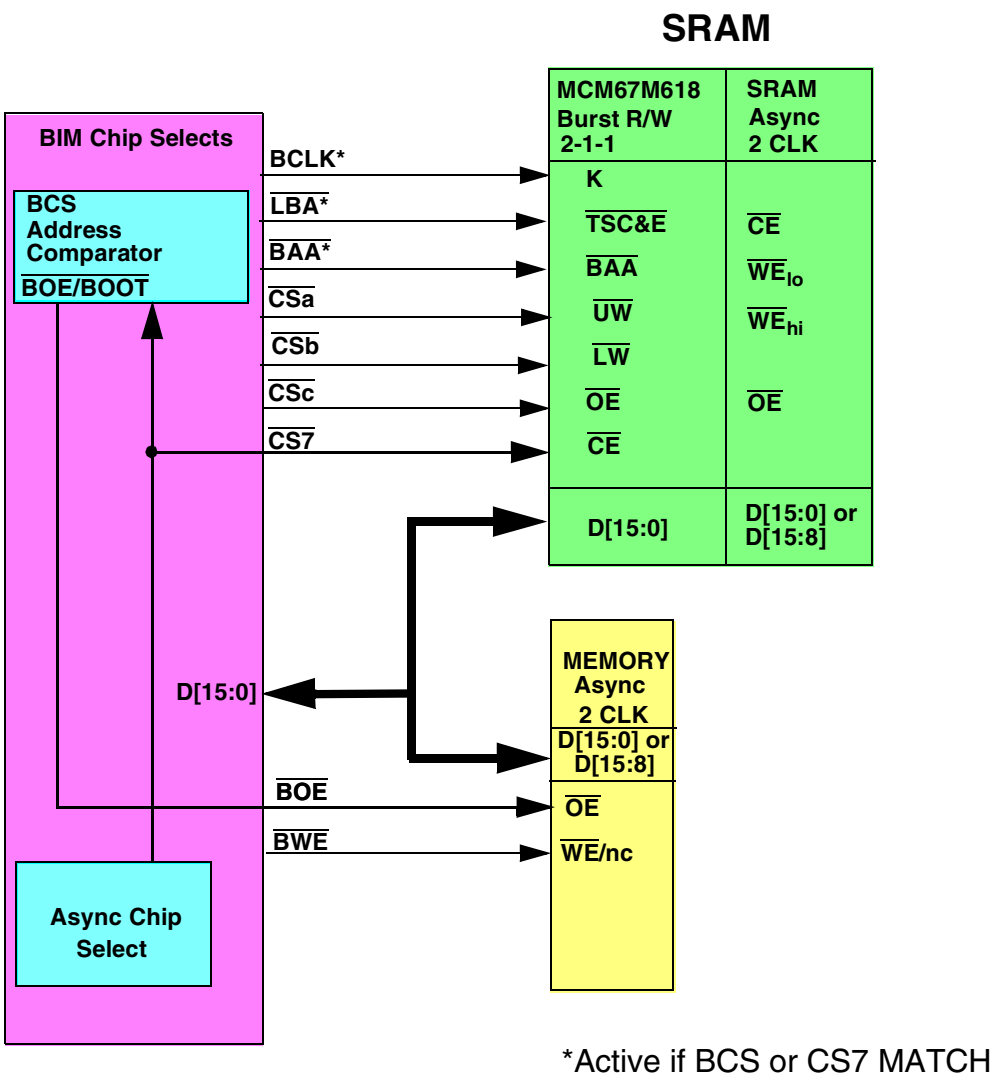


Figure 3-22 SRAM Configurations

The BCS can also be configured to control two memories which use the same burst protocol, as shown in **Figure 3-23**, provided at least three other ASC pins are available. If the NV memory is the boot memory, the chip enable must be tied to ground. If it is not the boot memory, the chip enable can be tied to $\overline{CS7}$ to save power.

In this configuration, the BCS drives the clock, load burst address and burst address advance functions of both memories through the BCLK, $\overline{LBA}$ and $\overline{BAA}$ pins. The $\overline{OE}$ and $\overline{WE}_{\overline{HI}}$ pins of the burst NV memory are driven through the $\overline{BOE}$ and $\overline{BWE}$ pins. The upper and lower byte write enables and the output enable of the burst SRAM are controlled by the three asynchronous chip select (ACS) pins. The $\overline{CS7}$ address comparator is programmed to match the burst RAM address range for both read and write cycles, and the BCS address comparator is programmed to match the burst flash

address range. The BCS is also programmed to trigger on both BCS and CS7 matches.

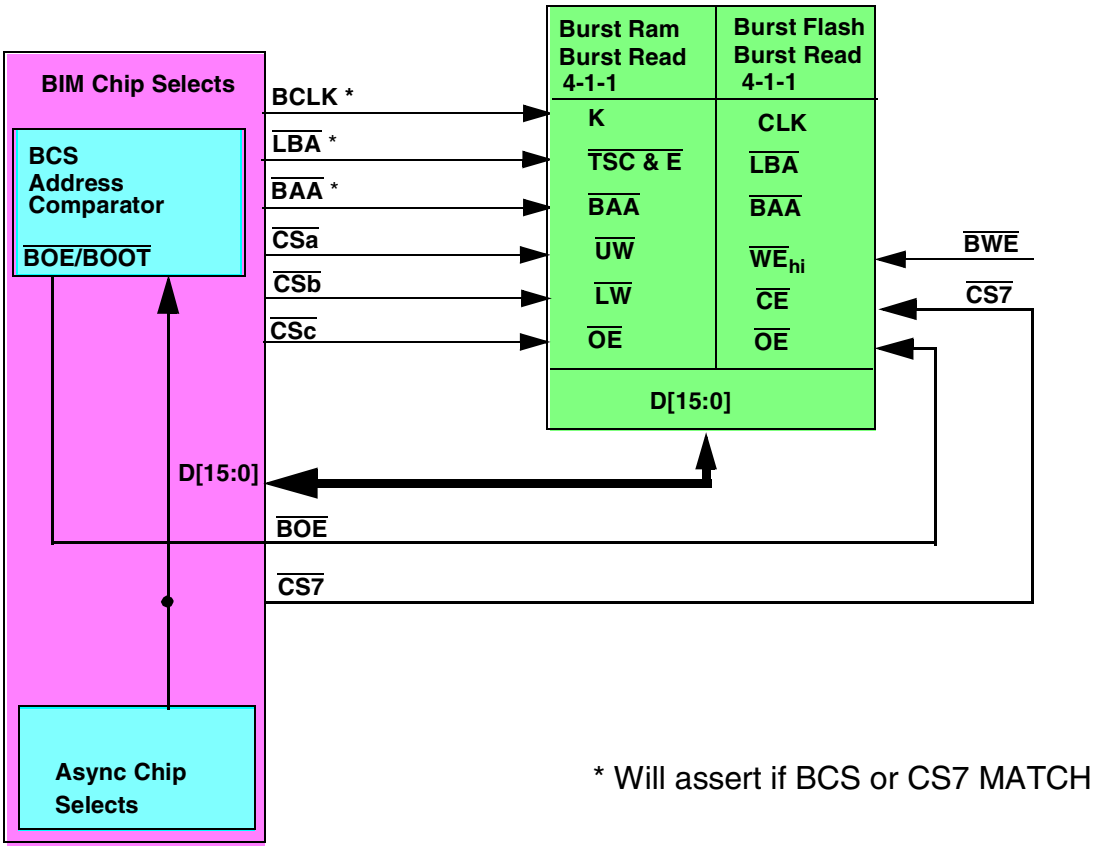


Figure 3-23 BCS Control of Two Like Type Memories, MT = 00

Figure 3-24 shows another two memory memory configuration consisting of an asynchronous NV memory and a non-burst synchronous SRAM. Using the pipelined protocol (MT = 10) in this application, it is possible to achieve high performance 2-1-1 bursts of lengths up to 32 words when using a high speed synchronous SRAM. This performance can be achieved because the BCS asserts LBA on every transfer along with a new address. A burst SRAM could also be utilized in this application, as long as its BAA input is tied high.

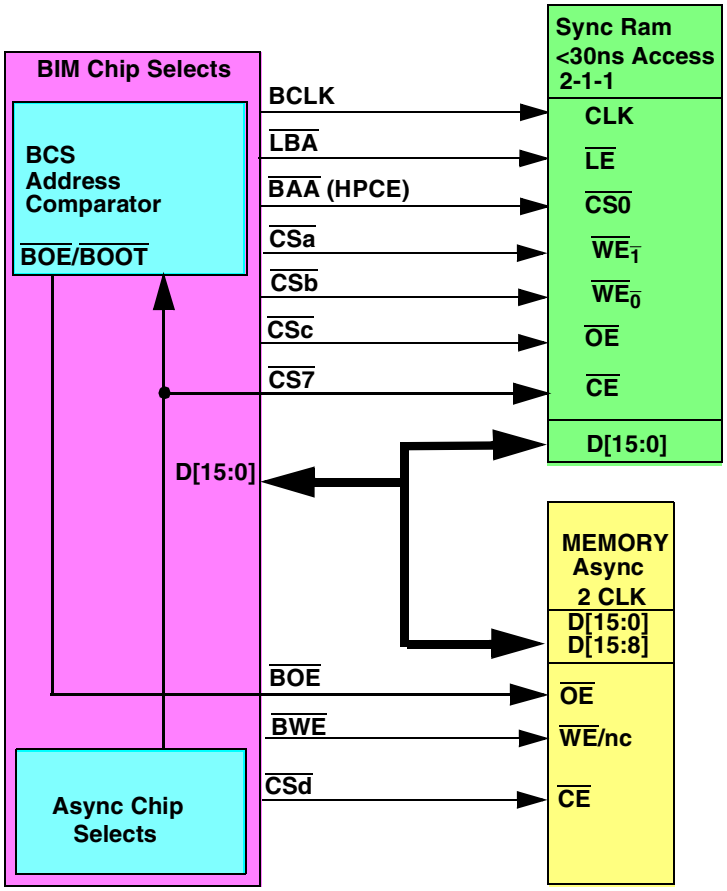


Figure 3-24 Sync SRAM in 2-1-1 Burst Read Application (MT = 10)

3.7.3 BCS Functional Units

The BCS is a full function programmable memory controller consisting of a base address register (BCSBAR), two option registers (BCSOR1, BCSOR2), and control/timing logic. BCSOR1 may be programmed to support

1. Asynchronous RAM, FLASH, and EPROM memories;
2. Burst FLASH, pipeline FLASH, and burst EPROM memories; and
3. Burst SRAM, BCSOR2 configures BCS pin functions.

The BCS contains its own $\overline{\text{DTACK}}$ and $\overline{\text{BTACK}}$ generators. These timing generators allow both the BCS and one standard CS ($\overline{\text{CS1}}$ — $\overline{\text{CS7}}$) to be active concurrently without causing conflicts in the respective $\overline{\text{DTACK}}$ timing generators. The BCS only supports external BTACK in memory type 01, but due to timing constraints, the BCS must be programmed with the same number of initial wait and burst data timing states as the external memory will generate. Any MCU cycle termination event causes the BCS to terminate the current memory access.

The BCS has default reset values in its registers to support CPU reset vector fetches during initialization. At reset, the BCS is configured for non burst read-only operation.

Figure 3-25 shows a block diagram of the BCS. Signals that generate memory control signals are taken from BIM internal buses.

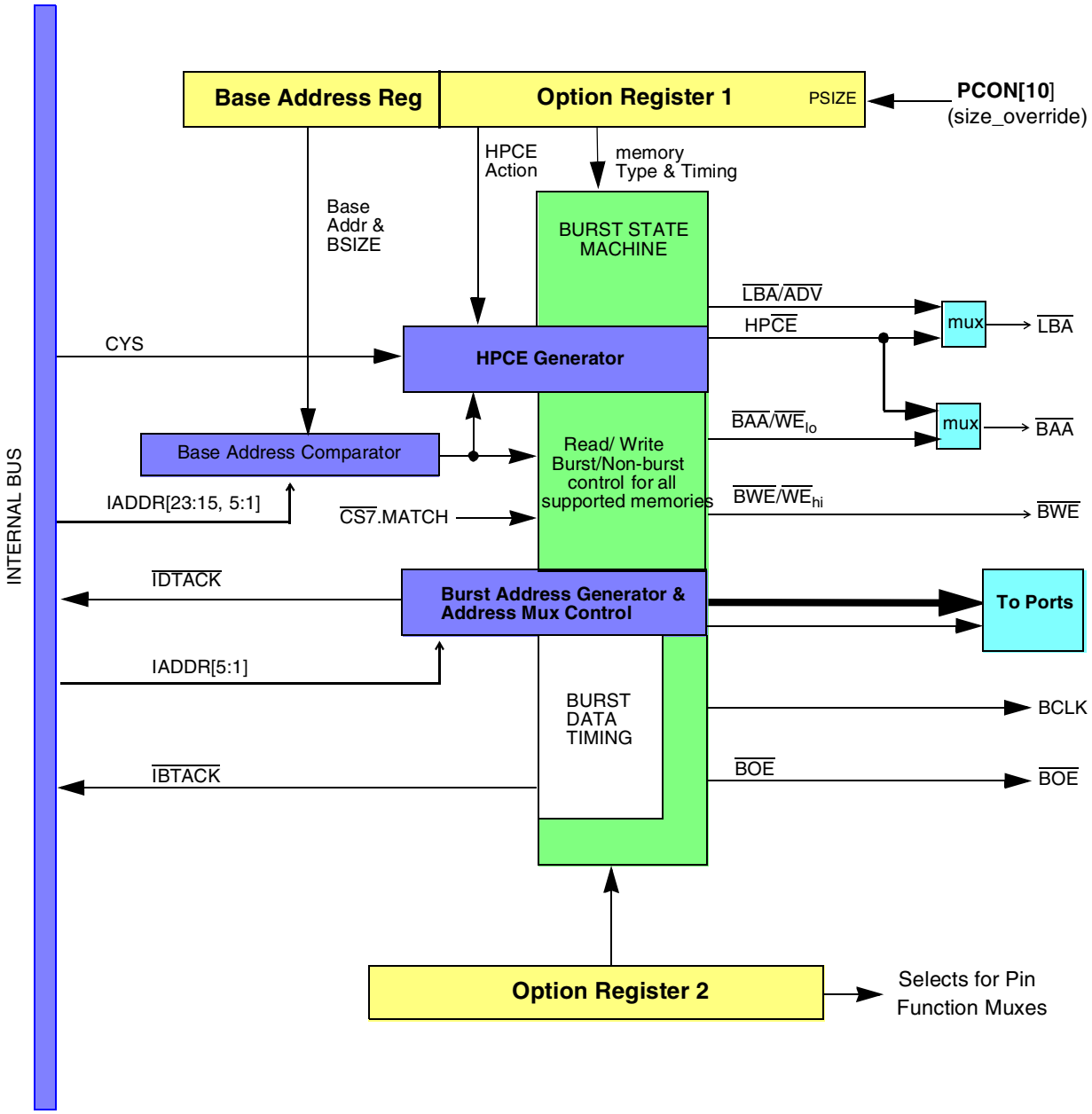


Figure 3-25 BCS Block Diagram

3.7.4 BCS Control Registers

The burst chip select registers are read/write from the IMB. Read operations on unused bits return zeros; writes to these bits do not have any effect. BCSBAR, BCSOR1 and BCSOR2 are reset at SRESET.

3.7.4.1 Base Address Register (BCSBAR)

The base address register consists of a base address field and block size field. The BASE ADDRESS is the starting address of an external memory to be enabled by the burst chip select. The BLOCK SIZE determines the extent of the address space from its base address.

To increase the usable BSC pin strobe window, BCS logic does not include the IAACKB signal (critical timing path) to indicate that an internal module plans to respond to an [internal] cycle. This requires that the address programmed in the base address register and/or the value of the SUPV bit in BCSOR1 be unique among all internal and external devices in the system to avoid executing an external BCS access on an internal cycle.

BCSBAR — Base Address Register **0xYF FA7C**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
BASE ADDRESS								RESERVED					BLOCK SIZE		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3-84 BCSBAR Bit Descriptions

Bit(s)	Name	Description
15:8	BASE ADDRESS	Base address. The base address field is contained in bits [15:8] of the base address register. In a supervisor/user space cycle, bits [15:8] are used to set the starting address of a particular address space. Address bits specified A[23:16].
7:3	—	Reserved
2:0	BLOCK SIZE	Block size. The address compare logic uses only the most significant bits to cause an address match within its block size, thus the value of the base address may be selected from any multiple of the block size to configure the address map in the application. **EXAMPLE**: Assuming a block size of 64 Kbytes, the comparator will compare only bits [15:8] of the base register with address lines A[23:16]. Therefore, each 64-Kbyte block starts on a 64-Kbyte boundary in the address map. 000 = 64 KB 001 = 128 KB 010 = 25 6KB 011 = 512 KB 100 = 1 MB 101 = 2 MB 110 = 4 MB 111 = 8 MB

3.7.4.2 BCS Option Register 1 (BCSOR1)

The option register 1 consists of individual fields which determine the timing and the conditions for asserting the BCS signals. Since option register 1 can be used to generate a number of different types of signals required to interface with external memories, care must be taken to program it properly for the desired access type.

A BCS MATCH is defined as a BCSBAR match and a SPACE match.

NOTE

Unlike asynchronous chip select (ACS) operation, memory size (MSIZE) and R/$\overline{W}$ are not option match parameters. Instead, these fields define whether the WE/OE pins(s) assert for the high byte, low byte, both bytes, or no bytes (R/$\overline{W}$ = 0 on a write cycle).

