



## SECTION 5

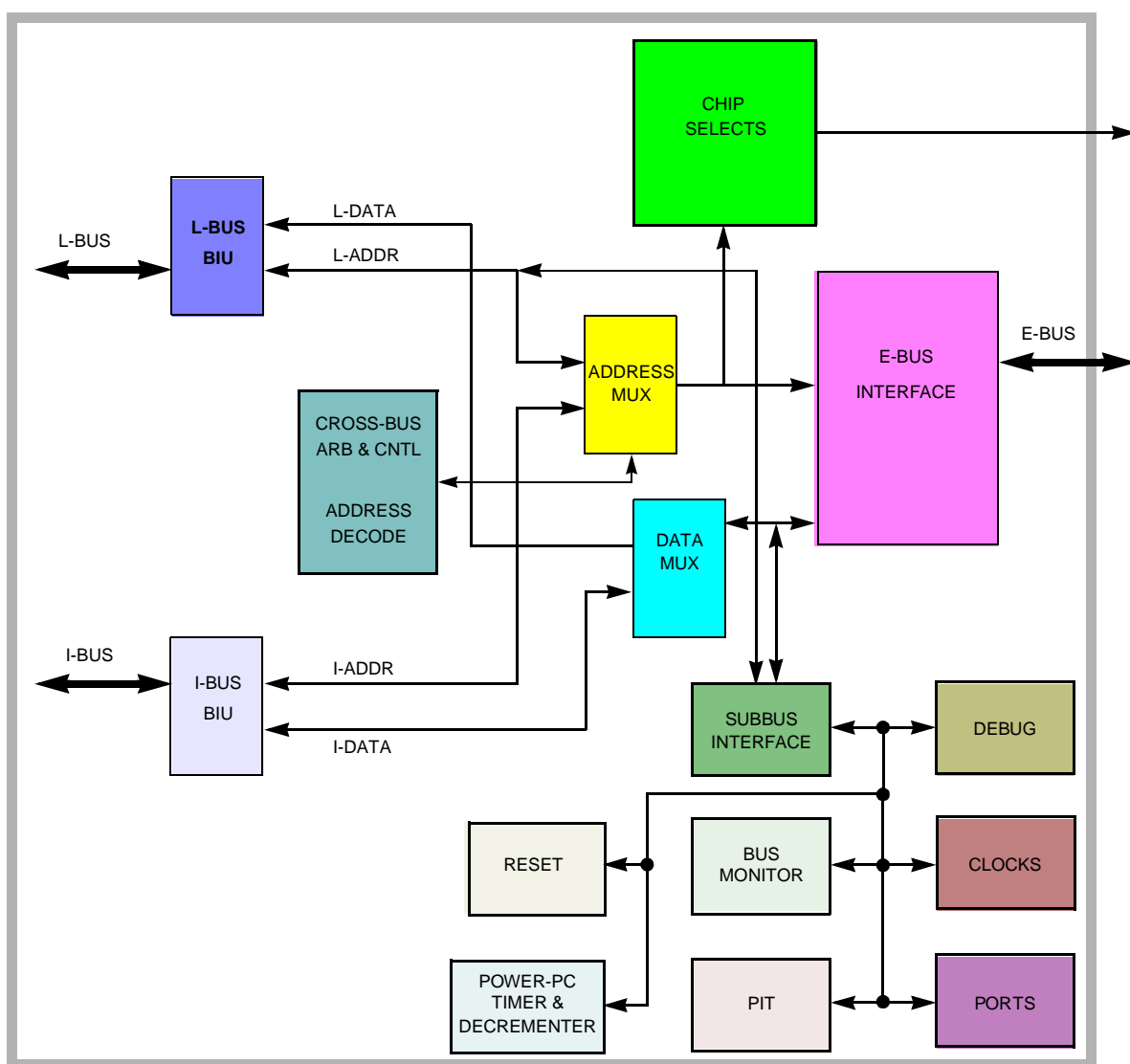
### SYSTEM INTERFACE UNIT

The system interface unit (SIU) consists of modules that control the buses of the chip, provide the clocks, and provide miscellaneous functions for the system, such as chip selects, test control, reset control, and I/O ports.

The MPC509 has an internal Harvard architecture and a single external bus. The internal buses are the instruction bus (I-bus) and the load/store bus (L-bus). The external bus interface (EBI) connects each of these internal buses with the external bus (E-bus). The chip select block provides user-programmable chip selects to select external memory or peripherals. The clock block controls the generation of the system clocks and such features as programmability of the clocks and low-power modes. The reset control function interfaces to the reset pins and provides a reset status register. The general-purpose I/O ports provide untimed I/O functions on pins that are not used for their primary function.

#### 5.1 SIU Block Diagram

A block diagram of the SIU is shown in [Figure 5-1](#).



**Figure 5-1 SIU Block Diagram**

## 5.2 SIU Address Map

**Table 5-1** is an address map of the SIU registers. An entry of “S” in the Access column indicates that the register is accessible in supervisor mode only. “S/U” indicates that the register can be programmed to the desired privilege level. “Test” indicates that the register is accessible in test mode only.



**Table 5-1 SIU Address Map**

Access	Address	Register
S	0x8007 FC00	SIU module configuration register (SIUMCR)
Test	0x8007 FC04	SIU test register 1 (SIUTEST1)
—	0x8007 FC08 – 0x8007 FC1C	Reserved
S	0x8007 FC20	Memory mapping (MEMMAP)
S	0x8007 FC24	Speculative address register (SPECADDR)
S	0x8007 FC28	Speculative mask register (SPECMASK)
Test	0x8007 FC2C	Termination status register (TERMSTAT)
—	0x8007 FC30 – 0x8007 FC3C	Reserved
S/U	0x8007 FC40	Periodic interrupt control and status register (PICSR)
S/U	0x8007 FC44	Periodic interrupt timer register (PIT)
S	0x8007 FC48	Bus monitor control register (BMCR)
S	0x8007 FC4C	Reset status register (RSR)
S	0x8007 FC50	System clock control register (SCCR)
S	0x8007 FC54	System clock lock and status register (SCLSR)
—	0x8007 FC58 – 0x8007 FC5C	Reserved
S	0x8007FC60	Port M data direction (DDRM)
S	0x8007FC64	Port M pin assignment (PMPAR)
S/U	0x8007FC68	Port M data (PORTM)
—	0x8007FC6C – 0x8007FC80	Reserved
S	0x8007FC84	Port A, B pin assignment (PAPAR, PBPAR)
S/U	0x8007FC88	Port A, B data (PORTA, PORTB)
—	0x8007FC8C – 0x8007FC94	Reserved
S	0x8007FC98	Port I, J, K, L data direction (DDRI, DDRJ, DDRK, DDRL)
S	0x8007FC9C	Port I, J, K, L pin assignment (PIPAR, PJPAR, PKPAR, PLPAR)
S/U	0x8007FCA0	Port I, J, K, L data (PORTI, PORTJ, PORTK, PORTL)
—	0x8007 FCA4 – 0x8007 FD94	Reserved
S	0x8007 FD94	$\overline{CS11}$ option register (CSOR11)
S	0x8007 FD98	Reserved

**Table 5-1 SIU Address Map (Continued)**

Access	Address	Register
S	0x8007 FD9C	$\overline{\text{CS10}}$ option register (CSOR10)
S	0x8007 FDA0	Reserved
S	0x8007 FDA4	$\overline{\text{CS9}}$ option register (CSOR9)
S	0x8007 FDA8	Reserved
S	0x8007 FDAC	$\overline{\text{CS8}}$ option register (CSOR8)
S	0x8007 FDB0	Reserved
S	0x8007 FDB4	$\overline{\text{CS7}}$ option register (CSOR7)
S	0x8007 FDB8	Reserved
S	0x8007 FDBC	$\overline{\text{CS6}}$ option register (CSOR6)
S	0x8007 FDC0	$\overline{\text{CS5}}$ base address register (CSBAR5)
S	0x8007 FDC4	$\overline{\text{CS5}}$ option register (CSOR5)
S	0x8007 FDC8	$\overline{\text{CS4}}$ base address register (CSBAR4)
S	0x8007 FDCC	$\overline{\text{CS4}}$ option register (CSOR4)
S	0x8007 FDD0	$\overline{\text{CS3}}$ base address register (CSBAR3)
S	0x8007 FDD4	$\overline{\text{CS3}}$ option register (CSOR3)
S	0x8007 FDD8	$\overline{\text{CS2}}$ base address register (CSBAR2)
S	0x8007 FDDC	$\overline{\text{CS2}}$ option register (CSOR2)
S	0x8007 FDE0	$\overline{\text{CS1}}$ base address register (CSBAR1)
S	0x8007 FDE4	$\overline{\text{CS1}}$ option register (CSOR1)
S	0x8007 FDE8	Reserved
S	0x8007 FDEC	$\overline{\text{CS0}}$ option register (CSOR0)
S	0x8007 FDF0	$\overline{\text{CSBOOT}}$ sub-block base address register (CSBTSBBAR)
S	0x8007 FDF4	$\overline{\text{CSBOOT}}$ sub-block option register (CSBTSBOR)
S	0x8007 FDF8	$\overline{\text{CSBOOT}}$ base address register (CSBTBAR)
S	0x8007 FDFC	$\overline{\text{CSBOOT}}$ option register (CSBTOR)

### 5.3 SIU Module Configuration

The SIU module configuration register (SIUMCR) configures various aspects of SIU operation. The internal memory mapping register (MEMMAP) enables and sets the base address of the L-bus and I-bus internal memory blocks. These registers are accessible in supervisor mode only.