BCSOR1 — BCS Option Register 1 **0xYF FA7E**

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
BR		PRO-GRAM SPACE	R/$\overline{W}$	MSIZE	WHOLD	HP$\overline{CE}$		MT		MEMORY BOUNDARY		BDT	INITIAL TIMING		
RESET:															
1	1	1	PCON [10]	PCON [10]	0	1	1	0	0	0	0	1	1	1	1

Table 3-85 BCSOR1 Bit Descriptions

Bit(s)	Name	Description
15:14	BR	Burst response. This field specifies how the BCS will respond to IMB burst cycles that result in a MATCH. The programmable combinations allow the BCS to respond to IMB burst requests by providing full burst read/write operation, burst read/non-burst write operation, and non-burst read/write operation. This field must be programmed to match the characteristics of the memory type (MT) as shown in **Table 3-86**.
13	PROGRAM SPACE	Supervisor/user address space. The SPACE option is compared with the address space indicated by the function codes during each bus cycle. The burst chip select recognizes function codes 0-3 as user space and function codes 4-6 as supervisor space. The BCS does not respond to any CPU space cycles (FC = 7). When the SPACE bit is cleared, the BCS matches user space accesses only. When the SPACE bit is set, the BCS matches all address spaces except CPU space. 0 = User space. 1 = S/U
12	R/$\overline{W}$	Memory read/write. When cleared, the BCS generates $\overline{BOE}$ on read cycles; when set, the BCS generates both $\overline{BOE}$ and write strobes ($\overline{BWE}$ and $\overline{WE}_{lo}$) as specified by the memory size (MSIZE) and memory type (MT) fields. $\overline{LBA}$ and $\overline{BAA}$ are generated regardless of this setting. 0 = Read only 1 = R/$\overline{W}$
11	MSIZE	Memory size. This bit specifies the memory width. Eight- and 16-bit memory systems are supported in non-burst cycles. **All burst memory types require this field to be set to 16 bits (0)**. $\overline{LBA}$ and $\overline{BAA}$ are generated regardless of this setting. 0 = 16-bit 1 = 8-bit
10	WHOLD	Increase write hold time. On non-burst write cycles ($\overline{BREQ}$ negated or BR field = 1X if the R/$\overline{W}$ field = 1) the BCS will negate the $\overline{BWE}$ and $\overline{WE}_{lo}$ strobes when the initial time period expires. $\overline{DTACK}$ will not be issued for an additional clock to increase write hold time to slave memories. This bit must be set to zero for burst write operation.

Table 3-85 BCSOR1 Bit Descriptions (Continued)

Bit(s)	Name	Description
9:8	HP$\overline{CE}$	High performance/low power chip enable. Programming options allow the HP$\overline{CE}$ to remain asserted for four or eight bus transactions (internal or external) after the last valid BCS memory access, to be negated after every memory access (always forcing an extra wait state to be inserted at the beginning of the next BCS cycle by requiring a valid address/SUPV match before strobe generation), or to always assert. Each time a valid access (MATCH) to the memory is made, the $\overline{CE}$ time-out period is restarted. The effect of the HP$\overline{CE}$ logic on the BCS memory state machine is shown in **Figure 3-27**. For highest possible performance when the HP$\overline{CE}$ state machine is in state CE_ON, the BCS starts a memory access (for MT = 00, MT = 01 or MT = 10) in S0 by asserting $\overline{LBA}$ without regard to a MATCH on any MCU cycle [internal or external] to maximize the time available for external memory access. In S1, the memory access will be aborted ($\overline{BOE}$/$\overline{BWE}$ pins will not assert) if the address decode and cycle type is not a MATCH. When in the CE_OFF state, a valid BCS address and SUPV match is required before $\overline{CE}$ asserts, this adds an extra wait state to the initial access time to allow the memory to power-up before an access is started. **Figure 3-27** shows the effect of HP$\overline{CE}$ on timing. In the first cycle HP$\overline{CE}$ is initially negated, and then asserted at the start of the following bus cycle. If HP$\overline{CE}$ is asserted when LPSTOP executes, the HPCE signal is negated to power down any external memory controlled by HP$\overline{CE}$ and the bus cycle counter is cleared. The HP$\overline{CE}$ state machine remains in the CE_ON state so that when LPSTOP is exited, HP$\overline{CE}$ re-asserts and four or eight non-BCS cycles must occur before HPCE negates. 00 = Negate 4 bus cycles after last access 01 = Negate 8 bus cycles after last access 10 = Always negate after cycle 11 = Never negate
7:6	MT	Memory type. These bits specifies the external memory type. The BCS supports burst memories with internal address generation (memory type = 00), burst memory with internal address generation and BTACK generation (memory type = 01), pipeline burst memory (memory type = 10), and asynchronous memory (memory type = 11). The memory state machine for asynchronous memory, and burst memory protocol is shown in **Figure 3-28**. Memory type 01 supports burst memory architectures with no physical boundry limitations (continous burst) that control each data transfer utilizing the memory's internal BTACK generator. BAA is not a required control signal and can function as the HP$\overline{CE}$. When the memory transfers the last word in its output buffer, one or more wait states may be needed to reload the output buffer with the next page of data from the [memory] array before contining the burst transfer. In the current version of the BCS, the initial wait, burst data timing and memory boundary fields are still used by the BCS to determine when to terminate the bus cycle by asserting IDTACK. These fields should be set to their maximum values to utilize the longest burst data transfer supported by the BCS. Although the number of data transfers is determined by the memory access time and the wait states needed to reload the output buffer, the BCS supports a maximum burst length of 72 words with a 2-1-1 burst memory. 00 = Burst FLASH and SRAM 01 = Burst FLASH with external BTACK generation 10 = Pipeline FLASH 11 = Async FLASH and SRAM
5:4	MEMORY BOUNDARY	Memory boundary. These bits set the physical memory boundary for the burst address generator. Refer to **3.7.5.6 Burst Address Generator (BAG)** for a complete description of the BAG operation. 00 = 4 01 = 8 10 = 16 11 = 32

Table 3-85 BCSOR1 Bit Descriptions (Continued)

Bit(s)	Name	Description
3	BDT	Burst data timing. After the first word of a burst transaction has been transferred, the burst data timing bit specifies if a wait state is inserted between burst data transfers. If the burst data timing bit is cleared, the BCS does not insert any wait states between burst data transfers; if set, the BCS inserts a single wait state between data transfers (see **Figure 3-34**). If BDT is set and memory type = 00 or 01, the BAA waveform changes. If BDT is set and memory type = 10, the LBA waveform and BAG increment timing change. 0 = No waits between burst data. 1 = 1 wait between burst data
2:0	INITIAL TIMING	Initial access timing/$\overline{\text{DTACK}}$ generation. On non-burst cycles or when the burst response field is set to non-burst operation, the initial access timing field is used to specify the number of wait states to insert in the external cycle before the BCS terminates the cycle (asserts $\overline{\text{DTACK}}$). Setting the WRITE HOLD bit adds one additional wait state to write cycles than the initial timing field specifies. On burst cycles, the initial timing field specifies the amount of wait states that are inserted before the first word of a burst transaction is transferred (first $\overline{\text{BTACK}}$ generated). Since $\overline{\text{DTACK}}$ and $\overline{\text{BTACK}}$ are asserted internally after the programed number of wait states expires, the user can adjust the bus timing to accommodate the access speed of the external memory. With up to seven wait states, slow memories can be interfaced with the MCU. If any MCU event terminates the cycle ($\overline{\text{BERR}}$ or $\overline{\text{DTACK}}$) before the BCS generates $\overline{\text{DTACK}}$, the BCS terminates the memory transfer. The BCS does not allow external $\overline{\text{BTACK}}$ to be asserted during a burst memory transfer in the BCS memory range, except as discussed for MT = 01. 000 = 0 001 = 1 010 = 2 011 = 3 100 = 4 101 = 5 110 = 6 111 = 7

Table 3-86 Legal Burst Response and Memory Type Combinations

Burst Response Field	MT = 00	MT = 01	MT = 10	MT = 11
00	LBA asserts if HPCE is on, BAA, BOE and BWE are disabled.			Disabled
01	Supported operation	Supported operation	Not supported	Not supported
10	Supported operation	Supported operation	Supported operation	Not supported
11	Supported operation	Supported operation	Supported operation	Supported operation

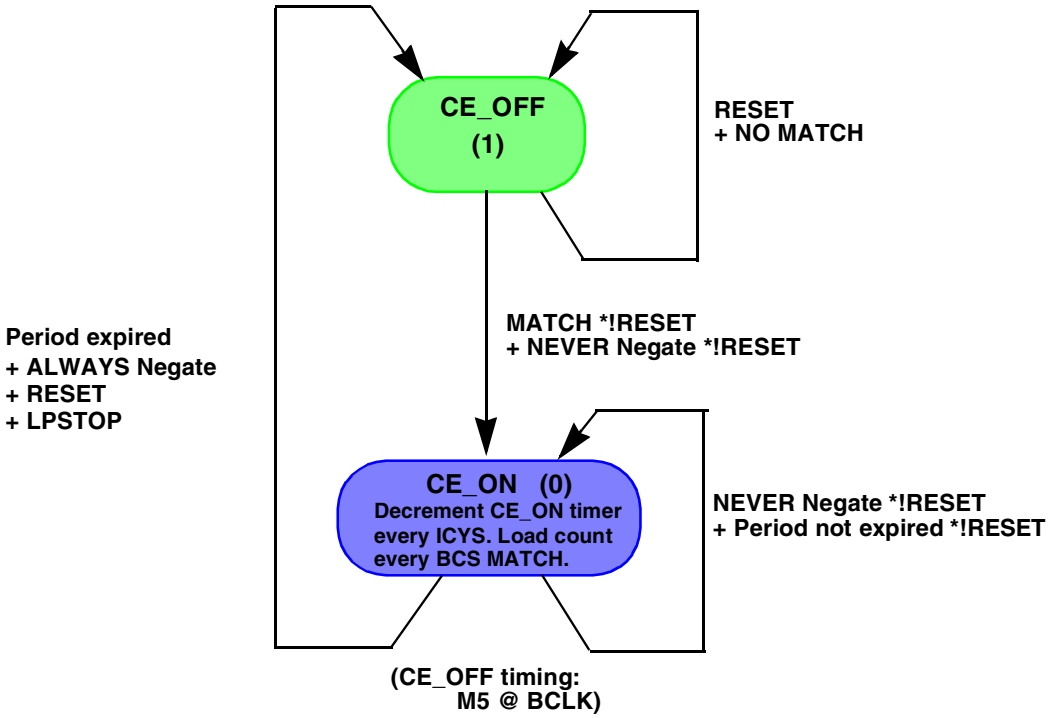


Figure 3-26 HP$\overline{CE}$ Functional Operation

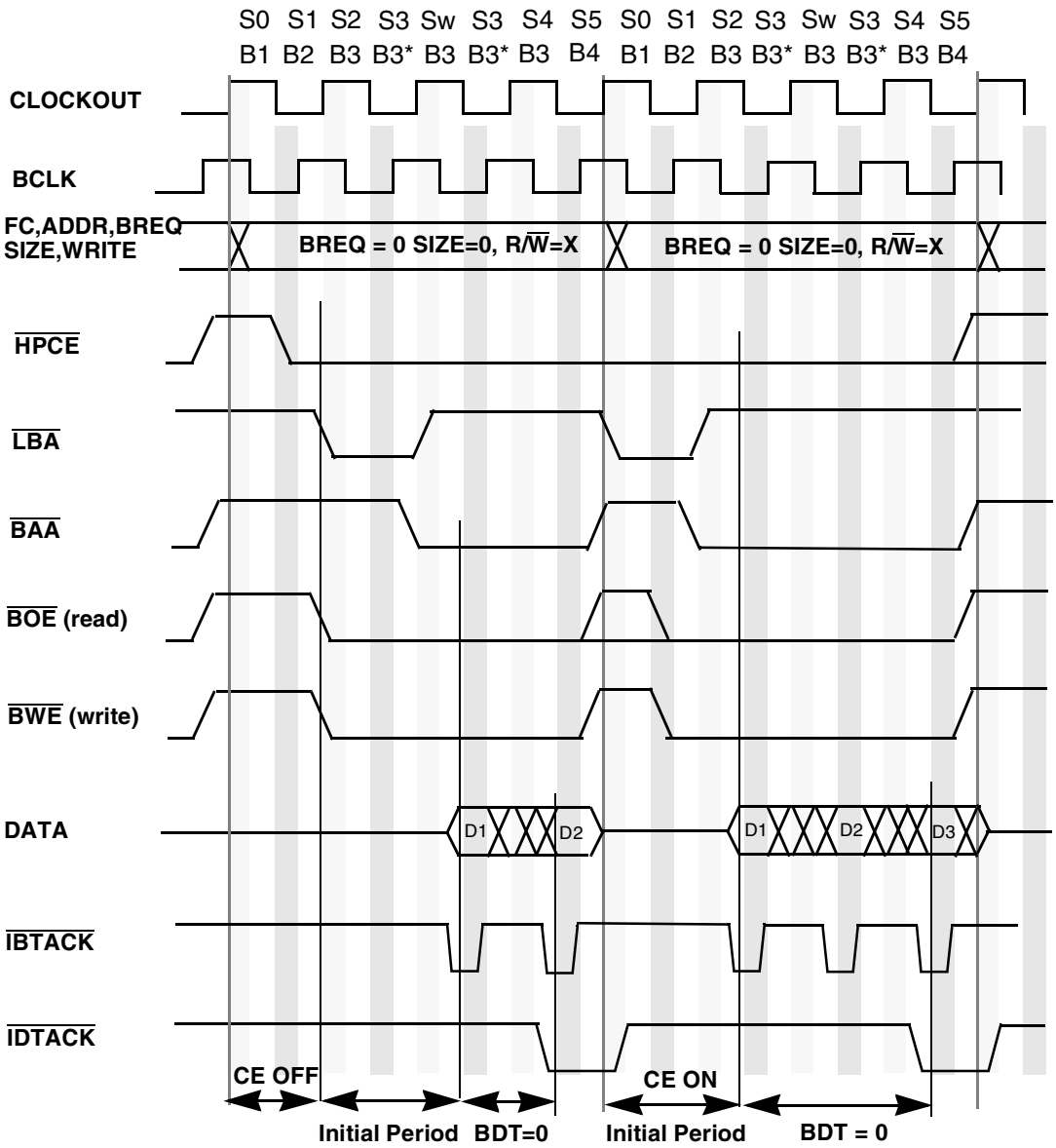


Figure 3-27 Effect of HP$\overline{CE}$ on BCS Operation, Initial Period = 0.

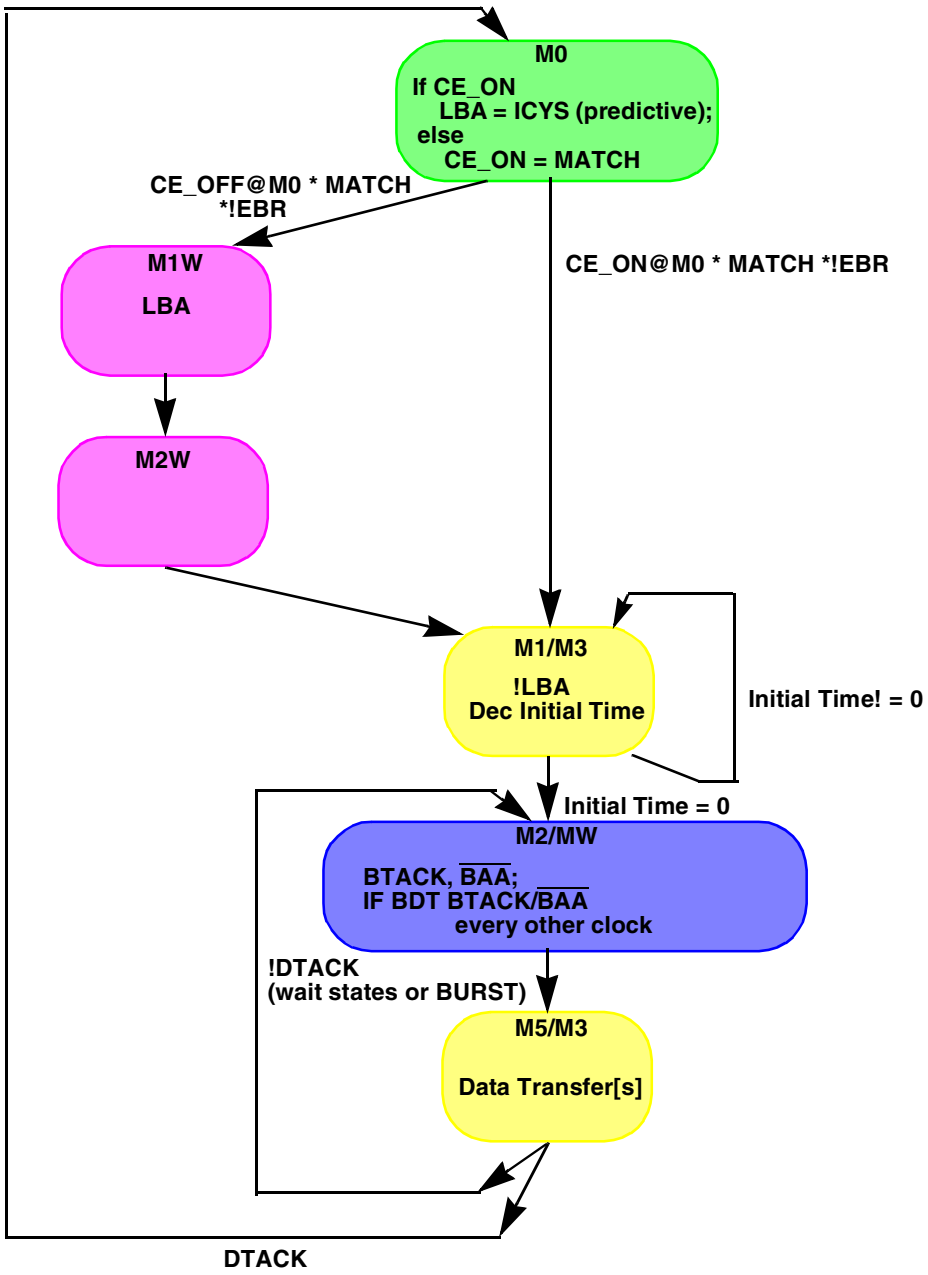


Figure 3-28 BCS State Machine for Burst, Pipeline, and Async Protocols

3.7.5 BCS Operation

3.7.5.1 RESET

During reset, BCLK is active, $\overline{BAA}$ and $\overline{BWE}$ are driven high. Any memory connected to the $\overline{BOE}$ override pin must be disabled (i.e., $\overline{CE}$ negated) during reset to avoid causing bus contention during the reset [PCON override] period.

3.7.5.2 Boot Memory Select

After reset, the burst chip select defaults to non-burst operation (burst response bits in option register 1 = 11). In this default mode the BCS is capable of accessing all supported memories.

In non-burst operation, the BCS supports 8- or 16-bit boot memories. The default memory size (MSIZE) is set by PCON[10] and may be overridden by the SIZE pin during reset. If MSIZE is set to 8 bits, the EBI will execute two [byte] cycles each time the IMB master requests a word transfer. $\overline{\text{LBA}}$ will be asserted during each byte cycle of a word transfer.

3.7.5.3 Non-Burst (Asynchronous) Operation

In this section, a non-burst memory access refers to a single data transfer per bus cycle to a synchronous or asynchronous memory. For synchronous memories, the memory type field in option register 1 (OR1) must be set to either 00, 01, or 10 depending on the specific protocol that is required. Asynchronous memories require the memory type field in OR1 to be programmed as 11. XXX through XXX show non-burst memory accesses for memory types 00, 01 and 10, while XXX shows access to an asynchronous memory (memory type 11).

When the burst response field of OR1 is programmed for non-burst operation, or during non-burst cycles ($\overline{\text{BREQ}}$ is negated), the BCS executes either an 8- or 16-bit non-burst memory access, depending on the programming of the MSIZ bit in OR1. When performing a non-burst memory access, the programming of the initial timing field of OR1 determines the number of wait states to be inserted in the external cycle before the BCS generates $\overline{\text{DTACK}}$. The memory boundary field, burst data timing field and BAG mode field are ignored by BCS logic during a non-burst access.

Figure 3-29 shows a 16-bit non-burst read access to a 16-bit memory with five wait states. If HPCE is on at the start of the memory access $\overline{\text{LBA}}$ will assert during S0 to load the current address into the memory; if HPCE is off an extra clock is added to the access to allow the memory to power up (assert CE). $\overline{\text{BOE}}$ will assert during S1, to enable the memory data drivers, and negates in the second half of S5. $\overline{\text{BAA}}$ and $\overline{\text{BWE}}$ remain negated throughout memory read access.

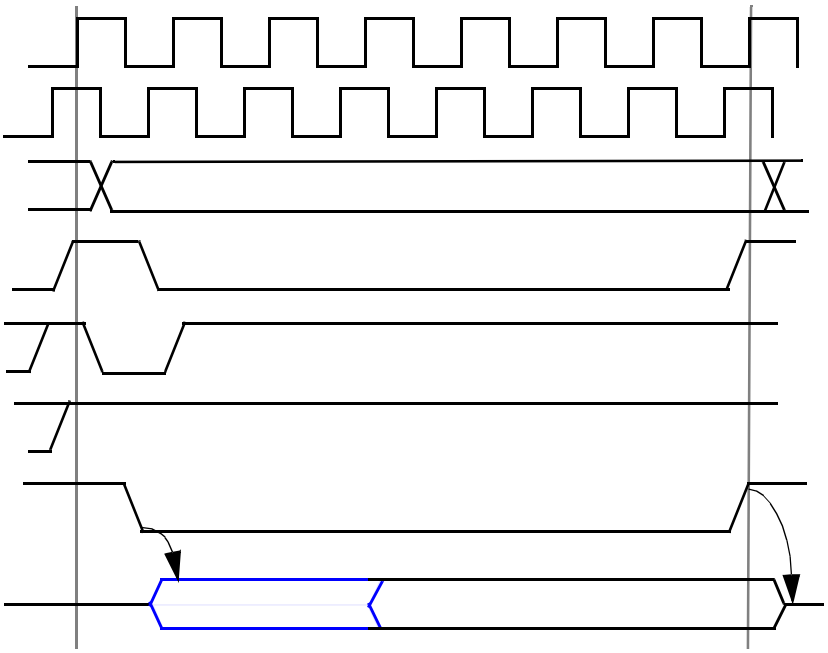


Figure 3-29 Non-Burst Read to 16-Bit Sync Memory

Figure 3-30 shows a 16-bit non-burst write access to a 16 bit memory, with five wait states. If $\overline{\text{HPCE}}$ is on at the start of the memory access, then $\overline{\text{LBA}}$ will assert during S0. $\overline{\text{BWE}}$ will assert during S1, if the write hold bit in OR1 is set to a zero, it will negate during the second half of S5. If the write hold bit is set to a one, an extra clock cycle is added to the number of wait states specified in the initial wait field and $\overline{\text{BWE}}$ will negate one clock before the cycle terminates (S3) to give a slow memory extra address and data hold time on the bus. $\overline{\text{BAA}}$ and $\overline{\text{BOE}}$ remains negated throughout a memory write access.

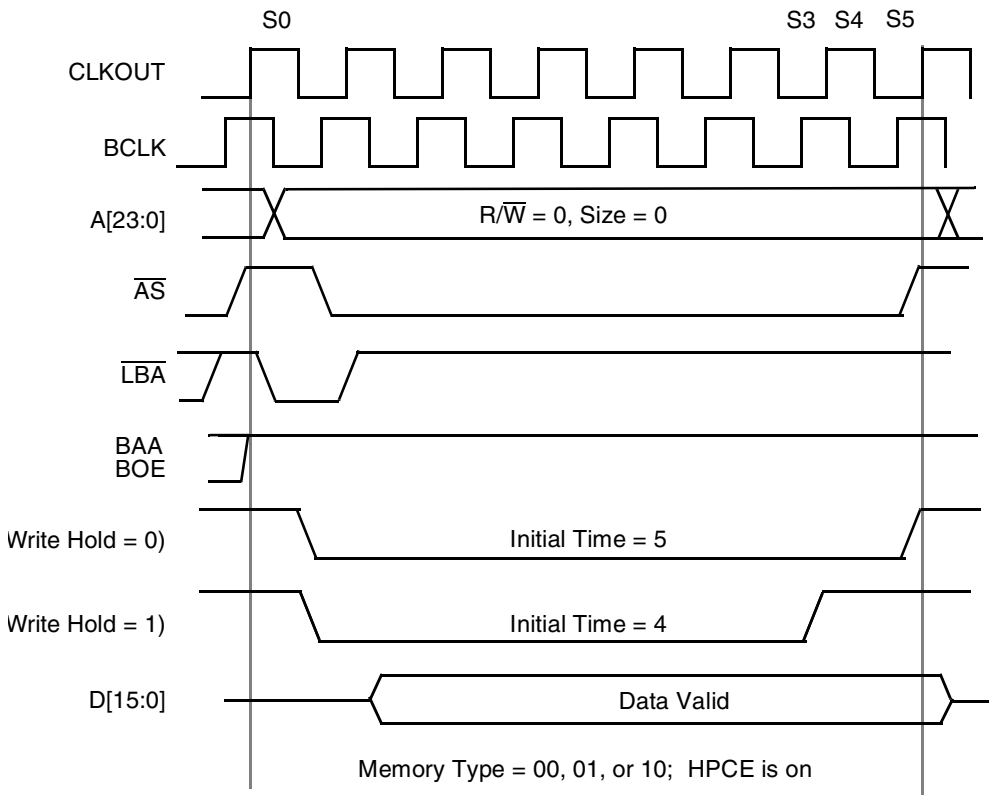
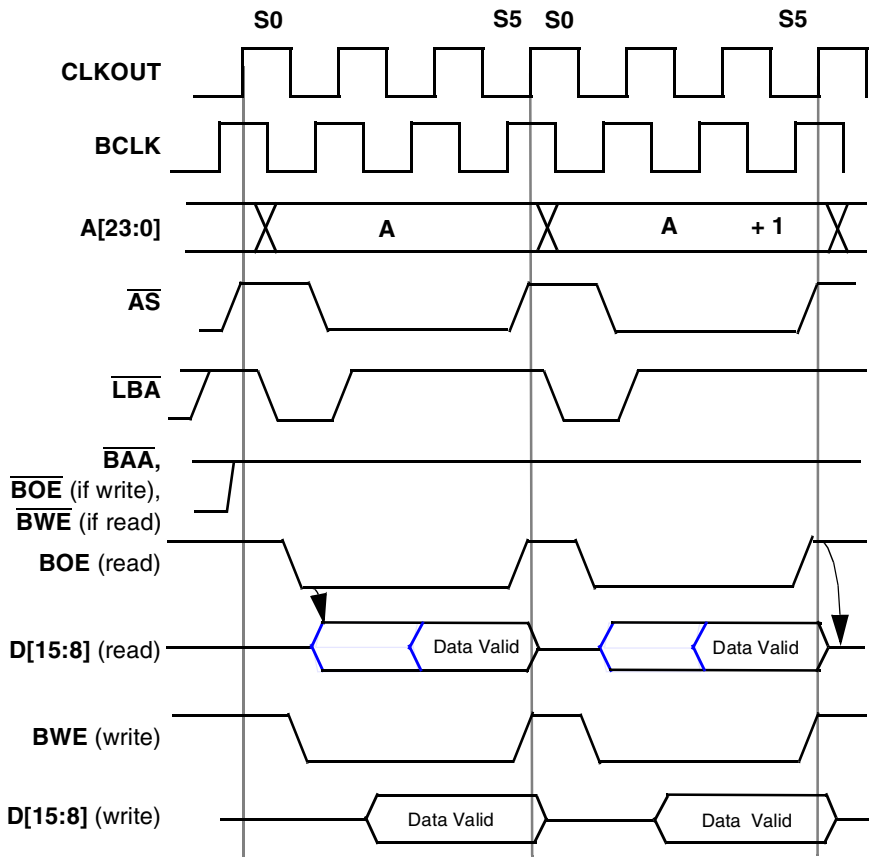


Figure 3-30 Non-Burst Write to 16-Bit Sync Memory

Figure 3-31 shows a 16-bit non-burst read and write access to an 8-bit memory. During the first memory access, A[0] = 0, and during the second A[0] = 1. For both read and write cycles, if HPCE is on at the start of the access, then LBA will assert for one clock during S0 of both byte accesses. During a memory read access, BOE will assert during S1 and will negate mid-S5. BAA and BWE will remain negated throughout the memory read access. During a memory write access, BWE will assert during S1 and will negate in either S3 or S5 depending on the setting of the write hold bit in OR1. BAA and BOE remain negated throughout the memory write cycle.



HPCE is on, Memory Type = 00,01,10, Write Hold = 0, Initial Wait = 1

Figure 3-31 16-Bit Non-Burst Cycle to 8-Bit Memory

When the memory type field in OR1 is set to 11, then the BCS will execute asynchronous read and write cycles. When using the asynchronous protocol to access [asynchronous] memories, $\overline{\text{LBA}}$ is not required, and can be used as $\overline{\text{HPCE}}$. $\overline{\text{BWE}}$ is the write enable for the high byte of a word ($\overline{\text{WE}}_{\text{hi}}$) and $\overline{\text{BAA}}$ functions as the write enable for the low byte of a word ($\overline{\text{WE}}_{\text{lo}}$).

Figure 3-32 shows a 16-bit asynchronous read followed by an 8-bit asynchronous write to the high byte (A[0] = 0) and an 8-bit asynchronous write to the low byte (A[0] = 1) (MSIZ = 0 in this figure). During a read cycle, $\overline{\text{BOE}}$ will assert when transferring data from the high byte, low byte or both bytes. During a write cycle $\overline{\text{BWE}}$ and/or $\overline{\text{BAA}}$ will assert depending whether the high byte, low byte or both bytes are being transferred. In the first write cycle in **Figure 3-32**, only the high byte of a word is being transferred, so only $\overline{\text{BWE}}$ asserts. In the second write cycle, only the low byte of a word is being transferred, so only $\overline{\text{BAA}}$ asserts. If both bytes are being transferred, then both $\overline{\text{BWE}}$ and $\overline{\text{BAA}}$ will assert.

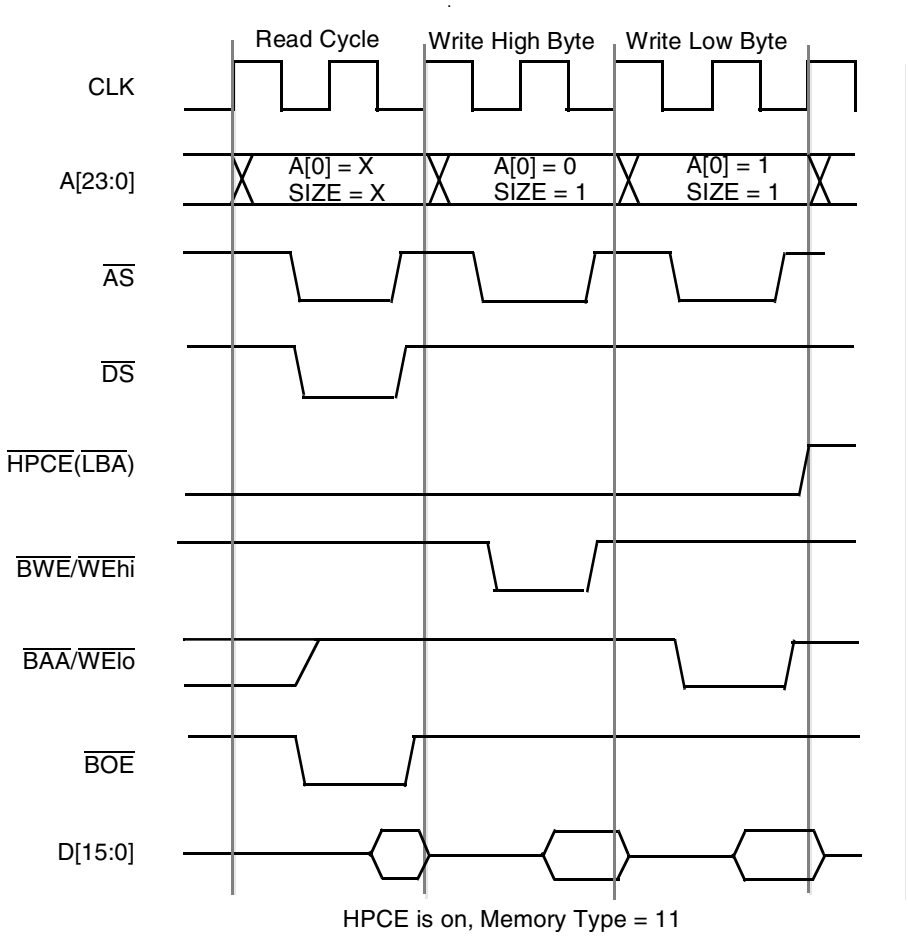


Figure 3-32 2 CLK Read/Write Cycles to Async Memory

3.7.5.4 Burst Operation

When the burst response field is programmed for burst operation and the match condition is satisfied, the BCS executes the memory access protocol as specified by the memory type field in OR1. The following programming restrictions apply for burst operation: the write hold bit must equal zero, MT cannot equal 11, BR must equal 10 or 01, and MSIZE must equal zero.

After waiting the INITIAL ACCESS TIMING wait states, internal $\overline{\text{BTACK}}$ and $\overline{\text{BAA}}$ assert to transfer the first data word between bus master and memory. If the BURST DATA TIMING bit indicates no wait states between data transfers, then internal $\overline{\text{BTACK}}$ and $\overline{\text{BAA}}$ remain asserted throughout the burst cycle; otherwise, internal $\overline{\text{BTACK}}$ and $\overline{\text{BAA}}$ toggle such that a wait state is inserted on each burst data transfer.

3.7.5.5 Burst Termination

The AMCU burst protocol allows a burst cycle to be terminated by either the bus master, or the slave. All burst cycles are terminated by the assertion of $\overline{\text{DTACK}}$ or $\overline{\text{BERR}}$.

DTACK can be generated by the burst address generator (BAG) or any DTACK/BERR source in the MCU.

A slave can terminate the burst cycle on the current data transfer by asserting DTACK if it cannot continue the burst. The IMB master may terminate the burst cycle on the current data transfer by negating BREQ. If the slave inserts wait states on this transfer, the cycle ends when the data is transferred (internal BTACK is asserted).

Depending on the BAG mode, the burst address generator terminates burst cycles based on the burst address or on the number of burst transfers.

3.7.5.6 Burst Address Generator (BAG)

The burst address generator (BAG), consists of two modulo 32 counters and logic to control the burst address mux (in Port A) to output the burst address on the external address bus A[5:1], generate each individual burst address during a burst cycle, and terminate the burst cycle based on the memory's internal architecture requirements.

The burst address (BA[5:1]) is brought out through the external address bus A[5:1] for burst memories requiring an [incremented] burst address for every data transfer. The BAG supports both program pre-fetch and cache line operation.

Table 3-87 BAG Termination Conditions

Memory Type	BAG Address Generation/Tracking Requirements	
	Pre-Fetch Operation	Cache Operation
00 - Burst FLASH / SRAM	Inc internal address@BTACK Terminate Cycle when BA[5:1] = MOD[1:0] value.	Inc modulo counter @BTACK Terminate cycle after MOD[1:0] data transfers.
10 - Pipeline FLASH / SRAM	Inc internal address@BTACK Gen external address@S2 ->A[] Terminate Cycle when BA[5:1] = MOD[1:0] value.	Inc internal address@BTACK Gen external address@S2 -> A[] Inc modulo counter @BTACK Terminate cycle after MOD[1:0] data transfers.
11 - Async FLASH / SRAM	None	None

3.7.5.7 Incrementing the Burst Address

The BAG latches the starting burst address and increments the burst address by one word every time internal BTACK is asserted.

3.7.5.8 BAG Cycle Termination Conditions

The BAG terminates the burst cycle based on the physical address or the number of burst transfers completed.