#### 5.3.1 SIU Module Configuration Register

The SIU module configuration register (SIUMCR) configures various aspects of SIU operation. This register is accessible in supervisor mode only.

## SIUMCR — SIU Module Configuration Register

0x8007 FC00



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SIU-FRZ	RESERVED		CSR	LST	0	SUP		DLK	LOK	RESERVED				LSHOW	

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
PARTNUM								MASKNUM							

RESET:

Read-Only Fixed Value

Read-Only Fixed Value

**Table 5-2 SIUMCR Bit Settings**

Bit(s)	Name	Description
0	SIUFRZ	SIU freeze 0 = Decrementer and time base registers and the periodic interrupt timer continue to run while internal freeze signal is asserted (reset value). 1 = Decrementer and time base registers and the periodic interrupt timer stop while the internal freeze signal is asserted. Refer to <a href="#">5.3.3 Internal Module Select Logic</a> and <a href="#">SECTION 8 DEVELOPMENT SUPPORT</a> for information on the freeze signal.
1:2	—	Reserved
3	CSR	Checkstop reset enable 0 = No action taken when SIU receives the checkstop signal from the CPU and debug mode not enabled (reset value). 1 = SIU causes a reset upon receiving checkstop signal from CPU and debug mode not enabled. If debug mode is enabled, the MCU enters debug mode when the checkstop signal is received, regardless of CSR value. Refer to the <i>RCPU Reference Manual</i> (RCPURM/AD) for more information on checkstop resets.
4	LST	Burst style: $\overline{\text{BDIP}}$ or $\overline{\text{LAST}}$ 0 = $\overline{\text{BDIP}}$ pin uses $\overline{\text{BDIP}}$ timing (reset value): assert $\overline{\text{BDIP}}$ during burst, negate $\overline{\text{BDIP}}$ during last beat of burst 1 = $\overline{\text{BDIP}}$ pin uses $\overline{\text{LAST}}$ timing: assert $\overline{\text{LAST}}$ during last beat of burst Refer to <a href="#">5.5.16.6 Synchronous Burst Interface</a> for more information.
5	—	Reserved
6:7	SUP	Supervisor/unrestricted space. These bits control access to certain SIU registers. (Other registers are always supervisor access only.) The access restrictions for each register are shown in <a href="#">Table 5-1</a> . 00 = Unrestricted access (reset value) 01 = Supervisor mode access only 10 = Supervisor mode write access only, unrestricted read access 11 = Supervisor mode access only

**Table 5-2 SIUMCR Bit Settings (Continued)**



Bit(s)	Name	Description
8	DLK	Debug register lock. This bit can be written only when internal freeze signal is asserted. DLK allows development software to configure show cycles and prevent normal software from subsequently changing this configuration. This bit overrides the LOK in controlling the LSHOW field. 0 = LSHOW field in SIUMCR can be written to (reset value). 1 = Writes to LSHOW field are not allowed.
9	LOK	Register lock. Once this bit is set, writes to the SIUMCR and chip-select registers have no effect and cause a data error to be generated in the internal bus. In normal operation, this is a set-only bit; once set, it cannot be cleared by software. When the internal freeze signal is asserted, the bit can be set or cleared by software. 0 = Normal operation (reset value) 1 = All bits in the SIUMCR and all of the chip-select registers are locked
10:13	—	Reserved
14:15	LSHOW	L-bus show cycles 00 = Disable show cycles for all internal L-bus cycles (reset value) 01 = Show address and data of all internal L-bus write cycles 10 = Reserved 11 = Show address and data of all internal L-bus cycles Refer to <a href="#">5.4.13 Show Cycles</a> for more information.
16:23	PARTNUM	Part number. This read-only field is mask programmed with a code corresponding to the number of the MCU.
24:31	MASKNUM	Mask number. This read-only field is mask programmed with a code corresponding to the mask number of the MCU.

### 5.3.2 Memory Mapping Register

The internal memory mapping register (MEMMAP) enables and sets the base address of the L-bus (SRAM) internal memory block. This register is accessible in supervisor mode only.

#### MEMMAP — Memory Mapping Register

**0x8007 FC20**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LEN	RESERVED							LMEMBASE	RESERVED						

RESET:

\* 0 0 0 0 0 0 0 0 \* \* 0 0 0 0 0 0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
IEN	LIX	RESERVED							IMEMBASE	RESERVED					

RESET:

\* 1 0 0 0 0 0 0 0 \* \* 0 0 0 0 0 0

\* Reset value depends on the value of the data bus configuration word during reset.



**Table 5-3 MEMMAP Bit Settings**

Bit(s)	Name	Description
0	LEN	L-bus memory enable 0 = L-bus memory disabled 1 = L-bus memory enabled Reset state depends on the value of the data bus configuration word.
1:7	—	Reserved
8:9	LMEMBASE	Base address of the L-bus memory block 00 = Starting address is 0x0000 0000 01 = Ending address is 0x000F EFFF 10 = Starting address is 0xFFFF 0000 11 = Ending address is 0xFFFF EFFF Reset value depends on the data bus configuration word.
10:15	—	Reserved
16	IEN	I-bus memory enable. This bit has no effect on the MPC509, which has no I-bus memory. 0 = I-bus memory disabled 1 = I-bus memory enabled
17	LIX	L-bus to I-bus cross-bus access enable 0 = Disable data accesses to I-bus memory 1 = Enable data accesses to I-bus memory (reset value)
18:23	—	Reserved
24:25	IMEMBASE	Base address of the I-bus memory block 00 = Starting address is 0x0000 0000 01 = Ending address is 0x000F FFFC 10 = Starting address is 0xFFFF 0000 11 = Ending address is 0xFFFF FFFC Reset state depends on the data bus configuration word.
26:31	—	Reserved

### 5.3.3 Internal Module Select Logic

The SIU has a unified memory map for the L-bus and the I-bus. The I-bus has two masters, the RCPU and the SIU. The SIU is designed so that one or more memory modules, such as flash EEPROM or instruction RAM, may be located on the I-bus. On the MPC509, however, no memory modules are located on the I-bus.

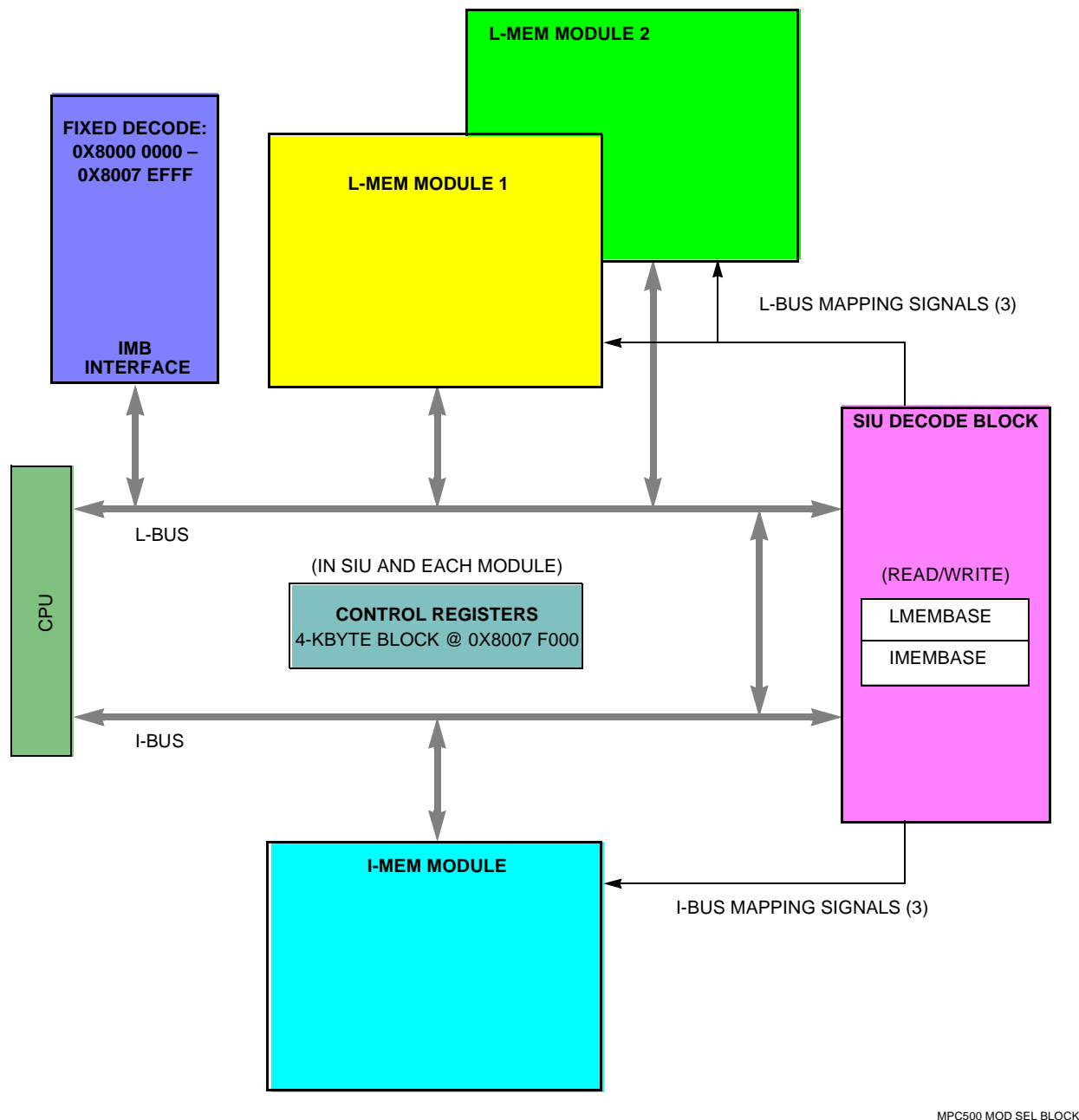
The L-bus has at least two masters, the RCPU and the SIU. One or more slave modules may reside on the L-bus. These may include memory modules (RAM, flash EEPROM) and on-chip (IMB2) peripherals, which are connected via the L-bus IMB2 interface (LIMB). On the MPC509, the SRAM module is located on the L-bus.