Table 3-88 BAG Termination Conditions

Memory Boundary	Burst Cycle Termination	
BCSOR1[5:4]	PRE_FETCH Mode	Cache Mode
00 = 4	BA[2:1] = 0x3	After 4 data transfers.
01 = 8	BA[3:1] = 0x7	After 8 data transfers
10 = 16	BA[4:1] = 0xF	After 16 data transfers
11 = 32	BA[5:1] = 0x1F	After 32 data transfers

In pre-fetch mode operation (BAG MODE = 0), if the burst cycle accesses memory location (2^N-1), the BCS terminates the burst by asserting $\overline{\text{DTACK}}$ internally. This action forces the bus master to restart the burst operation on the next bus cycle, starting at address 2^N. For burst memories with no physical boundary restrictions, this field is programmed to %11 to select the maximum burst length supported by the BCS.

In cache mode operation (BAG MODE = 1), the BAG terminates the burst transaction on the 2^N data transfer regardless of the burst address.

3.7.5.9 Burst Waveforms

Figure 3-33 and **Figure 3-34** show several different waveforms for the three burst protocols that the BCS supports. These figures show the affect of the programming of the initial timing and BDT fields in OR1 and the HPCE circuit on the waveforms produced on the $\overline{\text{LBA}}$, $\overline{\text{BAA}}$, $\overline{\text{BOE}}$ and $\overline{\text{BWE}}$ pins.

Figure 3-33 through **Figure 3-38** apply to both MT = 00 and MT = 01, but the external $\overline{\text{BTACK}}$ assertion at the bottom of these figures only applies to MT = 01. **The assertion of external $\overline{\text{BTACK}}$ in memory types 00, 10 or 11 causes abnormal BCS operation.**

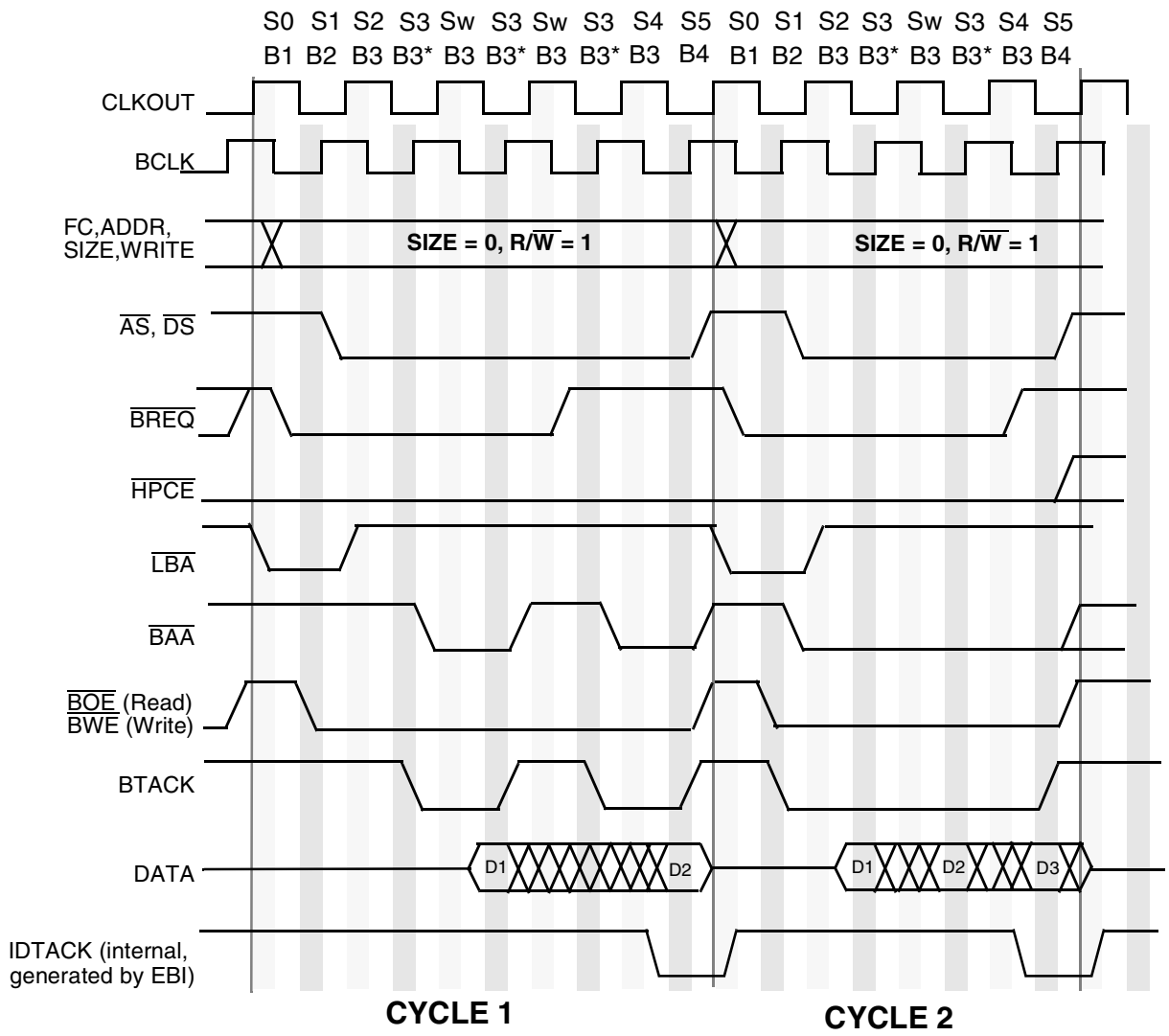
Figure 3-33 demonstrates the burst read protocol assuming that the $\overline{\text{HPCE}}$ circuit is on at the start of cycle #1. Cycle #1 also assumes both the initial wait and the BDT fields in OR1 were programmed to one, thus inserting one wait state in the initial access time, and one wait state between data transfers. This is known as a 3-2-2 burst cycle. Cycle #2 assumes that both the initial wait and the BDT fields in OR1 were programmed to zero, thus producing a burst read cycle with no wait states. This is know as a 2-1-1 burst cycle.

When $\overline{\text{HPCE}}$ is on at the start of a cycle, then $\overline{\text{LBA}}$ asserts by mid-S0 and negates one clock later. When the BAG counts the number of clocks programmed in the initial wait field of OR1, it will generate IBTACK and $\overline{\text{BAA}}$ will be asserted. If BDT = 1, then $\overline{\text{BAA}}$ will negate 1 clock after it asserts, as in cycle 1. If BDT = 0, then $\overline{\text{BAA}}$ remains asserted through the entire cycle, as in cycle 2. When $\overline{\text{BREQ}}$ negates during a burst cycle, the EBI terminates the bus cycle on the next data transfer (i.e., BTACK assertion). This is known as master terminated burst. In master terminated burst cycles, EBI logic is responsible for asserting IDTACK when BTACK asserts.

The waveform for $\overline{\text{BTACK}}$ at the bottom of shows how the external memory would need to assert and negate the $\overline{\text{BTACK}}$ pin in MT = 01 to produce the same waveforms on $\overline{\text{BAA}}$ and IBTACK as the BAG produces in MT = 00.

Figure 3-33 demonstrates burst read protocol with HPCE off at the start of cycle 1. In cycle 1 the initial wait field is programmed as one and the BDT field is programmed to 0, thus inserting one wait state in the initial access time, and no wait states between data transfers. This is known as a 3-1-1 burst cycle. In cycle 2 both the Initial Wait and the BDT fields are zero.

If $\overline{\text{HPCE}}$ is off at the start of a bus cycle, the BCS will assert $\overline{\text{HPCE}}$, and will wait one clock before asserting $\overline{\text{LBA}}$. This is to allow time for the external memory to power-up before the address is loaded into it. In comparing the assertion of $\overline{\text{LBA}}$ in cycle 1, to cycle 1, note that $\overline{\text{LBA}}$ asserts in the same clock state as $\overline{\text{AS}}$ instead of 1 clock before $\overline{\text{AS}}$. Once $\overline{\text{LBA}}$ asserts, the rest of the bus cycle progresses as described above. Since HPCE does not negate at the end of cycle 1, cycle 2 executes the same as cycle 2 in .



Cycle1: HPCE is on, Initial Timing = 1, BDT = 1 Cycle2: Initial Timing = 0, BDT = 0

NOTE: If the master terminates the burst cycle, the BIM asserts internal DTACK (IDTACK) when the [last] BTACK asserts.

Figure 3-33 Burst Read Protocol, (MT = 00 or 01)

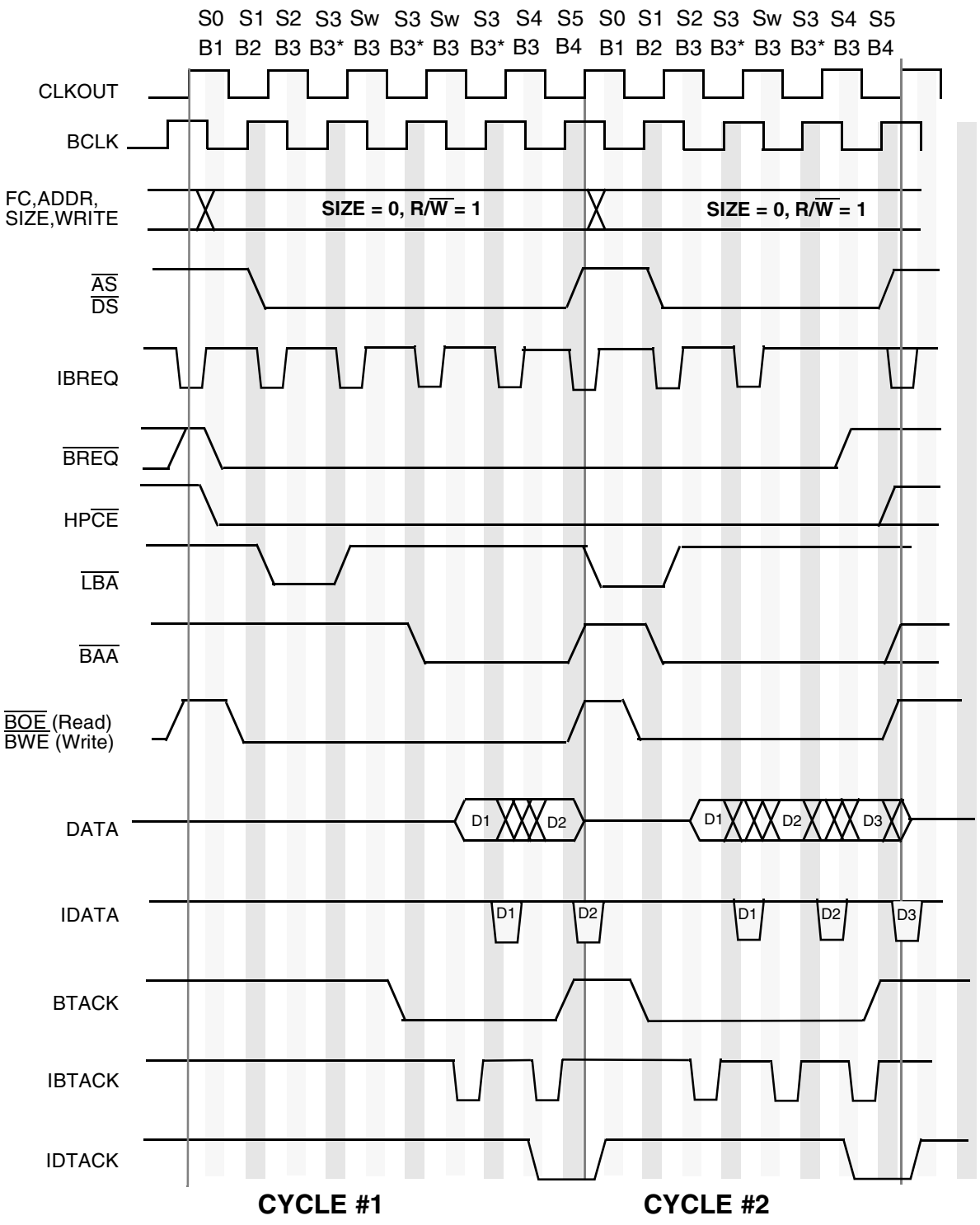


Figure 3-34 Burst Read Protocol, (MT = 00 or 01)

Cycle #1: Initial Timing = 1, BDT = 0, HPCE is initially off
Cycle #2: Initial Timing = 0, BDT = 0

The pipelined protocol (MT = 10) is intended for external memories which must load a new address for each beat of data it is to provide. See **Figure 3-35** for burst waveforms for the pipelined protocol (MT = 10).

Figure 3-35 demonstrates the pipelined protocol, assuming that the $\overline{\text{HPCE}}$ circuit is on at the start of cycle #1. Cycle #1 also assumes both the initial wait and the BDT fields in OR1 were programmed to one, thus inserting one wait state in the initial access time, and one between beats of data. Cycle #2 assumes that both the Initial Wait and the BDT fields in OR1 were programmed to zero, thus producing a burst read cycle with no wait states.

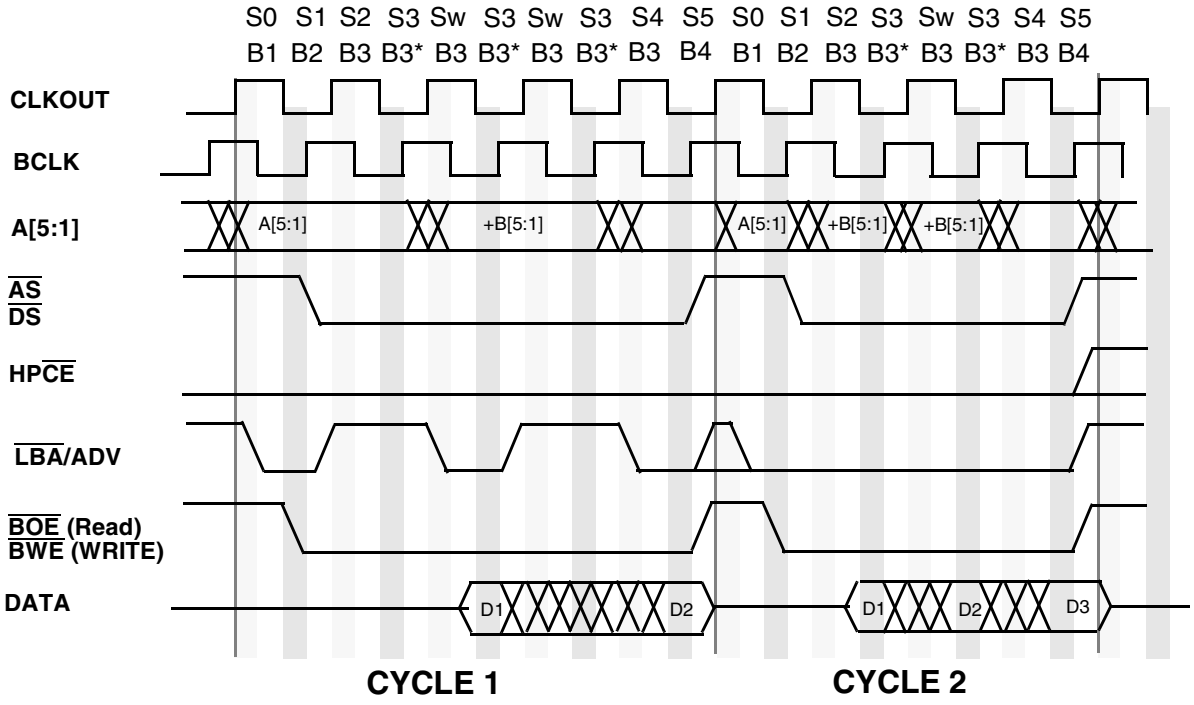
In cycle #1, $\overline{\text{LBA}}$ asserts at the start of the cycle to load the initial address into the external memory. Since BDT = 1 in this case, $\overline{\text{LBA}}$ will negate one clock later to prevent a new address from being loaded into the external memory on the next rising edge of BCLK. With BDT = 1, $\overline{\text{LBA}}$ will toggle every BCLK until the cycle terminates. Also note that when BDT = 1, the BAG will increment A[5:1] every other BCLK, instead of every BCLK, as demonstrated in cycle #2 (where BD T =0). In pipelined mode, the BAG still counts the Initial Timing number of clocks before generating IBTACK, and will add one wait state between beats of data if BDT is set.

Figure 3-36 also demonstrates the pipelined protocol, this time with the initial timing field set to two, and the BDT field set to zero. In this example, $\overline{\text{LBA}}$ asserts at the beginning of the bus cycle, and negates at the end of the bus cycle so that a new address in loaded into the external memory on every rising edge of BCLK. The only difference between cycle #2 in **Figure 3-36** and cycle #1 in **Figure 3-37** is then number of wait states that are inserted before the first beat of data is transferred. Both cycle #2's are identical.

Figure 3-37 demonstrates the pipelined protocol, but with $\overline{\text{HPCE}}$ off at the start of the bus cycle. As in the burst protocol, when $\overline{\text{HPCE}}$ is initially off, the BCS will assert $\overline{\text{HPCE}}$, and will insert a wait state before the assertion of $\overline{\text{LBA}}$. Once $\overline{\text{LBA}}$ is asserted, the rest of the bus cycle will progress as in **Figure 3-38**.

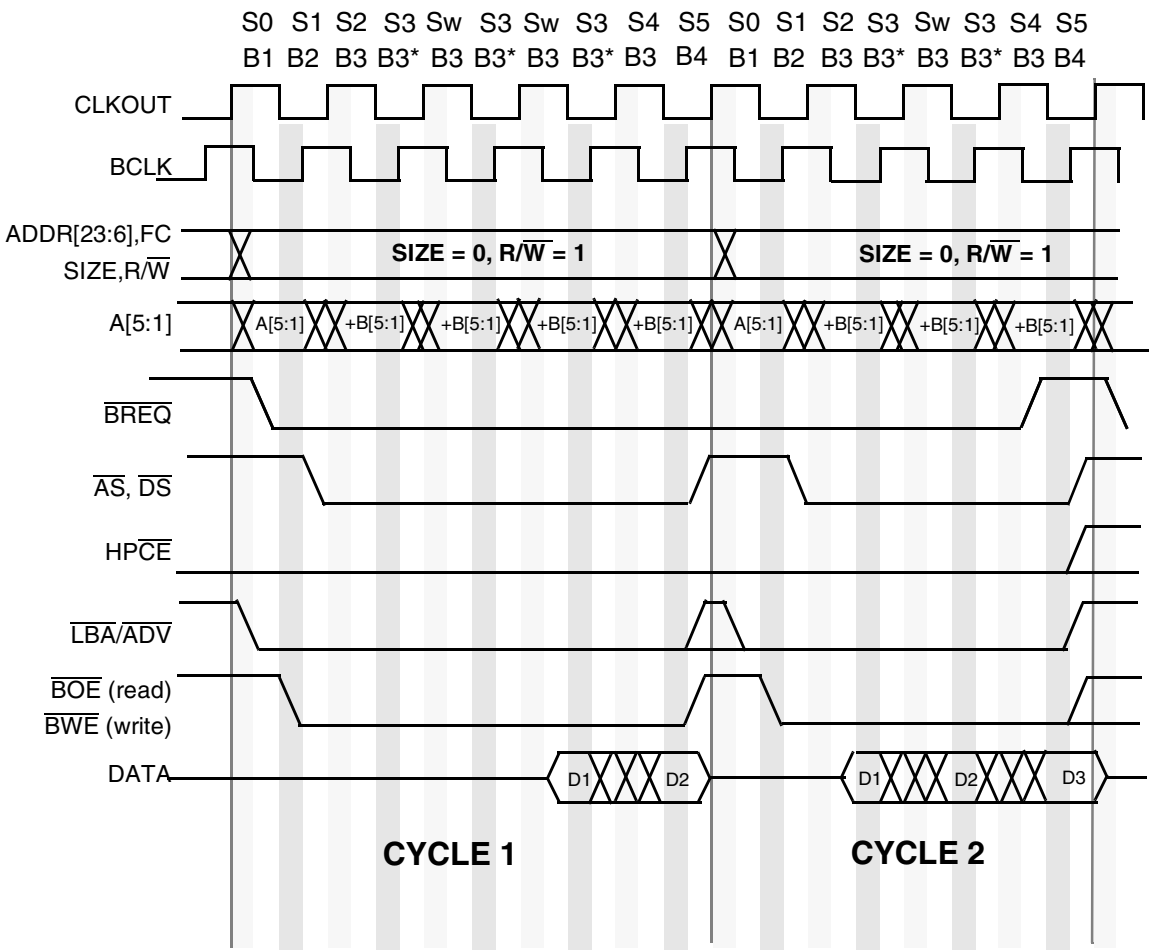
NOTE

When $\overline{\text{HPCE}}$ is off at the start of a bus cycle, the initial address will be valid for two clocks instead of just one.



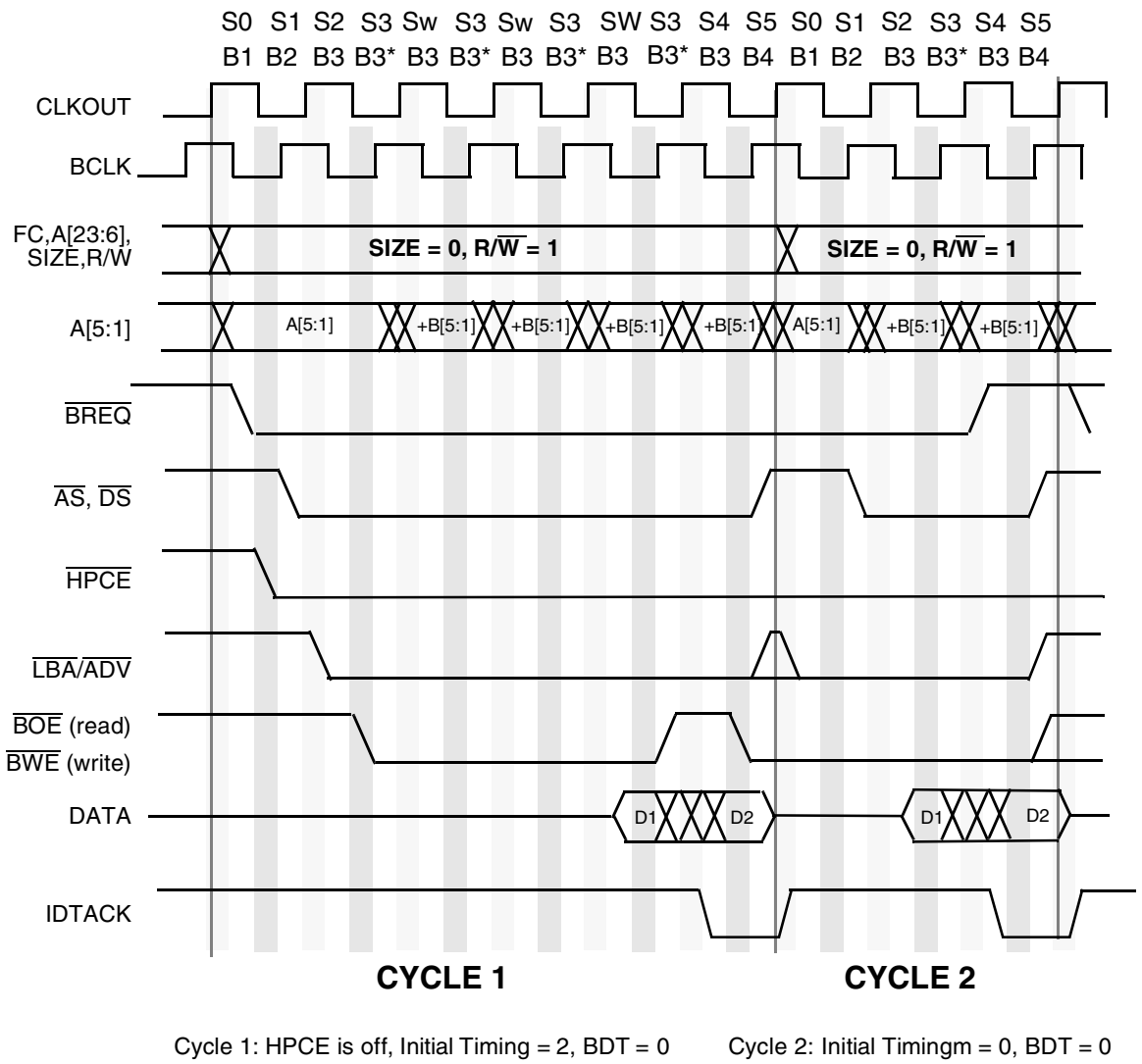
Cycle 1: Initial Timing = 1, BDT = 1, HPCE is initially on.
Cycle 2: Initial Timing = 0, Burst Data Timing = 0

Figure 3-35 Pipelined Read Protocol (MT = 10)



Cycle 1: Initial Timing = 2, BDT = 0, HPCE is initially on.
Cycle 2: Initial Timing = 0, BDT = 0

Figure 3-36 Pipelined Read Protocol (MT = 10)



Cycle 1: HPCE is off, Initial Timing = 2, BDT = 0

Cycle 2: Initial Timingm = 0, BDT = 0

Figure 3-37 Pipelined Read Protocol (MT = 10)

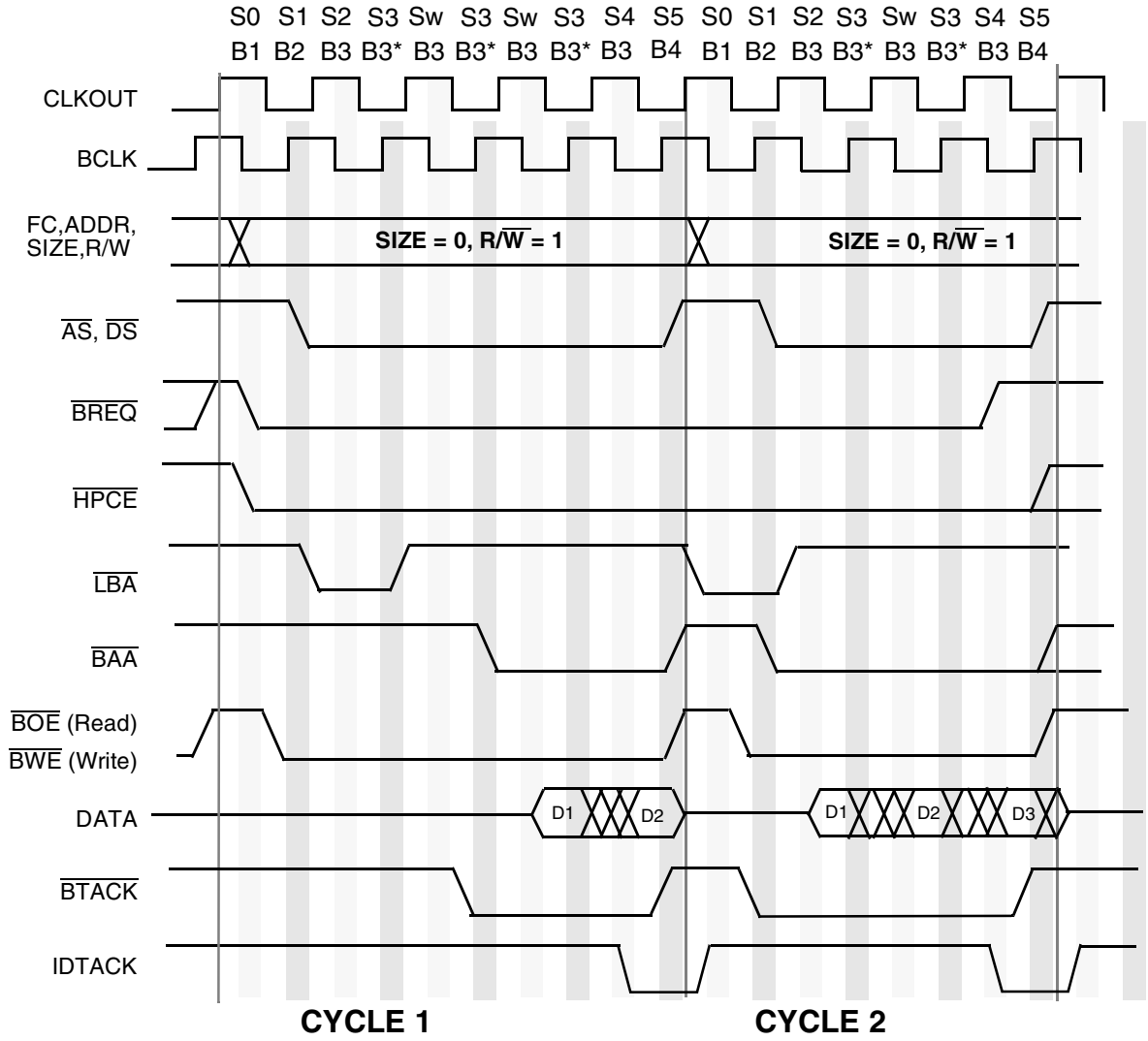


Figure 3-38 Burst Read Protocol, (MT = 00 or 01)

3.8 External Bus Interface (EBI)

The external bus interface (EBI) is the section of the BIM responsible for controlling the transfer of information between the IMB and external address space.

The EBI functions as a bus master or as a slave, depending on the BIM operation mode. In some BIM operation modes the EBI is not active. In master mode, emulation mode the EBI operates as the external bus master and allows internal bus cycles to access external resources. The EBI also gives limited support to alternate external masters with the external bus request ($\overline{EBR}$) signal. Refer to **3.8.7.1 External Bus Arbitration** for more details.

The EBI supports data transfers to both 16-bit and 8-bit devices. The chip select channels are programmed to define the device size for specific address ranges. When no chip select is active during an external data transfer, the device size is assumed to be 16 bits.

The EBI supports a variable length external bus cycle to accommodate the access speed of any device. During an external data transfer, the EBI drives the address, function codes, size and read/write information. $\overline{AS}$ signals the start of the external bus cycle, and $\overline{DS}$ signals valid data on a write cycle. Wait states are inserted until the bus cycle is terminated by the assertion of the internal DTACK signal by a chip select channel, or by the assertion of the external $\overline{DTACK}$ or $\overline{BERR}$ pins. For more details on chip select programming, refer to **3.6 Asynchronous Chip Select (ACS)**.

The minimum external bus cycle is two clocks. Termination of two clock bus cycles is hard to achieve because of $\overline{AS}$ timing. Typically, two clock bus cycles are terminated using a chip select to generate DTACK internally. The minimum 2 clock external bus cycle can also be achieved by asserting the $\overline{DTACK}$ pin or by asserting the $\overline{BERR}$ pin as specified by the AC timing requirements for two cycle termination. Generation of external $\overline{DTACK}$, $\overline{BTACK}$, or $\overline{BERR}$ based solely on addresses decoding without $\overline{AS}$ qualification will result in valid cycle termination since the termination signal(s) are not recognized by the EBI until the correct time in the external cycle (after addresses have been valid for at least one half clock).

An internal data transfer is terminated by the BIM or other IMB module. Internal cycles cannot be terminated by a chip select channel, external $\overline{DTACK}$ assertion, or external $\overline{BERR}$ assertion. Internal cycles may or may not have wait states.

In slave access mode (SLAM), the EBI operates as a slave interface to the external bus, allowing external bus cycles to access internal resources. Slave accesses are typically to internal memory locations. If a slave access is to an external address, the EBI does not present the bus cycle on the external bus. However, other BIM submodules, like the chip selects, respond normally. Alternate bus masters are not supported via $\overline{EBR}$ in this mode.

In single chip mode, the EBI is not active. All data transfers are between internal modules. $\overline{EBR}$ is meaningless in this mode because the external bus is never driven by the EBI, and show cycles are not supported.

Background debug mode is supported by the BIM in master mode, emulation mode, and single chip mode. The EBI operation is unaffected by background debug mode. Background debug mode operation in SLAM is not defined.

3.8.1 Security Mode

The BIM provides a security feature which disables all external reads of internal memory and registers, showcycle monitoring of internal bus activity, and visibility data bus activity. Security mode is enabled during reset by the state of a mask ROM bit. Security mode is enabled if the ROM bit is programmed as a 1. If enabled, security mode disables background debug mode, show cycles in master modes, external read and

write cycles in factory test modes and the visibility data bus. For testing purposes and authorized customer access, security mode can be disabled.

3.8.2 Bus Master Operation

3.8.3 Operand Transfer

From a CPU point of view, the possible operand accesses are: even byte, odd byte, even word, odd (misaligned) word, even long-word, and odd (misaligned) long-word. The even byte, odd byte, and even word accesses appear directly on the IMB. An even long-word access is presented to the IMB as two even word accesses.

The CPU32 does not support misaligned accesses and takes an exception. So, there are only four IMB access cases that must be handled by the BIM: even byte, odd byte, even word, and odd word. See **Table 3-89** for a listing of all supported transfer cases.

Table 3-89 MCU Interface to Various Port Sizes

Operand				
	OP0	OP1	OP2	OP3
31		OP0	OP1	OP2
	23		OP0	OP1
			15	OP0
				7 0
				Data Bus

Case	Transfer Case	SIZ1	SIZ0	A0	DSACK1	$\overline{\text{DSACK0}}$	D15 D8	D7 D0
a	Byte to byte	0	1	X[1]	1	0	OP0	(OP0)[2]
b	Byte to word (even)	0	1	0	0	X	OP0	(OP0)
c	Byte to word (odd)	0	1	1	0	X	(OP0)	OP0
d	Word to byte (aligned)	1	0	0	1	0	OP0	(OP1)
e	Word to byte (misaligned)[3]	1	0	1	1	0	OP0	(OP0)
f	Word to word (aligned)	1	0	0	0	X	OP0	OP1
g	Word to word (misaligned)[3]	1	0	1	0	X	(OP0)	OP0
h	3-Byte to byte (aligned)[3,4]	1	1	0	1	0	OP0	(OP1)
i	3-Byte to byte (misaligned)[3,4]	1	1	1	1	0	OP0	(OP0)
j	3-Byte to word (aligned)[3,4]	1	1	0	0	X	OP0	OP1
k	3-Byte to word (misaligned)[3,4]	1	1	1	0	X	(OP0)	OP0
l	Long-word to byte (aligned)	0	0	0	1	0	OP0	(OP1)
m	Long-word to byte (misaligned)[3]	0	0	1	1	0	OP0	(OP0)
n	Long-word to word (aligned)	0	0	0	0	X	OP0	OP1
o	Long-word to word (misligned)[3]	0	0	1	0	X	(OP0)	OP0

NOTES:
1. X in a column means that the state of the signal has no effect.
2. Operands in parentheses are ignored by the CPU during read cycles.
3. Misaligned transfer cased not supported.
4. Three-byte transfer cases occur only as a result of an aligned long-word to byte.

The EBI controls the byte or word operand transfers between the IMB and 8-bit or 16-bit devices. Each device is assigned to particular bits of the data bus. A 16-bit device is assigned to data bus bits [15:0] and an 8-bit device is assigned to data bus bits [15:8]. Depending on the device size, the EBI may run more than one external bus cycle to complete the operand transfer. Signals issued by the EBI control data movement on the external bus.

In the operand transfer cases that follow, the EBI is the exernal bus master. The upper portion of each figure shows the operand for the current cycle. Its position indicates the location of the data on the IMB data bus. OP0 is the most significant byte, if the operand contains two bytes of data. For each external bus cycle performed, “OPx” indicates where operand byte “x” is driven on the 16-bit external data bus.

3.8.3.1 Byte Operand, 8-Bit Device, Even (A0 = 0)

For a byte transfer between an even memory location and an 8-bit device, the EBI drives the external address bus with the desired address and drives the SIZE pin high to indicate a byte operand transfer. In the case of a read, the slave device responds by placing the data on bits [15:8] of the data bus. In the case of a write, the EBI drives the single byte operand on bits [15:8] of the external data bus.