Each module on either bus has one or more internal control registers which control the configuration and operation of the module. On a memory module (i.e., the SRAM on the MPC509), these registers are not mapped with the memory array, but stay at a fixed address in the memory map.

Capability is provided to allow masters on one bus to access slaves on the opposite bus. L-bus masters must be able to access peripherals on the I-bus to program their control registers or to program flash memory arrays. This is because the CPU instruction fetch unit can only run read cycles. Similarly, I-bus masters are able to execute diagnostic programs out of the RAM on the L-bus. These capabilities require that the addresses of the memory modules on the I-bus be known to the L-bus address decode logic and vice versa. Refer to [5.3.4 Internal Cross-Bus Accesses](#) for details.



A block diagram of the internal module select scheme is shown in [Figure 5-2](#).



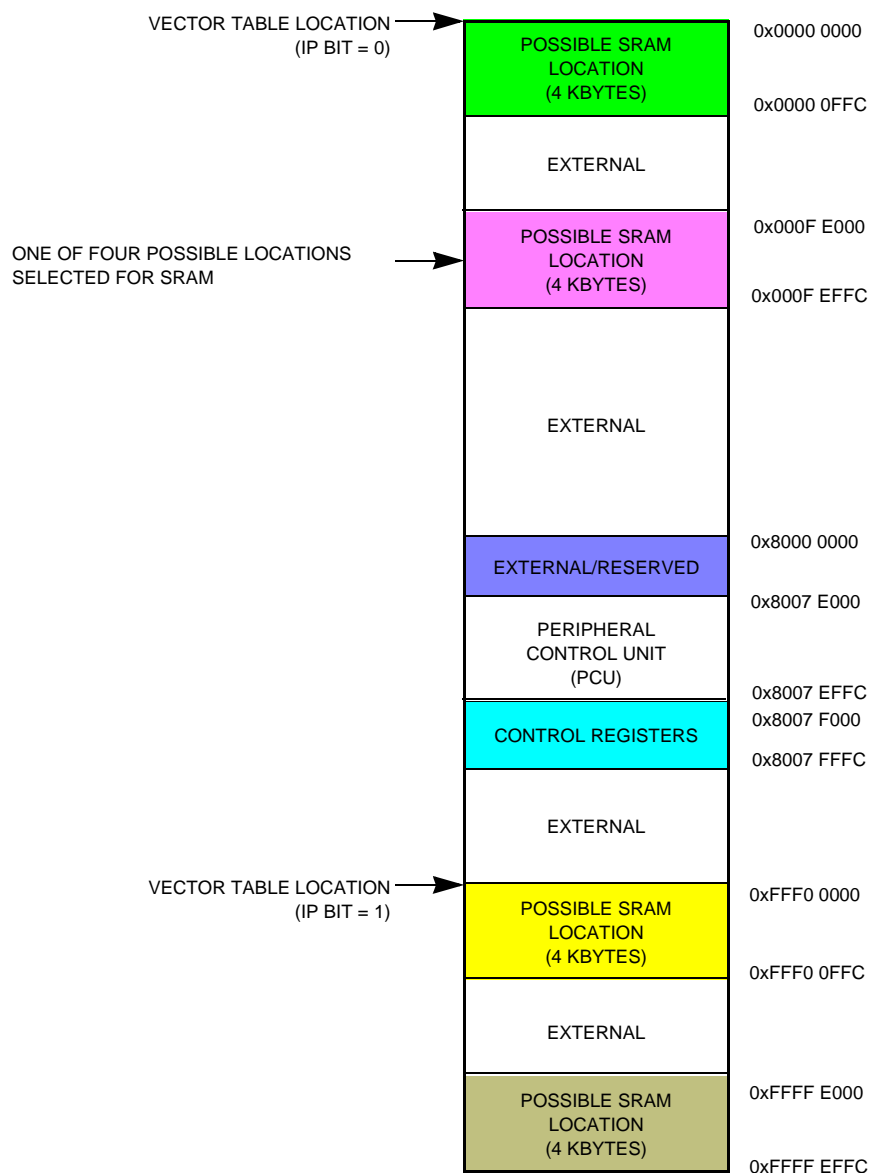
**Figure 5-2 Internal Module Select Scheme**



### 5.3.3.1 Memory Block Mapping

The SRAM array can be mapped to one of four locations. These locations are at the top and bottom of the 4-Gbyte address range. They include the two alternatives for the PowerPC vector map (0x0000 0100 and 0xFFFF 0100). The LMEMBASE field in the memory mapping register (MEMMAP) determine the locations of the SRAM array.

**Figure 5-3** shows the mapping of the memory blocks within the memory map.



**Figure 5-3 Placement of Internal Memory in Memory Map**

### 5.3.3.2 Accesses to Unimplemented Internal Memory Locations

If an access is made to a location within the  $2^n$ -sized memory block that is not implemented in any memory module on the chip, then the CPU takes a machine check exception.



### 5.3.3.3 Control Register Block

The internal control registers include all of the SIU registers and all of the configuration, control, and status registers of each module on the I-bus or L-bus. The internal control registers and the IMB2 are allocated a 512-Kbyte block from 0x8000 0000 to 0x8007 FFFF. The internal control registers always occupy the highest-numbered four Kbytes of this address range (0x8007 F000 to 0x8007 FFFF). The IMB2 modules (on the MPC509, this includes only the PCU) are allocated the remainder of the 512-Kbyte block.

Unlike the memory arrays, the internal control registers and IMB2 modules cannot be disabled for development purposes. In addition, the IMB2 and control register block are only available in data space, not in instruction space. An instruction access to the address of a control register results in a data error on the I-bus, causing the internal  $\overline{\text{TEA}}$  signal to be asserted.

### 5.3.3.4 Internal Memory Mapping Field (LMEMBASE)

The LMEMBASE field in the MEMMAP register maps the L-bus memory (i.e., the SRAM) to one of four possible locations in the memory map. The following table shows the meaning of the field. Note that these locations include the two possible locations for the CPU vector table. The address not given (start or end, depending upon the block) depends on the block size.

**Table 5-4 Internal Memory Array Block Mapping**

LMEMBASE	Block Placement
00	Starting address: 0x0000 0000
01	Ending address: 0x000F EFFC
10	Starting address: 0xFFFF 0000
11	Ending address: 0xFFFF EFFC

### 5.3.3.5 Memory Mapping Conflicts

Any access to a memory that does not exist causes the cycle to appear on the external bus. Because the MPC509 does not have an I-bus memory module, an access to I-bus memory is sent to the external bus, even if the I-bus memory enable (IEN) bit in the MEMBASE register is set.

### 5.3.4 Internal Cross-Bus Accesses



Each internal bus (I-bus and L-bus) has a master/slave interface in the SIU. The slave interface is used for accesses by the internal master (RCPU) to the external bus, to memory on the opposite bus (e.g., L-bus to I-bus access), or to SIU registers.

The SIU allows masters on either internal bus (I-bus or L-bus) to access slaves on the other internal bus. Accesses from one internal bus to resources on the other bus take at least three clocks, because of arbitration and cycle termination delays.

Instruction fetching from L-bus memory is intended primarily as a mechanism to allow a customer test program to be downloaded to on-chip RAM and executed. This is not a high-performance instruction fetching mechanism. Accesses from the I-bus to the L-bus are at least three clocks and not burstable.

Cross-bus accesses occur inside the SIU, consuming SIU resources during the access. Internal SIU registers are not available during cross-bus accesses. Internal-to-external cycles are not pipelined with cross-bus accesses, nor are two consecutive cross-bus accesses pipelined.

Clearing the L-bus to I-bus cross-bus access (LIX) bit in the MEMMAP register disables data accesses to I-bus memory. This allows load/store data stored in a flash memory on the I-bus to be moved off-chip for development purposes. When this bit is cleared, L-bus to I-bus transactions are run externally.

### 5.3.5 Response to Freeze Assertion

The RCPU asserts the freeze signal to the rest of the MCU when one of the following conditions occurs:

- Debug mode is entered; or
- A software debug monitor program is entered as the result of an exception when the associated bit in the debug enable register (DER, SPR149) is set.

The following paragraphs explain how the assertion of the freeze signal affects the SIU. See [SECTION 8 DEVELOPMENT SUPPORT](#) for additional details on this signal.

#### 5.3.5.1 Effects of Freeze and Debug Mode on the Bus Monitor

When the freeze signal is asserted and debug mode is disabled, the bus monitor is unaffected. This means that a software monitor must configure the bus monitor to provide protection from unterminated bus cycles that occur during debugging.

When the processor is in debug mode (debug mode is enabled and the freeze signal is asserted), the bus monitor is enabled. The bus monitor is also enabled when debug mode is enabled and a non-maskable breakpoint is asserted by the development port. These enables override the bus monitor enable bit (BME) in the bus monitor control register (BMCR) in the SIU. In both cases the bus monitor time-out period is whatever was programmed in the BMCR. This override allows an external development tool to retain control over the CPU in debug mode by not allowing an external bus cycle to hang the processor in an endless wait for a transfer acknowledge.