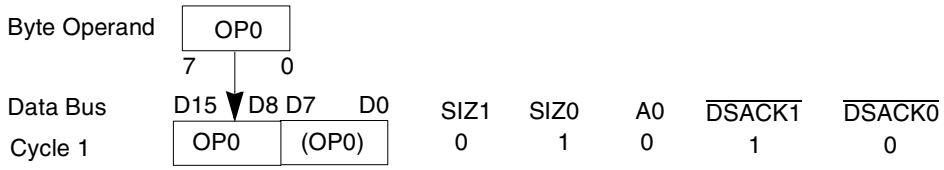


Figure 3-39 Even Byte Transfer to 8-Bit Device

3.8.3.2 Byte Operand, 8-Bit Device, Odd (A0 = 1)

For a byte transfer between an odd memory location and an 8-bit device, the EBI drives the external address bus with the desired address and drives the SIZE pin high to indicate a byte operand transfer. In the case of a read, the slave device responds by placing the data on bits [15:8] of the data bus. In the case of a write, the EBI drives the single byte operand on bits [15:8] of the external data bus.

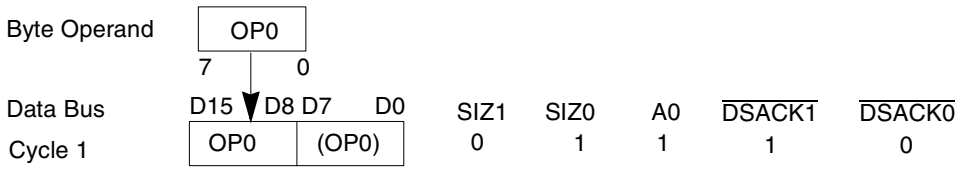


Figure 3-40 Byte Operand to 8-Bit Device, Odd (A0 = 1)

3.8.3.3 Byte Operand, 16-Bit Device, Even (A0 = 0)

For a byte transfer between an even memory location and a 16-bit device, the EBI drives the address bus with the desired address and drives the SIZE pin high to indicate a byte operand transfer. In the case of a read, the slave device places the data on bits [15:8] of the data bus. The EBI ignores bits [7:0]. In the case of a write, the EBI drives the single byte operand on bits [15:8] of the data bus. Bits [7:0] are not driven.

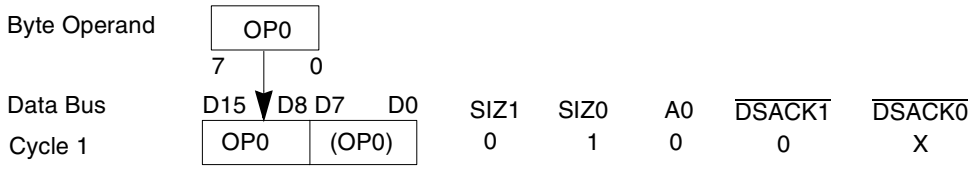


Figure 3-41 Byte Operand to 16-Bit Device, Even (A0 = 0)

3.8.3.4 Byte Operand, 16-Bit Device, Odd (A0 = 1)

For a byte transfer between an odd memory location and a 16-bit device, the EBI drives the address bus with the desired address and drives the SIZE pin high to indicate a byte operand transfer. In the case of a read the slave device places the data on bits [7:0] of the data bus. The EBI ignores bits [15:8]. In the case of a write, the EBI drives the single byte operand on bits [7:0] of the data bus. Bits [15:8] are not driven.

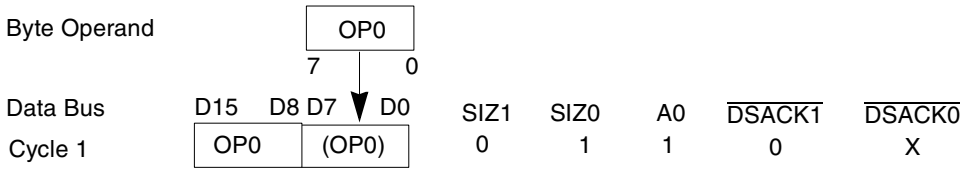


Figure 3-42 Byte Operand to 16-Bit Device, Odd (A0 = 1)

3.8.3.5 Word Operand, 8-Bit Device, Even (A0 = 0)

For a word transfer between an even memory location and an 8-bit device, two external bus cycles are performed. First, the EBI drives the address bus with the even address and drives the SIZE pin low to indicate a word operand transfer. In the case of a read, the slave device places the data on bits [15:8] of the data bus. In the case of a write, the EBI drives the most significant byte of the operand on bits [15:8] of the data bus. In the second bus cycle the EBI drives the address bus with the odd address

and the SIZE pin high to indicate a byte operand transfer. In the case of a read, the slave device places the data on bits [15:8] of the data bus. In the case of a write, the EBI drives the least significant byte of the operand on bits [15:8] of the data bus.

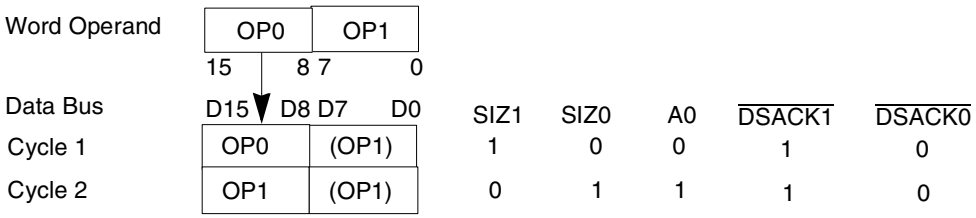


Figure 3-43 Word Operand to 8-Bit Device, Aligned

3.8.3.6 Word Operand, 16-Bit Device, Even (A0 = 0)

For a word transfer between an even memory location and a 16-bit device, the EBI drives the address bus with the desired address, and drives the SIZE pin low to indicate a word operand transfer. In the case of a read, the slave device responds by placing the data on bits [15:0] of the data bus. In the case of a write, the EBI drives the word operand on bits [15:0] of the data bus.

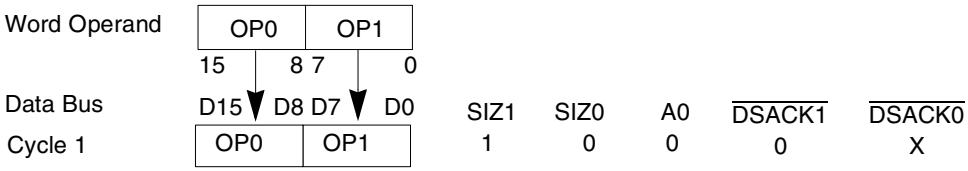


Figure 3-44 Word Operand to 16-Bit Device, Aligned

3.8.3.7 Long-Word Operand to 16-Bit Device, Aligned

Figure 3-45 shows both a long-word and word read and write timing to a 16-bit device. The MCU drives the address bus with the desired address and drives the size pins to indicate a long-word operand.

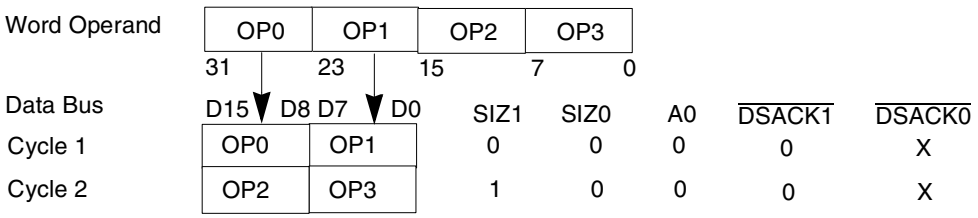


Figure 3-45 Long-Word Operand to 16-Bit Device, Aligned

For a read operation, the slave responds by placing the two most significant bytes of the operand on bits [15:0] of the data bus, and asserting $\overline{\text{DSACK1}}$ to indicate a 16-bit device. The MCU reads the two most significant bytes of the size counter, increments the address, initiates a new cycle, and reads bytes 2 and 3 of the operand from bits [15:0] of the data bus.

For a write operation, the MCU drives the two most significant bytes of the operand on bits [15:0] of the data bus. The slave device then reads the two most significant bytes of the operand (bytes 0 and 1) from bits [15:0] of the data bus and asserts $\overline{\text{DSACK1}}$ to indicate that it received the data and is a 16-bit device. The MCU then decrements the transfer size counter by two, increments the address by two, and writes bytes 2 and 3 of the operand to bits [15:0] of the data bus.

3.8.3.8 Long-Word Operand to 8-Bit Device, Aligned

The MCU drives the address bus with the desired address and the size pins to indicate a long-word operand, see **Figure 3-46**.

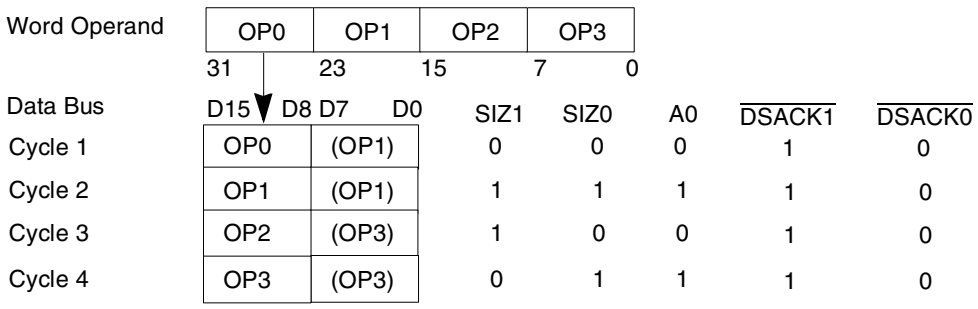


Figure 3-46 Long-Word Operand to 8-Bit Device, Aligned

For a read operation, the slave places the most significant byte of the operand on bits [15:8] of the data bus, and asserting $\overline{\text{DSACK0}}$ to indicate an 8-bit device. The MCU

reads the most significant byte of the operand (byte 0) from bits [15:8] and ignores bits [7:0]. The MCU then decrements the transfer size counter, increments the address, initiates a new cycle, and reads byte 1 of the operand from bits [15:8] of the data bus. The MCU repeats the process of decrementing the transfer size counter, incrementing the address, initiating a new cycle and reading a byte to trnasfer the remaining two bytes.

For a write operation, the MCU drives the two most significant bytes of the operand on bits [15:0] of the data bus. The slave divice then reads only the most significant byte of the operand (byte 0) from bits [15:8] of the data bus and asserts $\overline{\text{DSACK0}}$ (but not $\overline{\text{DSACK1}}$) to indicate that it received the data and it is an 8-bit device. The MCU then decrements the transfer size counter, increments the address, and writes byte 1 of the operand to bits [15:8] of the data bus. The MCU continues to decrement the transfer size counter, increment the address, and write a byte to transfer the remaining two bytes to the slave device.

3.8.4 Bus Master Cycles

In this section, each EBI bus cycle type is defined in terms of actions associated with a succession of internal states. These internal states are only for reference.

Read or write operations may require multiple bus cycles to complete based on the operand size, operand alignment, and target device size. Refer to **3.8.3 Operand Transfer** for more information. In the discussion that follows, it is assumed that only a single bus cycle is required to transfer the data.

In the timing diagrams, data transfers are related to clock cycles, independent of the clock frequency. The external bus states are also referenced.

Note that WAIT states on the BIM EBI are counted differently than on previous Integration Modules. A BIM 0 wait state cycle is a 2-clock bus cycle.

3.8.4.1 Read Cycles

During a read cycle, the EBI receives data from a memory or peripheral device. A flowchart of a typical read cycle operation is shown in **Figure 3-47**. On **Figure 3-49** and **Figure 3-50** are timing diagrams of external bus master cycles with and without wait states. The timing diagrams also show IMB bus activity to assist in understanding. The EBI does support delayed ds cycles, although this case is not shown.

- State 0 — The EBI drives the address bus and function codes. R/$\overline{\text{W}}$ is driven high, indicating a read cycle, and the SIZE pin is driven to indicate the number of bytes in the transfer.
- State 1 — One-half clock later in S1, the EBI asserts $\overline{\text{AS}}$, indicating that the address on the address bus is valid. $\overline{\text{DS}}$ is asserted indicating that the EBI is ready for data. $\overline{\text{CS}}$[1] may be asserted with either $\overline{\text{AS}}$ or $\overline{\text{DS}}$.
  The selected device uses R/$\overline{\text{W}}$, SIZE, A[0], and $\overline{\text{DS}}$ to place its information on the data bus (D[15:8] and/or D[7:0]). One or both bytes of the data bus are selected by SIZE and A[0].

1. A chip select pin may not be available in all BIM configurations.

- State 2 — No new control signals are issued during S2.
  If $\overline{\text{DTACK}}$[2] was asserted before the beginning of S2, the EBI proceeds to State 5.
- Optional Wait States — Wait states[3] are inserted until the slave asserts $\overline{\text{DTACK}}$.
- State 5 — During S5, the incoming data is latched into the internal data bus holding register. $\overline{\text{CS}}$, $\overline{\text{AS}}$ and $\overline{\text{DS}}$ are negated. The address bus, function codes, R/$\overline{\text{W}}$, and SIZE remain valid through S5 to allow for static memory operation and signal skew.
  The slave device asserts $\overline{\text{DTACK}}$ and data until it detects the negation of $\overline{\text{AS}}$ or $\overline{\text{DS}}$ (whichever is detected first). The slave must remove its data and negate $\overline{\text{DTACK}}$ within approximately one-half of a clock period after recognizing the negation of $\overline{\text{AS}}$ or $\overline{\text{DS}}$. Note that the data bus is not free until S1.

2. $\overline{\text{DTACK}}$ can be asserted internally by a chip select channel or the $\overline{\text{DTACK}}$ pin can be asserted.
3. Wait states are counted in full clocks, two half-clock tics each.

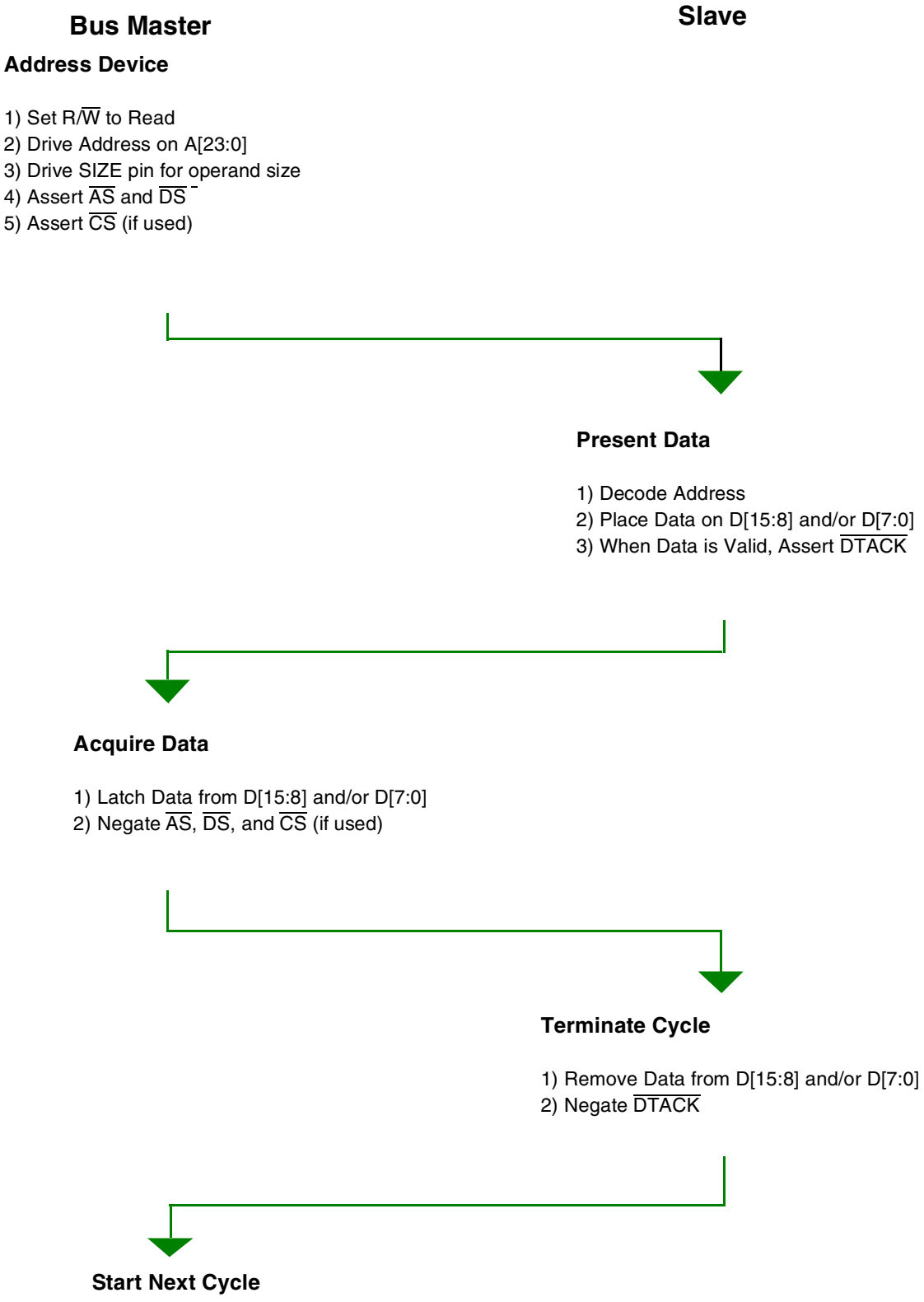


Figure 3-47 Read Cycle Flowchart

3.8.4.2 Write Cycles

On a write cycle, the EBI transfers data to a memory or peripheral device. **Figure 3-48** is a flowchart of the write cycle operation. The timing diagrams also show IMB bus activity to assist in understanding. The EBI does support delayed ds cycles, although this case is not shown.

- State 0 — The EBI drives the address bus and function codes. R/$\overline{W}$ is driven low, indicating a write cycle, and the SIZE pin is driven to indicate the number of bytes in the transfer.
- State 1 — One-half clock later in S1, the EBI asserts $\overline{AS}$, indicating that the address on the address bus is valid. $\overline{CS}$ may be asserted with $\overline{AS}$.
- State 2 — The EBI drives its data onto the data bus (D[15:0] and/or D[7:0]). The selected device uses R/$\overline{W}$, SIZE, and A[0] to take its information from the data bus. One or both bytes of the data bus are selected by SIZE and A[0].