In addition, if the processor is executing normally and runs a bus cycle that is not terminated, a non-maskable breakpoint always gains control of the processor by terminating the bus cycle with the bus monitor so the processor can enter debug mode. In this case, the non-maskable breakpoint is not restartable. The processor takes the breakpoint before completing the prologue of the exception handler called as a result of the bus monitor.



### **5.3.5.2 Effects of Freeze on the Programmable Interrupt Timer (PIT)**

When freeze is asserted and the SIU freeze bit (SIUFRZ) is set in the SIU module configuration register (SIUMCR), the PIT is disabled. This disable overrides the periodic interrupt enable bit (PIE) in the periodic interrupt control and select register (PICSR) in the SIU. This allows the count in the PIT to be preserved when execution stops.

### **5.3.5.3 Effects of Freeze on the Decrementer**

When freeze is asserted and the SIUFRZ is set in the SIUMCR, the decrementer is disabled. This allows the value in the decrementer to be preserved when execution stops.

### **5.3.5.4 Effects of Freeze on Register Lock Bits**

When freeze is asserted the lock bits in various registers can be set or cleared. This allows the protected configurations to be changed and then re-locked by a development support system.

## **5.4 External Bus Interface**

The external bus interface (EBI) interfaces the external bus (E-bus) with the two internal buses (I-bus and L-bus). The E-bus is synchronous and supports pipelined and burst transfers. Signals driven onto the E-bus are required to meet the set-up and hold times relative to the rising edge of the bus clock. The bus has the ability to support multiple masters, but its protocol is optimized for a single-processor environment.

### **5.4.1 Features**

- No external glue logic required for a simple system.
- Supports different memory (SRAM, EEPROM) types: asynchronous, synchronous, pipelineable, burstable.
- Fast (one-clock) arbitration possible.
- Bus is synchronous — all signals are referenced to the rising edge of the bus clock.
- 32-bit data bus, 32-bit address bus with byte enables.
- Compatible with PowerPC architecture.
- Protocol allows wait states to be inserted during the data phase and supports early burst termination.
- Supports both 16-bit and 32-bit port sizes.
- Bus electrical specification minimizes system power consumption.

## 5.4.2 External Bus Signals

**Table 5-5** summarizes the E-bus signals. The following abbreviations are used in this table:

M = Bus master  
 S = Slave device  
 A = Central bus arbiter  
 T = Bus watchdog timer  
 X = Any device on the system



**Table 5-5 EBI Signal Descriptions**

Mnemonic	Direction	Description
<b>Address Phase Signals</b>		
ADDR[0:29]	M → S	32-bit address bus. Least significant two bits (ADDR[30:31]) are not pinned out; they can be determined from the $\overline{BE}[0:3]$ pins. ADDR0 is the most significant bit. Address bus is driven by the bus master to index the bus slave.
$\overline{TS}$	M → S	Transfer start. This address control signal is asserted for one clock cycle at the beginning of a bus access by the bus master.
$\overline{WR}$	M → S	Write/read. When this address attribute is asserted, a write cycle is in progress. When negated, a read cycle is in progress. For use of $\overline{WR}$ during show cycles, refer to <a href="#">SECTION 8 DEVELOPMENT SUPPORT</a> .
$\overline{BE}[0:3]$	M → S	Byte enables. These address attribute signals indicate which byte within a word is being accessed. External memory chips can use these signals to determine which byte location is enabled. <a href="#">Table 5-7</a> shows the encodings for these pins during accesses to 32-bit and 16-bit ports.  A device need only observe the byte enables corresponding to the data lanes on which it resides. For example, a device on data lane DATA[0:7] should use $\overline{BE}0$ , and a device on DATA[0:15] should use $\overline{BE}[0:1]$ . The device should not respond to the bus cycle unless its byte enables are active at the start of the bus cycle.
AT[0:1]	M → S	Address types. These address attribute signals define addressed space as user or supervisor, data or instruction. Refer to <a href="#">Table 5-6</a> for encodings. These signals have the same timing as ADDR[0:29].
CT[0:3]	M → S	Cycle type signals. These address attribute signals indicate what type of bus cycle the bus master is initiating. Used for development support. Refer to <a href="#">Table 5-14</a> for encodings.
$\overline{BURST}$	M → S	Burst cycle. This address attribute indicates that the transfer is a burst transfer. If a burst access is burst-inhibited by the slave, the $\overline{BURST}$ pin is driven during each single-beat (decomposed) cycle.
$\overline{AACK}$	S → M	Address acknowledge. When asserted, indicates the slave has received the address from the bus master. This signal terminates the address phase of a bus cycle. When the bus master receives this signal from the slave, the master can initiate another address transfer. This signal must be asserted at the same time or prior to $\overline{TA}$ assertion.

**Table 5-5 EBI Signal Descriptions (Continued)**



Mnemonic	Direction	Description
$\overline{\text{ARETRY}}$	S, A $\rightarrow$ M	Address retry. This is an address phase termination signal. It is designed to resolve deadlock cases on hierarchical bus structures or for error-correcting memories. $\overline{\text{ARETRY}}$ assertion overrides $\overline{\text{AACK}}$ assertion and causes the SIU to re-arbitrate and to re-run the bus cycle.
$\overline{\text{BI}}$	S $\rightarrow$ M	Burst inhibit. When asserted, indicates the slave does not support burst mode. Sampled at same time as $\overline{\text{AACK}}$ . If $\overline{\text{BI}}$ is asserted, the SIU transfers the burst data in multiple cycles and increments the address for the slave in order to complete the burst transfer.
<b>Data Phase Signals</b>		
DATA[0:31]	M $\leftrightarrow$ S	32-bit data bus. DATA0 is most significant bit; DATA31 is the least significant bit. During small-port accesses, data resides on DATA[0:15].
$\overline{\text{BDIP}}$	M $\rightarrow$ S	Burst data in progress. This signal is asserted at the beginning of a burst data phase and is negated during the last beat of a burst. The master uses this signal to give the slave advance warning of the remaining data in the burst. This can also be used for an early termination of a burst cycle.  When the LST bit in the SIUMCR is asserted, the $\overline{\text{BDIP}}$ pin uses $\overline{\text{LAST}}$ timing. If the LST bit is negated, the $\overline{\text{BDIP}}$ pin uses $\overline{\text{BDIP}}$ timing. Refer to <a href="#">5.5.16.6 Synchronous Burst Interface</a> for more information.
$\overline{\text{TA}}$	S $\rightarrow$ M	Transfer acknowledge. When asserted, indicates the slave has received the data during a write cycle or returned the data during a read cycle. During burst cycles, the slave asserts this signal with every data beat returned or accepted.
$\overline{\text{TEA}}$	T, S $\rightarrow$ M	Transfer error acknowledge. Assertion of $\overline{\text{TEA}}$ indicates an error condition has occurred during the bus cycle, and the bus cycle is terminated. This signal overrides any other cycle termination signals (e.g., $\overline{\text{TA}}$ or $\overline{\text{ARETRY}}$ ).
$\overline{\text{DS}}$	M $\rightarrow$ S	Data strobe. Asserted by EBI at the end of a chip-select-controlled bus cycle. Asserted after the chip-select unit asserts the internal $\overline{\text{TA}}$ or $\overline{\text{TEA}}$ signal or the bus monitor timer asserts the internal $\overline{\text{TEA}}$ signal. Also asserted at the end of a show cycle.
<b>Arbitration</b>		
$\overline{\text{BR}}$	M $\rightarrow$ A	Bus request. When asserted, indicates the potential bus master is requesting the bus. Each master has its own bus request signal.
$\overline{\text{BG}}$	A $\rightarrow$ M	Bus grant. When asserted by bus arbiter, the bus is granted to the bus master. Each master has its own bus grant signal.
$\overline{\text{BB}}$	M $\rightarrow$ M, A	Bus busy. Asserted by current bus master to indicate the bus is currently in use. Prospective new master should wait until the current master negates this signal.
<b>Miscellaneous</b>		
$\overline{\text{CR}}$	X $\rightarrow$ M	Cancel reservation. Each PowerPC CPU has its own $\overline{\text{CR}}$ signal. This signal shows the status of any outstanding reservation on the external bus. When asserted, $\overline{\text{CR}}$ indicates that there is no outstanding reservation. This is a level signal.

**Table 5-5 EBI Signal Descriptions (Continued)**

Mnemonic	Direction	Description
$\overline{\text{RESET}}$	Source $\rightarrow$ M	This input-only signal resets the entire MCU. While $\overline{\text{RESET}}$ is asserted, the MCU asserts the $\overline{\text{RESETOUT}}$ signal.
$\overline{\text{RESETOUT}}$	M $\rightarrow$ S	Reset output. This output-only signal indicates that the MCU is in reset. When asserted, instructs all devices monitoring this signal to reset all parts within themselves that can be reset by software.
CLKOUT	Source $\rightarrow$ M, S	Continuously-running clock. All signals driven on the E-bus must be synchronized to the rising edge of this clock.

**Table 5-6 Address Type Encodings**

AT0	AT1	Address Space
0	0	User data space
0	1	User instruction space
1	0	Supervisor data space
1	1	Supervisor instruction space

**Table 5-7 Byte Enable Encodings**

Byte Enable	Use During 32-Bit Port Access	Use During 16-Bit Port Access
$\overline{\text{BE0}}$	Byte enable for DATA[0:7]	Byte enable for DATA[0:7]
$\overline{\text{BE1}}$	Byte enable for DATA[8:15]	Byte enable for DATA[8:15]
$\overline{\text{BE2}}$	Byte enable for DATA[16:23]	ADDR30
$\overline{\text{BE3}}$	Byte enable for DATA[24:31]	0 = Operand size is word 1 = Operand size is byte or half word

### 5.4.3 Basic Bus Cycle

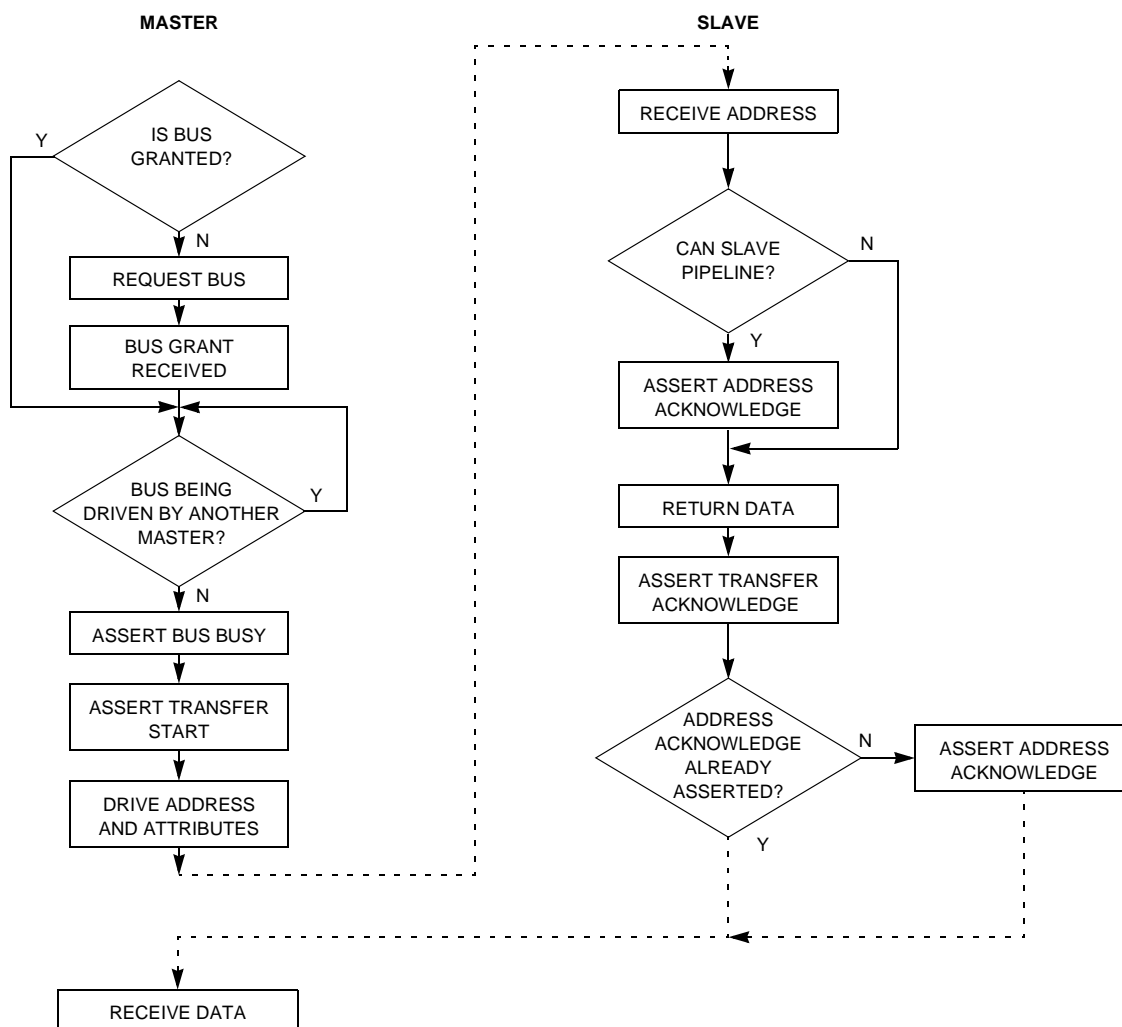
The basic external bus cycle consists of two phases: the address phase and the data phase. If the external bus is not available when the SIU is ready to start an external cycle, a bus arbitration phase is also required.

External bus cycles can be single or multiple (burst) data cycles. Burst cycles normally have four data words associated with the cycle. Refer to [5.4.6 Burst Cycles](#) for information on burst cycles.

#### 5.4.3.1 Read Cycle Flow

[Figure 5-4](#) is a flow diagram of a single read cycle on the external bus.





**Figure 5-4 Flow Diagram of a Single Read Cycle**

**Figure 5-5** is a simplified timing diagram of a read cycle.



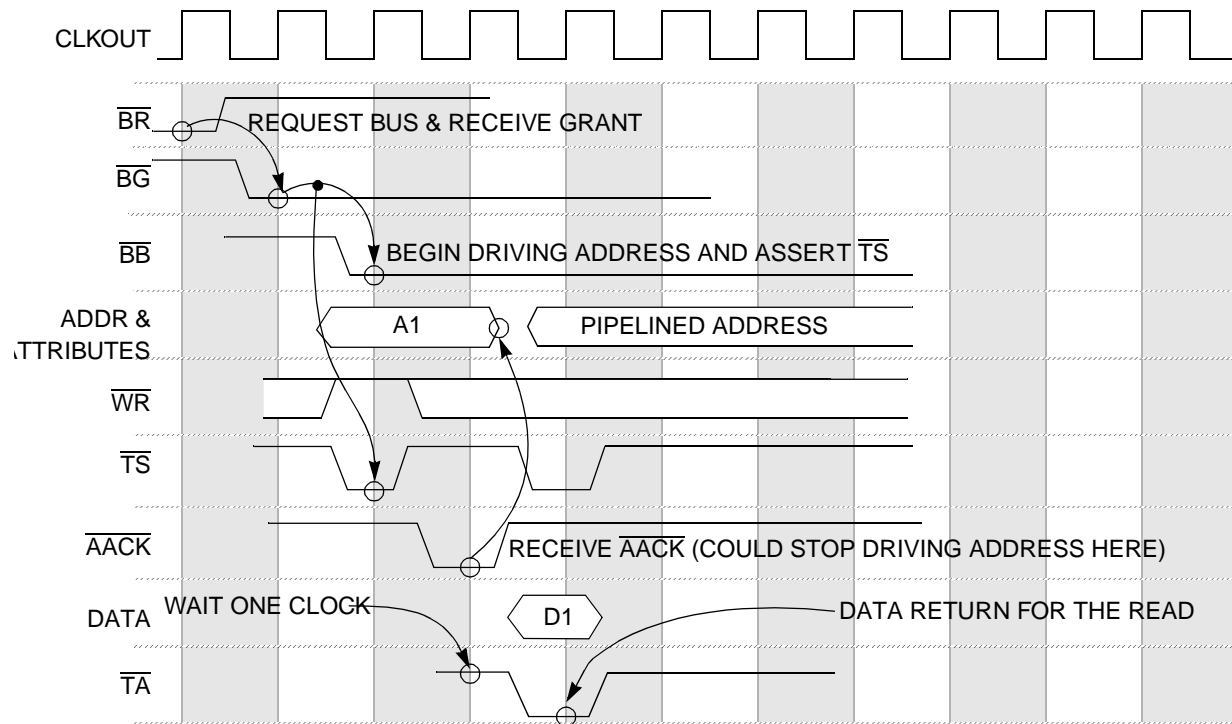
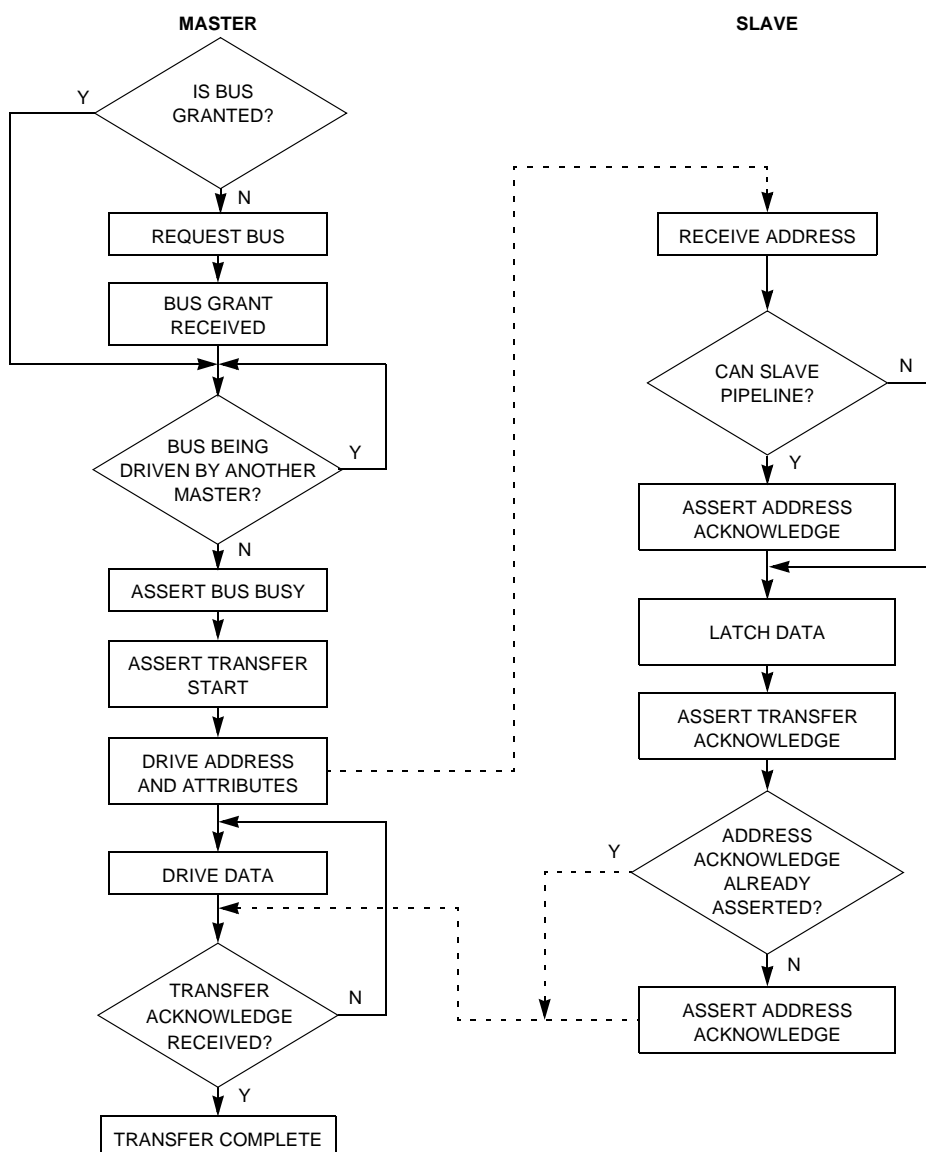