  If $\overline{DTACK}$[4] is asserted by the slave device before the beginning of S2, the EBI proceeds to State 5.
- Optional Wait States — Wait states[5] are inserted until the slave asserts $\overline{DTACK}$. When data is stable on the data bus, $\overline{DS}$ is asserted. $\overline{CS}$ may be asserted with $\overline{DS}$.
- State 5 — During S5, $\overline{CS}$, $\overline{AS}$ and $\overline{DS}$ are negated. The address, function codes, R/$\overline{W}$, and SIZE remain valid through S5 to allow for static memory operation and signal skew.
  The EBI asserts data through S5.

  The slave must negate $\overline{DTACK}$ within approximately one-half of a clock period after recognizing the negation of $\overline{AS}$ or $\overline{DS}$.

4. $\overline{DTACK}$ can be asserted internally by a chip select channel or the $\overline{DTACK}$ pin can be asserted.
5. Wait states are counted in full clocks, two half-clock tics each.

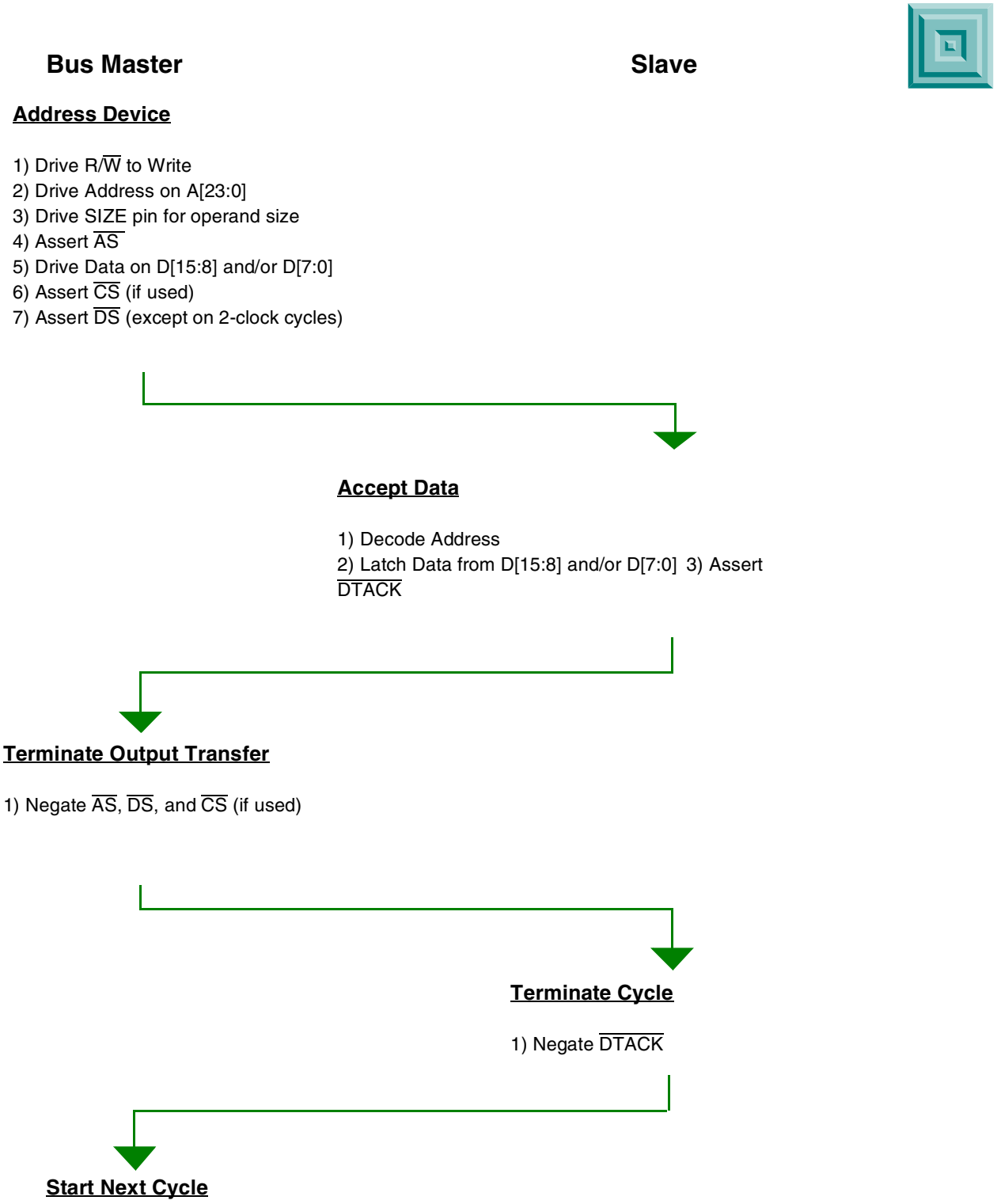


Figure 3-48 Write Cycle Flowchart

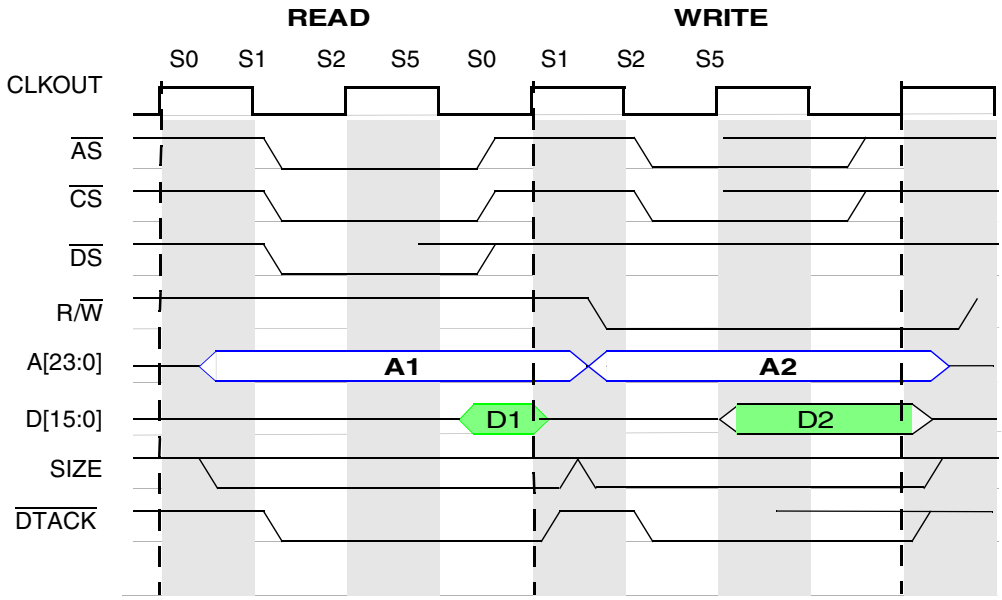


Figure 3-49 Master Mode — 2-Clock Bus Cycle

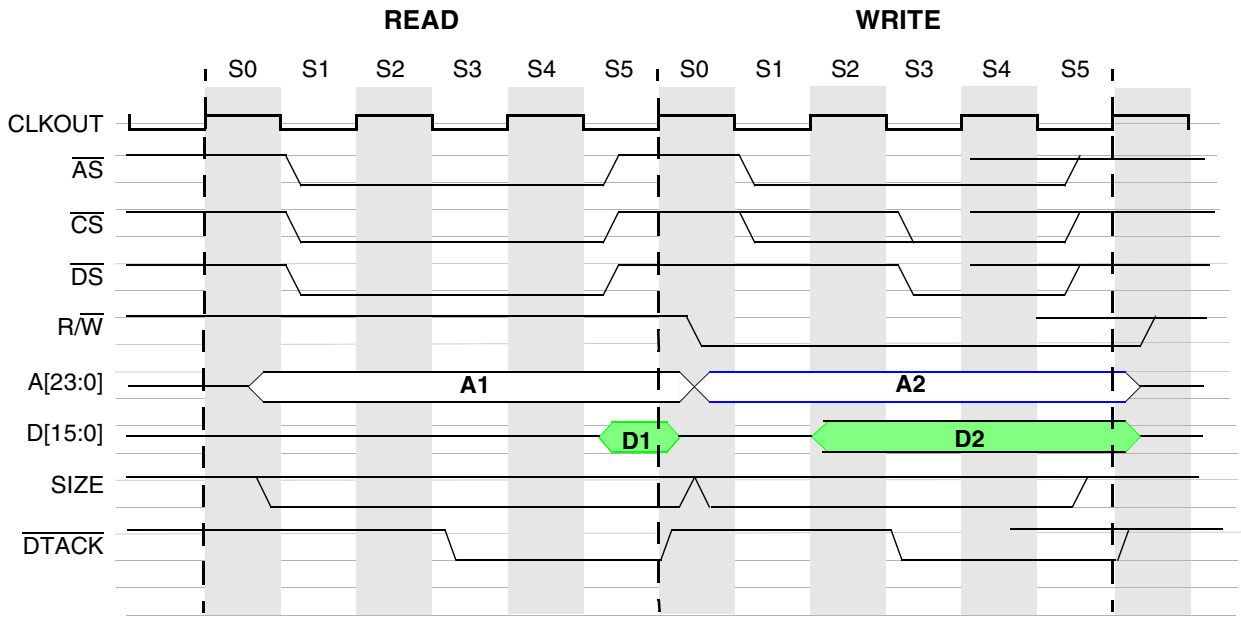


Figure 3-50 Master Mode — 3 Clock Bus Cycle

3.8.5 Burst Read and Write Cycles

The BIM is able to run synchronous burst cycles when a burst request is issued by a burst-capable CPU. Burst protocol, a superset of the basic cycle, allows one or more data transfers to be completed in a single bus cycle by stretching the DATA portion of the basic cycle. In a burst cycle the EBI issues the address of the first data item to be

transferred. The slave acknowledges data transfers by asserting $\overline{\text{BTACK}}$. Data transfers continue until either the master or the slave terminate the cycle by asserting $\overline{\text{DTACK}}$. The final transfer of a burst cycle occurs in state S5.

$\overline{\text{BTACK}}$ is both an indication that a device supports burst protocol and that the device is ready to transfer burst data. In most systems, $\overline{\text{BTACK}}$ cannot be tied to VSS, rather it must be asserted only for the burst address range decoded by the burst device. If $\overline{\text{BTACK}}$ is tied to V_{SS}, every external burst request must have a corresponding burst device capable of sustaining zero wait state burst access through it's entire address range. Finally, if $\overline{\text{BTACK}}$ is tied to VSS, an access to an unimplemented address will not cause the bus time-out monitor to assert internal $\overline{\text{BERR}}$.

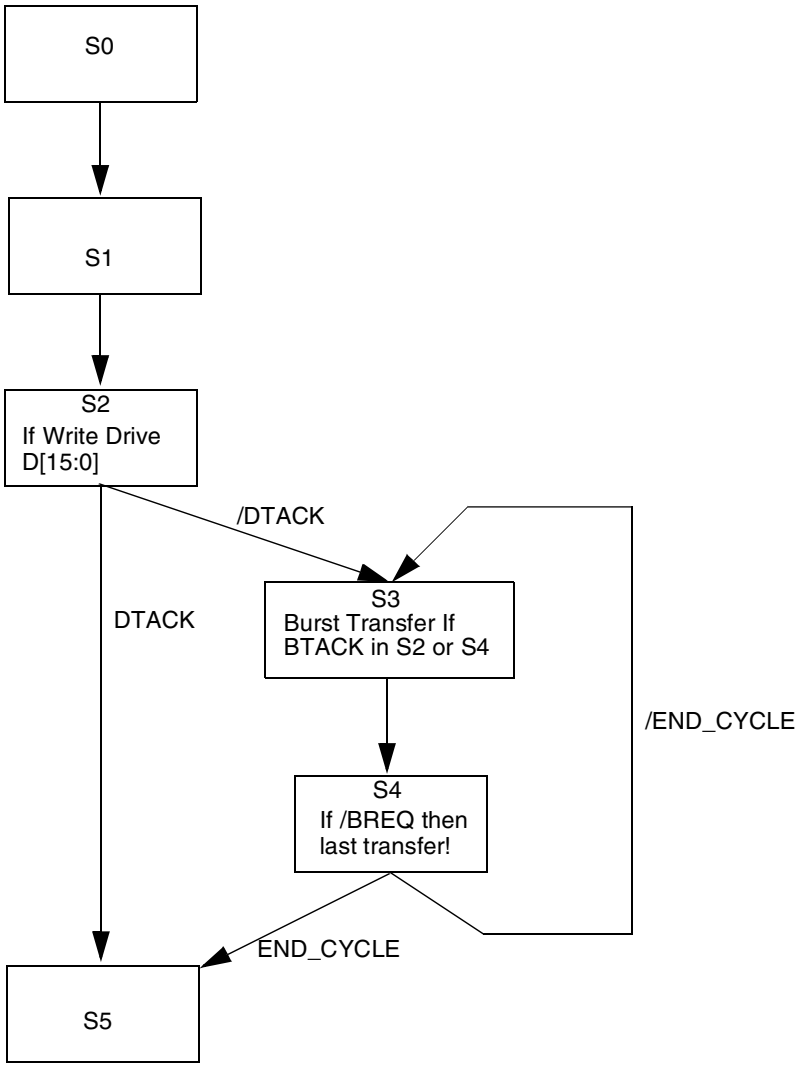
Both the master and the slave maintain the address of the data being transferred and increment the address each time data is successfully transferred. Every burst device is required to implement the same address modification algorithm. Burst cycles always transfer bus width aligned data; no partial bus width transfers, (i.e., 8-bit devices, are supported).

To support low latency bus operation required in microcontroller systems, the bus master may terminate a burst cycle by negating $\overline{\text{BREQ}}$, or a slave may terminate the burst cycle by asserting $\overline{\text{DTACK}}$ or $\overline{\text{BERR}}$. Due to burst timing handshake constraints, if the bus master terminates the burst cycle by negating $\overline{\text{BREQ}}$, the EBI will terminate the bus cycle for the burst device by internally asserting $\overline{\text{DTACK}}$ when the burst device indicates that it is ready to transfer data ($\overline{\text{BTACK}}$ asserts). If a reset is pending, the EBI also terminates burst read cycles by asserting $\overline{\text{DTACK}}$. Burst write cycles are allowed to complete when a reset is pending without EBI interference.

For increased cycle issue flexibility the bus master may convert a burst cycle to a basic cycle without a performance penalty by negating $\overline{\text{BREQ}}$ before the slave transfers the first data item (i.e. a burst length of 1 gracefully degrades to a two clock cycle). Additional waveform specifications for burst transfers can be found later in this section.

NOTE

At this time, the only burst-capable CPU is the CPU32X. In addition, the CPU32X only supports burst reads.



END_CYCLE = S3 * DTACK - slave terminates burst on current data transfer
+ S3 * /BREQ * BTACK - master terminates burst at next data transfer

Figure 3-51 Burst Cycle State Machine

3.8.5.1 Burst Read Cycles

During a burst read cycle, the EBI receives data from a burst memory device.

- State 0 —The EBI drives the address bus and function codes. $\overline{\text{BREQ}}$ is asserted, indicating the bus master is requesting a burst transfer. R/$\overline{\text{W}}$ is driven high, indi-

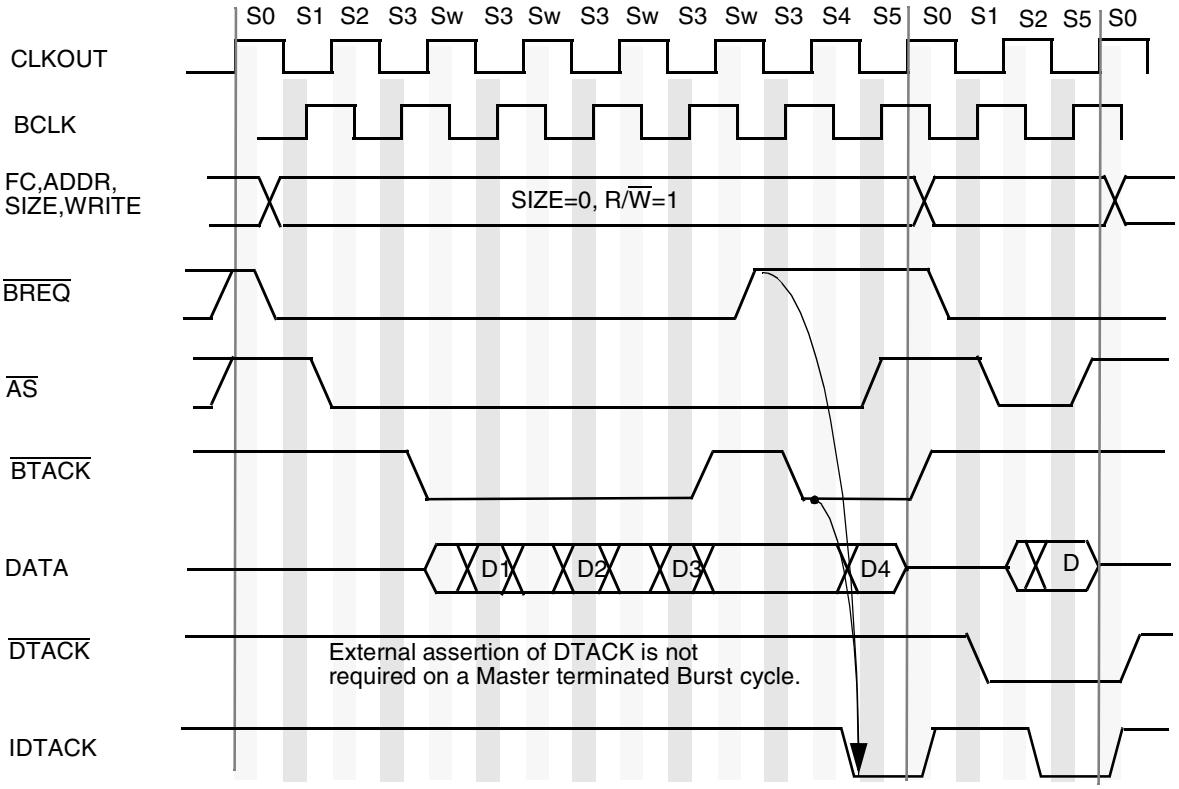
cating a read cycle, and the SIZE pin is driven low to indicate a word-size transfer.