Figure 5-5 Example of a Read Cycle

#### 5.4.3.2 Write Cycle Flow

Figure 5-6 is a flow diagram of a single write cycle on the external bus.



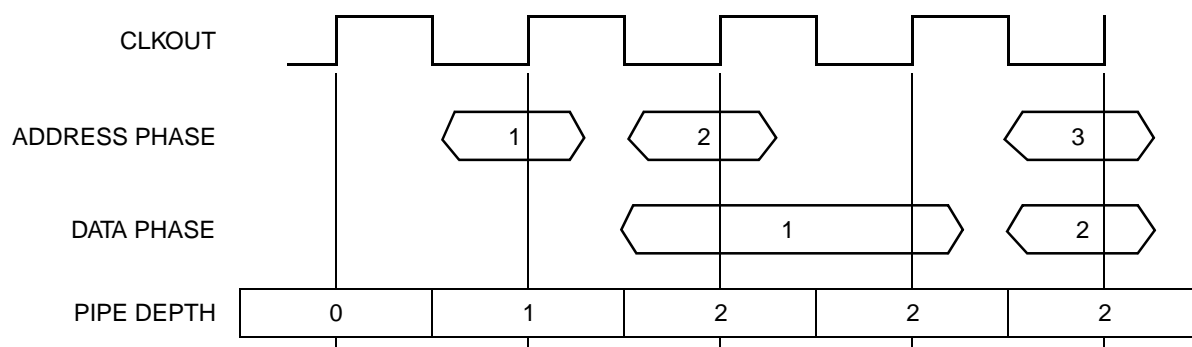
**Figure 5-6 Flow Diagram of a Single Write Cycle**

#### 5.4.4 Basic Pipeline

The EBI supports a maximum pipeline depth of two; that is, up to two addresses can be active on a bus at the same time. Pipelining is simplified by using SIU chip selects, since chip-select registers can have the information about the characteristics of each external memory. [5.5 Chip Selects](#) discusses which cycles can be pipelined.

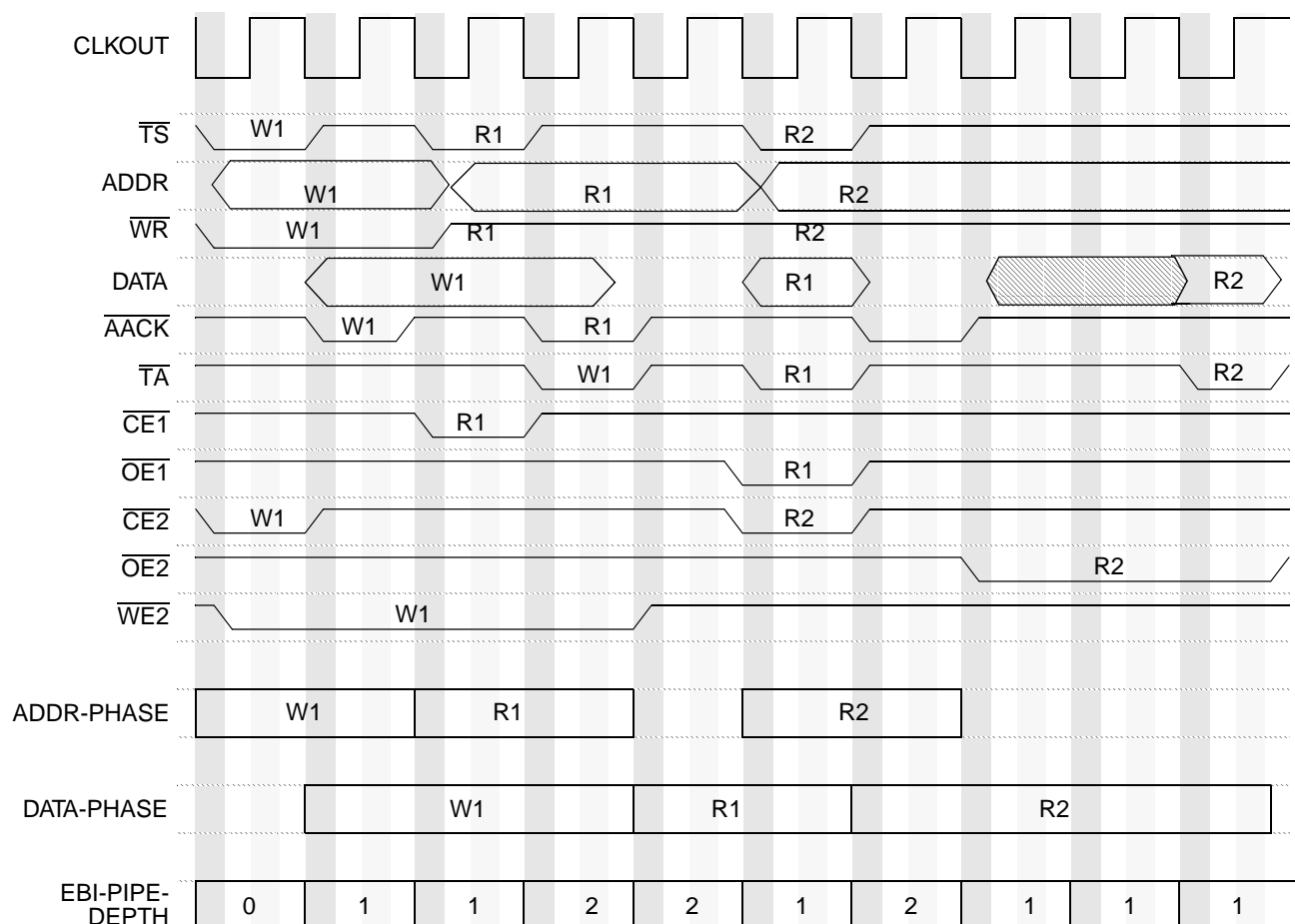
The EBI supports pipelined accesses for read cycles only. A write bus cycle starts only when the pipe depth is zero, or would have gone to zero if a new cycle had not started. A read bus cycle will start when the pipe depth is either zero or one, or would have gone to one if a new cycle had not started.

An example of bus pipelining is shown in [Figure 5-7](#).



**Figure 5-7 Example of Pipelined Bus**

**Figure 5-8** illustrates a write access followed by two read accesses on the external bus.



**Figure 5-8 Write Followed by Two Reads on the E-Bus (Using Chip Selects)**

## 5.4.5 Bus Cycle Phases

The following paragraphs describe the three bus cycle phases: arbitration phase, address phase, and data phase.



### NOTE

There is no separate arbitration for the address and data buses.

#### 5.4.5.1 Arbitration Phase

The SIU supports multiple masters but is optimized for single-master systems. Each master must have bus request, bus grant, and bus busy signals. Arbitration signals of the masters feed into a central arbiter for arbitration.

Before the SIU can start an external cycle, it must have a qualified bus grant. A qualified bus grant occurs when the external arbiter asserts  $\overline{BG}$  (bus grant) and the previous bus master negates  $\overline{BB}$  (the bidirectional bus busy signal). This means that no other master is currently running a cycle on the external bus. If the SIU is ready to start an external cycle and it does not have a qualified bus grant, then it asserts  $\overline{BR}$  (bus request) until it receives the qualified bus grant. Once the SIU receives a qualified bus grant, it asserts  $\overline{BB}$  and begins the address phase of the cycle.

A word-aligned access to a 16-bit port results in two bus cycles. To preserve word coherency, the SIU does not release the bus between these two cycles.

The external arbiter can park the bus by keeping  $\overline{BG}$  asserted. Single-master systems should tie this signal low permanently, or configure the pin as a port pin, which has the same effect. Each potential master has its own  $\overline{BG}$  input signal.

#### 5.4.5.2 Address Phase

Once the SIU has a qualified bus grant, it asserts  $\overline{BB}$  and starts an address phase. The SIU drives a new address at the start of the address phase and maintains it on the pins throughout the address phase.

The signals shown in [Table 5-8](#) are driven at the start of the address phase.

**Table 5-8 Signals Driven at Start of Address Phase**

Mnemonic	Signal Name	Type
ADDR[0:29]	Address bus	Address bus
$\overline{TS}$	Transfer start	Control
$\overline{WR}$	Write/read	Address attribute
$\overline{BE}[0:3]$	Byte enables	Address attribute
AT[0:1]	Address type	Address attribute
CT[0:3]	Cycle type	Address attribute
$\overline{BURST}$	Burst	Address attribute

$\overline{TS}$  is a control signal that is valid for only one clock cycle at the start of the address phase. The address attributes listed in [Table 5-8](#) are updated at the start of the address phase and are maintained until the start of the next address phase.



The address phase is the period of time from the assertion of  $\overline{TS}$  until the address phase is terminated by one of the following signals:

- Address acknowledge ( $\overline{AACK}$ )
- Address retry ( $\overline{ARETRY}$ )
- Transfer error acknowledge ( $\overline{TEA}$ )

If the external memory is under chip-select control and the chip selects are enabled to return handshakes, then the chip selects normally generate  $\overline{AACK}$  internally. However, if the external  $\overline{AACK}$  pin is asserted before the chip select module generates the signal, the chip select module accepts the external pin information and does not generate the  $\overline{AACK}$  signal internally.

Burst inhibit ( $\overline{BI}$ ) is sampled when  $\overline{AACK}$  is asserted.  $\overline{BI}$  is asserted by the slave to indicate to the SIU that the addressed device does not have burst capability. Refer to [5.4.6.2 Burst Inhibit Cycles](#) for more information.

$\overline{ARETRY}$  and  $\overline{TEA}$  can also be used to terminate the address phase. Refer to [5.4.10 Address Retry](#) and [5.4.11 Transfer Error Acknowledge Cycles](#) for more information.

### 5.4.5.3 Data Phase

If the pipe depth before a cycle starts is zero (or would have gone to zero if the new cycle had not started), then the data phase always starts one clock cycle after the address phase starts. If there is a previous data phase in progress one clock after an address phase starts, then the data phase for that address phase starts as soon as the previous data phase completes. The data phase completes when it is terminated by  $\overline{TA}$  or  $\overline{TEA}$ . If the cycle is a burst cycle, then multiple  $\overline{TA}$  assertions are required to terminate the data phase.

During the data phase, the following signals are used:

- DATA[0:31]
- Burst data in progress ( $\overline{BDIP}$ )

The data phase can be terminated with either of the following signals:

- Transfer acknowledge ( $\overline{TA}$ )
- Transfer error acknowledge ( $\overline{TEA}$ )

$\overline{AACK}$  and  $\overline{TA}$  are required for every cycle. If, under some error condition, the slave asserts  $\overline{TA}$  but not an  $\overline{AACK}$ , the SIU does not recover from this error condition.

If the external memory is under chip-select control and the chip selects are programmed to return handshakes (ACKEN = 1 in the chip-select option registers), then the chip selects return  $\overline{TA}$  unless the external  $\overline{TA}$  pin is asserted first. In that case, the

chip select module accepts the external pin information and does not generate  $\overline{TA}$  internally.



A bus timer or system address protection mechanism can assert transfer error acknowledge ( $\overline{TEA}$ ) to terminate the data phase when a bus error condition is encountered. Refer to **5.4.11 Transfer Error Acknowledge Cycles** for further information.

The EBI asserts the data strobe ( $\overline{DS}$ ) signal at the end of a chip-select controlled bus cycle, provided that either

- The chip select unit asserts the internal  $\overline{TA}$  signal; or
- The bus monitor timer asserts the internal  $\overline{TEA}$  signal.

$\overline{DS}$  is not asserted, however, if  $\overline{TA}$  or  $\overline{TEA}$  is asserted externally, even if  $\overline{TA}$  or  $\overline{TEA}$  is simultaneously asserted internally.

In addition to being asserted at the end of the bus cycles mentioned above,  $\overline{DS}$  is asserted at the end of a show cycle.

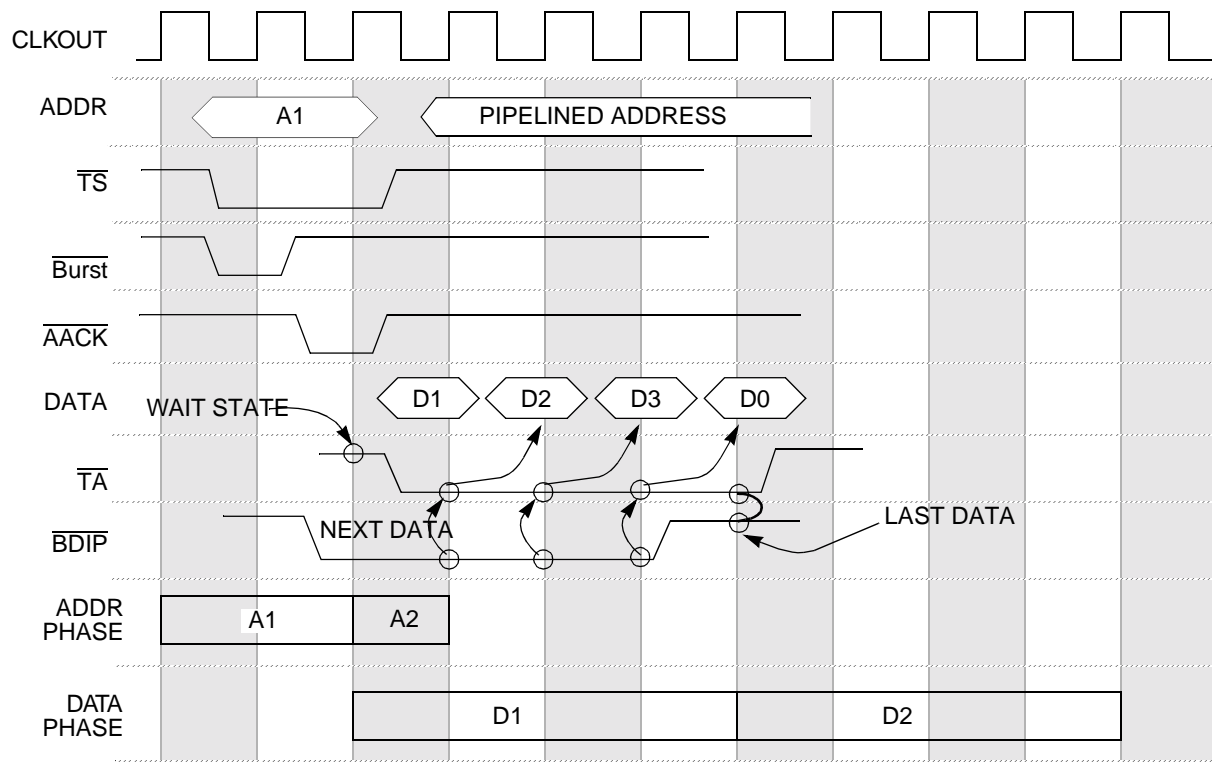
### 5.4.6 Burst Cycles

Burst cycles allow the fast transfer of instructions over the bus. The MPC509 supports fixed-length burst cycles of four beats for instruction reads only. Burst cycles can be terminated early with the  $\overline{BDIP}$  signal.

Burst reads on the external bus do not start until the internal data bus is available. For example, when the SIU starts a burst read and does not have L-bus data bus grant because there is an L-bus cycle in progress (an IMB2 access), then the SIU holds off the burst read until it can guarantee that it will have the internal bus grant.

At the start of a burst transfer, the master drives the address, the address attributes, transfer start, and the  $\overline{BURST}$  signal to indicate a burst transfer. If the slave can perform burst transfers, it negates the burst inhibit signal ( $\overline{BI}$ ). If the slave does not support burst transfers, it asserts  $\overline{BI}$ .

An example of a burst read on the external bus is shown in **Figure 5-9**.



**Figure 5-9 External Burst Read Cycle**

#### 5.4.6.1 Termination of Burst Cycles

During the data phase of a burst read cycle, the master receives data from the addressed slave. The EBI asserts the  $\overline{\text{BDIP}}$  signal at the beginning of a burst data phase and negates  $\overline{\text{BDIP}}$  during the last beat of a burst. The slave device stops driving new data after it receives the negation of  $\overline{\text{BDIP}}$  at the rising edge of the clock.

The EBI can terminate a burst cycle early by asserting the  $\overline{\text{BDIP}}$  pin. Early termination is used for a word aligned (not double-word aligned) burst to a small port.

The LST bit in the SIU module configuration register (SIUMCR) determines the timing used for the  $\overline{\text{BDIP}}$  pin. If LST is cleared, then the pin uses  $\overline{\text{BDIP}}$  timing. If the bit is set, the pin uses  $\overline{\text{LAST}}$  timing. The timing protocol of the external memory determines whether this bit should be set or cleared. Refer to [5.5.16.6 Synchronous Burst Interface](#) for examples of both types of timing.

Burst cycles can also be terminated with the  $\overline{\text{ARETRY}}$  signal. Refer to [5.4.10 Address Retry](#) for more information.

#### 5.4.6.2 Burst Inhibit Cycles

Burst inhibit ( $\overline{\text{BI}}$ ) is an address phase termination attribute that is sampled when  $\overline{\text{AACK}}$  is asserted. The slave asserts  $\overline{\text{BI}}$  to indicate to the SIU that the addressed device does not have burst capability. If this signal is asserted, the SIU transfers the data in multiple cycles and increments the address for the slave in order to complete the burst transfer.