- State 1 — The EBI asserts AS, DS, (LBA and BOE) indicating that the address on the address bus is valid and that the slave device may drive data on D[15:0]. The slave device may assert DTACK[6] by the end of S1 to convert the burst cycle to a 2 clock non-burst cycle (State sequence S0, S1, S2, S5).
- State 2 — If BTACK, burst data is to be transferred in S3. The EBI may negate BREQ in S2 to convert the burst cycle to a standard read cycle. LBA is negated. If DTACK then S5 else S3.
- State 3 — If BTACK was asserted previously, then at the beginning of S3 the burst device places data on D[15:0]. Midway through S3, the EBI latches data into the internal data bus holding register.
  If DTACK and BTACK are negated then a wait state is inserted by proceeding to SW.

  If DTACK is asserted the cycle will terminate, proceed to State 4.
- State SW — State SW is a wait state. EBI may negate BREQ in SW to terminate the burst cycle after the current data transfer has completed Proceed to State S3.
- State 4 — No signal changes.
- State 5 — At the beginning of S5 the burst device places the final data on D[15:0]. Midway through S5, the EBI latches data into the internal data bus holding register. AS, DS, and BOE are negated. The address bus, function codes, R/W, BREQ, and SIZE remain valid through S5 to allow for static memory operation and signal skew.
  The slave device negates DTACK, BTACK and its data before the end of S0.

  The following burst read waveforms illustrate various burst cycle scenarios.

6. DTACK can be asserted internally by the Burst Chip Select or the DTACK pin can be asserted.



NOTE: BIM asserts IDTACK @ $\overline{\text{BTACK}}$ if Master terminates a burst cycle by negating $\overline{\text{BREQ}}$.

Cycle 1: Master terminates burst cycle by negating $\overline{\text{BREQ}}$. Slave inserts 1 wait on last transfer.
Cycle 2: Burst cycle terminated by slave after 1 data transfer without wait states.

Figure 3-52 Burst Read — Master Termination

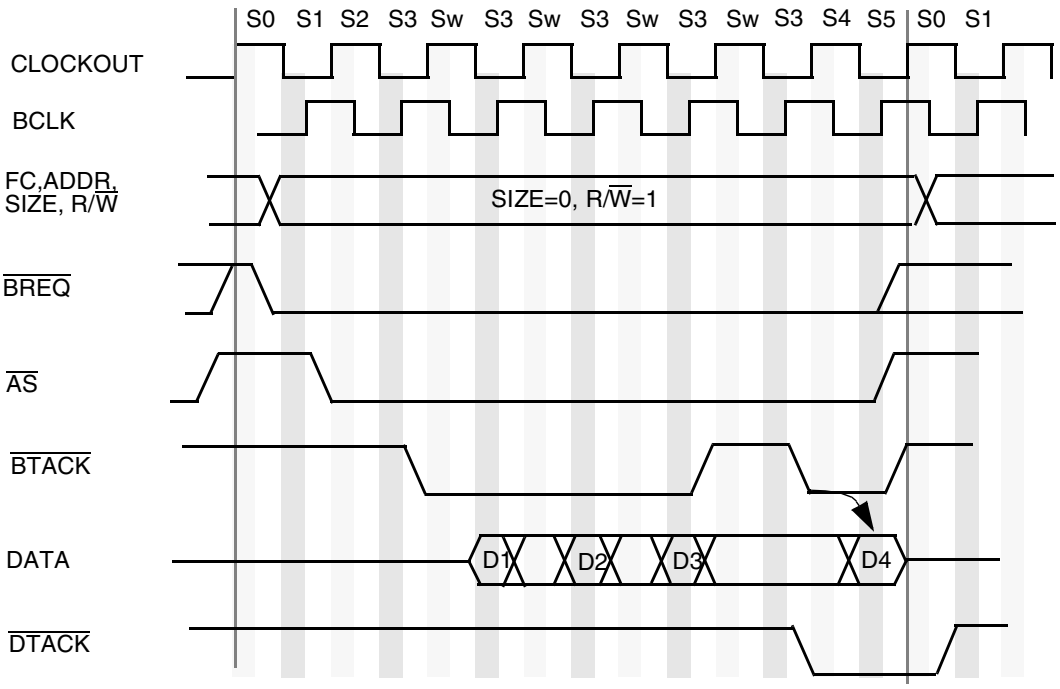


Figure 3-53 Burst Read — Slave Termination

3.8.5.2 Read-Modify-Write (RMC)

When the read-modify-write instruction is executed, an internal, indivisible read-modify-write cycle is performed. The CPU first performs a read, modifies the data, and then writes the data back to the same address. The BIM does NOT provide external indication that a read-modify-write cycle is in progress.

For both the read and write cycles, multiple bus cycles may be required to transfer the entire operand. The external bus may be arbitrated away from the BIM at any point during the read-modify-write operation.

3.8.6 CPU Space Cycles

CPU space cycles are used for LPSTOP broadcast cycles, interrupt acknowledge cycles, and breakpoint acknowledge cycles. The CPU space type is encoded on bits [19:16] of the address bus and the function codes are driven to 0x7 to distinguish these cycles. ***Refer to the User's Manual for the appropriate CPU for a complete description of CPU space cycle encoding.***

CPU space cycles are handled like other data transfers. The EBI drives the address, function codes, SIZE and R/W signals. $\overline{AS}$ remains negated for all CPU space cycles, internal or external. For applications that require an external indication of CPU space cycles, a chip select may be programmed to assert. See **3.6.4 Asynchronous Chip Select Operations**.

3.8.6.1 LPSTOP Broadcast Cycles

The LPSTOP write cycle is generated by the CPU when executing the LPSTOP instruction. The BIM latches the interrupt mask value driven on the data bus by the CPU and terminates the cycle internally. The BIM does not drive the interrupt mask value on the external data bus, unless show cycles is enabled . A chip select may be programmed to assert on LPSTOP broadcast cycles. Refer to **3.1.8 LPSTOP Operation** for an LPSTOP overview.

While in LPSTOP mode, the $\overline{\text{EBR}}$ pin controls whether the external pins are driven. If $\overline{\text{EBR}}$ is asserted, the BIM does not drive the address, data, $\overline{\text{AS}}$, $\overline{\text{DS}}$, R/$\overline{\text{W}}$, $\overline{\text{SIZE}}$, and $\overline{\text{BERR}}$ pins. If $\overline{\text{EBR}}$ is negated, the BIM drives the pins to the state they were in when LPSTOP was entered. If the state of $\overline{\text{EBR}}$ is changed during LPSTOP, the EBI reacts accordingly.

3.8.6.2 Breakpoint Acknowledge Cycles

The CPU performs a breakpoint acknowledge cycle in response to a BKPT instruction or the assertion of the $\overline{\text{BKPT}}$ pin. When the breakpoint acknowledge cycle is terminated internally by an IMB module and show cycles is enabled, the EBI drives the external data bus to reflect the internal data activity. When the breakpoint acknowledge cycle is an external cycle, $\overline{\text{AS}}$ is not asserted. A chip select pin can be programmed to assert for external breakpoint cycles.

3.8.6.3 Interrupt Acknowledge Cycles

An interrupt acknowledge cycle is performed to service an interrupt request. When the interrupt acknowledge cycle is terminated internally by an IMB module the EBI drives the external data bus to reflect the internal data bus activity. When the interrupt acknowledge cycle is to service an interrupt request from an IRQ pin and an autovector is not selected, the EBI performs an external interrupt acknowledge cycle to acquire the interrupt vector.

An external interrupt acknowledge flow chart is shown in **Figure 3-54**, and a timing diagram for external IACK cycles is shown in **Figure 3-55**. Refer to **3.1.7 Interrupt Operation** for an interrupt overview.

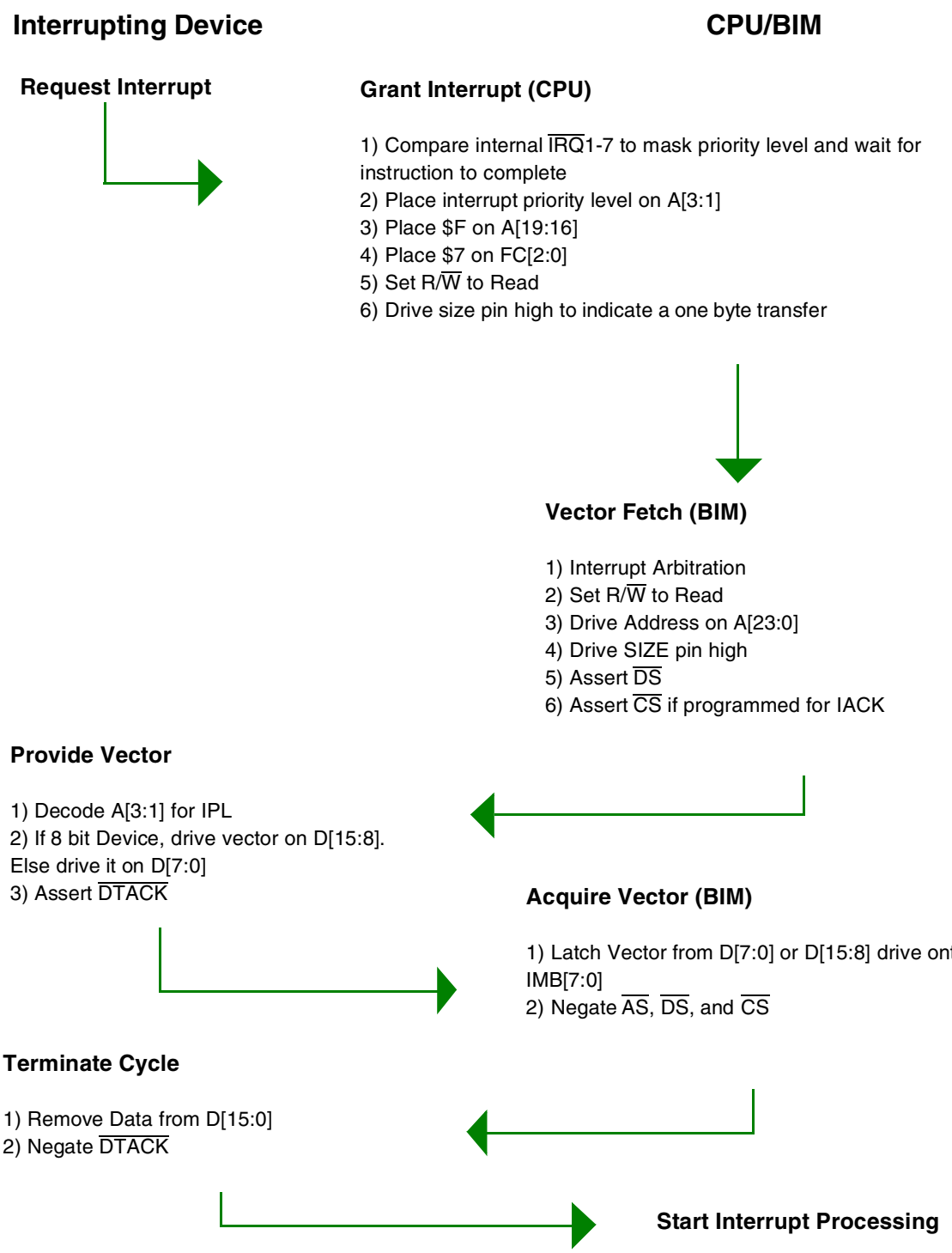


Figure 3-54 IACK Sequence Flow, External Vector

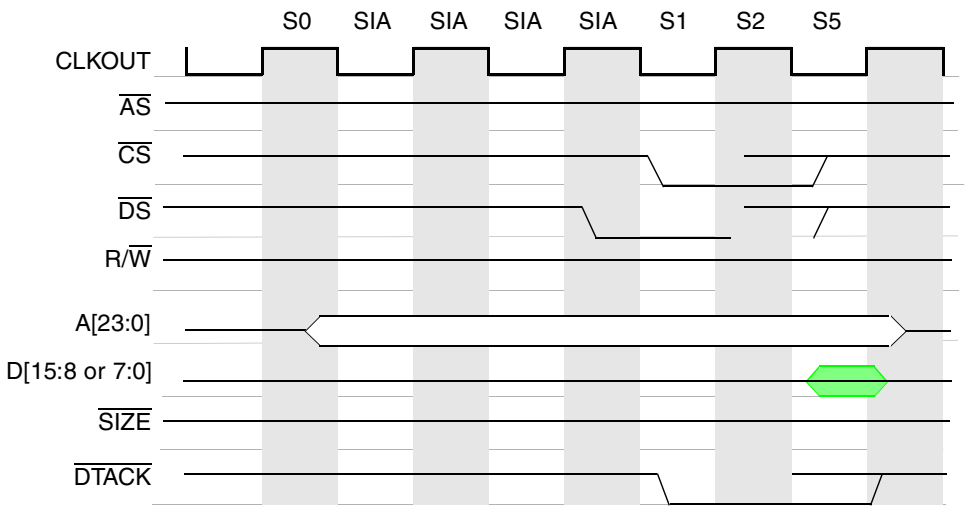


Figure 3-55 IACK Cycle — External Vector

3.8.7 Bus Exception Operation

Normal bus cycle termination requires the assertion of the DTACK pin or the internal IDTACK signal. Minimal bus exception support is provided by bus error cycle termination. For bus error cycle termination, the external BERR pin or the internal IBERR signal is asserted. Bus error cycle termination takes precedence over normal cycle termination, provided BERR assertion meets the timing constraints described in **E.2.2 BIM AC Timing**.

The BIM provides an internal bus monitor which optionally asserts the internal IBERR signal when DTACK or BTACK response time is too long. Also, a spurious interrupt monitor terminates IACK cycles asserting the internal IBERR signal, when no interrupt arbitration occurs. For more information on these internal IBERR functions, see **3.5 System Protection**.

Address error execptions are also supported by the BIM. Other **MC68000** family bus exceptions, (late bus error, retry, relinquish and retry, and halt), are **not** supported by the BIM.

3.8.7.1 External Bus Arbitration

The BIM external bus has limited support for an alternate bus master via the external bus request (EBR) signal. When EBR is asserted, the BIM completes its bus cycle, and relinquishes control of the external bus. After AS and DS have been negated for three clock cycles, the alternate master may take control of the external busThe alternate master cannot access any MCU internal resources.

If the external bus is granted to an alternate bus master in master, MFTM, emulation, or background debug operation mode, the pins that tri-state depend on the pin function specified in the pin assignment register. Pins programmed to function as A[23:0],

FC[2:0], D[15:0], $\overline{\text{BREQ}}$, $\overline{\text{AS}}$, $\overline{\text{DS}}$, R/$\overline{\text{W}}$, SIZE, $\overline{\text{CS1}}$ – $\overline{\text{CS7}}$, $\overline{\text{LBA}}$, $\overline{\text{BAA}}$, $\overline{\text{BWE}}$, and $\overline{\text{BOE}}$ tristate.

$\overline{\text{EBR}}$ operation is not defined in SLAM or single chip mode. Asserting $\overline{\text{EBR}}$ causes undefined pin operation.

While $\overline{\text{EBR}}$ is asserted, the CPU will continue to execute internal bus cycles until an external bus cycle is needed. At this point, the BIM will disable its bus time-out logic and insert wait states on the IMB until the external bus master has relinquished the bus mastership. Pins programmed to function as DTACK, BTACK, BERR, and BKPT have no affect on the internal bus cycle execution when the external bus is granted to an alternate master.

Because the external bus may be relinquished at bus cycle boundaries, data coherency for external long-word or misaligned accesses is not maintained. If show cycles are enabled, data coherency for read-modify-write cycles and internal longword or misaligned accesses is not maintained. Refer to **Figure 3-56** for a timing diagram.

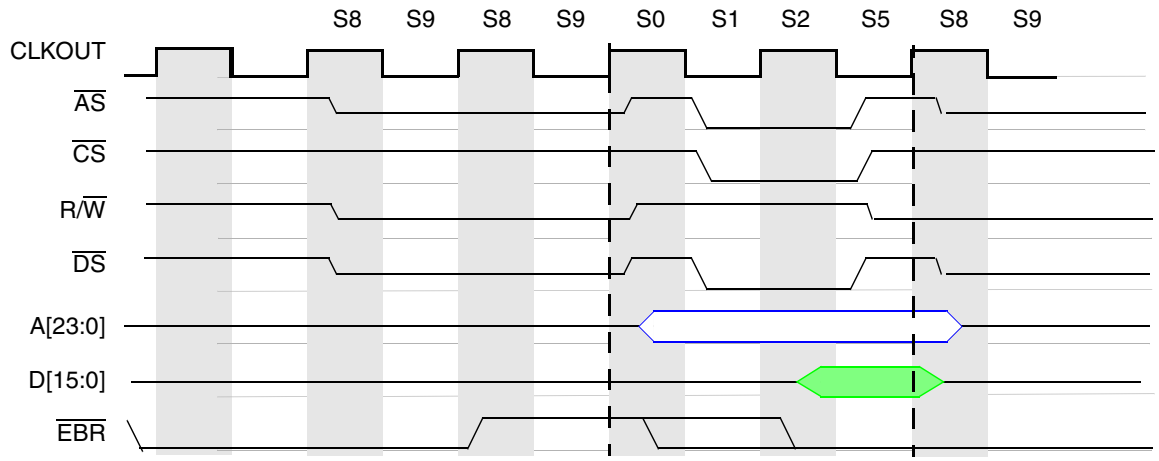


Figure 3-56 $\overline{\text{EBR}}$ Timing