A burst can only be burst inhibited until the first data is acknowledged ( $\overline{TA}$  asserted). Since  $\overline{BI}$  is not sampled until  $\overline{AACK}$  is asserted,  $\overline{AACK}$  must be asserted before or at the same time as  $\overline{TA}$ . Otherwise, the  $\overline{BI}$  pin is never sampled.



The EBI supports three types of memory. These memory types use the  $\overline{AACK}$  and  $\overline{BI}$  signals as follows:

- A simple asynchronous memory keeps  $\overline{AACK}$  negated to keep the address valid. The device can assert  $\overline{BI}$  along with  $\overline{AACK}$  or before  $\overline{AACK}$  and with the first  $\overline{TA}$ .
- A synchronous, pipelineable, non-burstable memory returns  $\overline{AACK}$  as soon as it is ready to receive the next address and asserts  $\overline{BI}$ .
- A burstable memory returns  $\overline{AACK}$  and negates  $\overline{BI}$  along with  $\overline{AACK}$  or before  $\overline{AACK}$ .

### CAUTION

If a memory region is under chip-select control, the chip-select unit generates  $\overline{BI}$  internally during burst accesses to interface types that do not support burst accesses. It is recommended that the  $\overline{BI}$  pin not be asserted during accesses to memory regions controlled by chip selects; instead, the chip-select unit will generate the  $\overline{BI}$  signal internally when appropriate.

#### 5.4.7 Decomposed Cycles and Address Wrapping

If a burst cycle initiated by one of the internal buses is burst inhibited by the chip selects or by the pins, the EBI decomposes this cycle into four single beat accesses. The EBI increments the address internally and sends the received data (from the four single external reads) back to the originating bus as a burst transaction.

The EBI breaks a burst access to a device with a 16-bit port into two or three cycles, depending on the starting address (or eight cycles if  $\overline{BI}$  is asserted) and increments the address appropriately. Examples of burst access address wrapping are shown in [Table 5-9](#). If a burst access to a device with a 16-bit port is burst inhibited by the chip selects or by external memory asserting the  $\overline{BI}$  pin, the EBI decomposes the transfer into eight single-beat accesses.

Depending on the starting address for the burst access and whether the address is word- or double-word-aligned, the EBI wraps the address to fetch the correct data from memory (four words or eight half words).

If the EBI receives  $\overline{TEA}$  for one part of a decomposed cycle, it generates  $\overline{TEA}$  internally for the remaining parts of the decomposed cycle as well.





**Table 5-9 Burst Access Address Wrapping**

Port Size	Starting Address ADDR[28:30]	Burst Address Wrapping ADDR[28:30]	Half-Word/Word Boundary Address
16 bit	000	000 (Starting address) 001 010 011  100 101 110 111	Double word boundary, Two bursts of four beats each
16 bit	010	010 (Starting address) 011  100 101 110 111  000 001	Odd word boundary, One burst of two beats One burst of four beats One burst of two beats The master (the EBI) terminates the two-beat-burst with <b>BDIP</b> .
32 bit	000	000 (Starting address) 010 100 110	Quad word boundary One burst of four beats
32 bit	100	100 (Starting address) 110 000 010	Double (non-quad) word boundary One burst of four words (word 3-4-1-2)

### 5.4.8 Preventing Speculative Loads

The SIU can be programmed to prevent speculative loads to a selected external region. A speculative operation is one which the hardware performs out of order and which it otherwise might not perform, such as executing an instruction following a conditional branch.

#### NOTE

The MPC509 never performs speculative stores; it always waits until the instruction is ready to be retired before writing to external memory.

When data loaded speculatively from RAM later needs to be discarded, this does not ordinarily present a problem. For example, a load instruction that follows a floating-point instruction in the instruction stream could begin execution before the floating-point instruction is retired. If the floating-point instruction generates an exception, the result of the load instruction is discarded, the exception is processed, and the processor automatically re-issues the load instruction.

However, if the address of the speculative load represents a FIFO device, the speculatively loaded data is lost when the exception is processed, and the re-issued load instruction loads the next data item in the queue. Preventing speculative loads is necessary to prevent this scenario from occurring.



As another example, a memory-mapped I/O device could have a status register that is updated whenever its data register is read. If the data register is read speculatively, the status register is updated, even if the result of the read is subsequently discarded (for example, if a previously issued instruction generates an exception).

Two registers and their associated logic allow a block ranging in size from 1 Kbyte to 64 Kbytes, or parts of the block, to be protected from speculative accesses. The most significant 22 bits of the address of each L-bus cycle are bitwise compared to the non-speculative base address register (SPECADDR), with each result bit equal to one if the bits match. A bitwise OR is performed on the lower six bits of the resulting word with the mask in the non-speculative mask register (SPECMASK). If all six result bits are ones and the upper 16 result bits are all ones, then speculative accesses are prevented during the current cycle.

## SPECADDR — Non-Speculative Base Address Register 0x8007 FC24

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
BASE ADDRESS																						RESERVED									
RESET:																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**Table 5-10 SPECADDR Bit Settings**

Bit(s)	Name	Description
0:21	BASE ADDRESS	22-bit base address of region protected from speculative loads.
22:31	—	Reserved

## SPECMASK — Non-Speculative Mask Register 0x8007 FC28

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED																MASK						RESERVED									
RESET:																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



**Table 5-11 SPECMASK Bit Settings**

Bit(s)	Name	Description
0:15	—	Reserved
16:21	MASK	Six-bit mask that specifies which block or blocks within region specified in SPECMASK register are actually protected from speculative accesses.
22:31	—	Reserved

Because the mask register can contain any six-bit value, the mask can allow for blocks of up to 64 Kbytes, and it can provide for smaller blocks of memory that alternately allow and prevent speculative loads. [Table 5-12](#) provides several examples. In these examples, the protected blocks are those that match the value in the SPECADDR register.

**Table 5-12 Example Speculative Mask Values**

Mask Value (Binary)	Protected Region
000000	1-Kbyte block
111111	64-Kbyte block
111110	Every second 1-Kbyte block within a 64-Kbyte block
111101	Every second 2-Kbyte block within a 64-Kbyte block
110011	Every fourth 4-Kbyte block within a 64-Kbyte block
100011	Every eighth 4-Kbyte block within a 64-Kbyte block
010000	Every sixteenth 1-Kbyte block within a 32-Kbyte block

Protection from speculative loads can be disabled by setting the SPECADDR register to an internal or unimplemented address range.

#### 5.4.9 Accesses to 16-Bit Ports

The EBI supports accesses to 16-bit ports on the external bus. 16-bit port size is a chip-select option; the access must be initiated using one of the chip selects. A 16-bit port must connect its data lines to the upper 16 bits of the external data bus (DATA[0:15]).

During an access to a 16-bit port, byte enable signals  $\overline{BE}[0:1]$  are used to indicate which bytes of the half-word are being accessed, and the  $\overline{BE2}/ADDR30$  pin functions as ADDR30 (active high).  $\overline{BE3}$  is asserted (low) if the operand size is a word and negated (high) if the operand size is a byte or half-word. This encoding is needed to maintain coherency of word accesses on the external bus.

[Table 5-13](#) shows how EBI decomposes the word, half word, and byte accesses to a 16-bit port. For each combination of operand size and address (placement on the inter-

nal L-data bus), the table shows the values of  $\overline{BE}[0:3]$  and indicates which bytes of the operand are accessed and where these bytes are placed on the E-bus.



**Table 5-13 EBI Read and Write Access to 16-Bit Ports**

Operand Size	Internal L-Bus ADDR[30:31]	Internal L-Bus Data Bytes Accessed	$\overline{BE}[0:3]$	Placement of L-Bus Data Bytes On E-Bus	
				E-Bus DATA[0:7]	E-Bus DATA[8:15]
Byte	00	Byte 0	0101	Byte 0	X
Byte	01	Byte 1	1001	X	Byte 1
Byte	10	Byte 2	0111	Byte 2	X
Byte	11	Byte 3	1011	X	Byte 3
Half Word	00	Bytes 0 to 1	0001	Byte 0	Byte 1
Half Word	10	Bytes 2 to 3	0011	Byte 2	Byte 3
Word	00	Bytes 0 to 3	0000 0010	Byte 0 Byte 2	Byte 1 Byte 3

All transfer errors that occur during a small port access terminate the cycle currently in progress. If an error occurs during any part of a word access to a small port, the current access is terminated. Subsequent bus cycles of the small port access will continue, but  $\overline{TEA}$  will be asserted internally with each beat.

#### 5.4.10 Address Retry

Address retry ( $\overline{ARETRY}$ ) can be used to terminate the address phase. Assertion of  $\overline{ARETRY}$  causes the master to re-arbitrate and to re-run the bus cycle. The address retry mechanism can be used to break deadlocks between the E-bus and the user's on-board I/O bus (for example, a PC/AT or VME bus in a hierarchical bus system). The address retry mechanism can also be used for error correction purposes.

After receiving  $\overline{TS}$ , the external device must wait at least one clock cycle before asserting  $\overline{ARETRY}$ . Note that this could be an issue at low frequencies — it is possible for an external device to receive  $\overline{TS}$ , decode the address, and assert  $\overline{ARETRY}$  in the same clock cycle. This is illegal.

The SIU does not guarantee word coherency if  $\overline{ARETRY}$  is asserted for the second half of a word cycle of a decomposed word transfer. The external arbiter is responsible for maintaining the coherency by monitoring the byte enable lines and making sure that no other master updates that location until the retried cycle is successfully completed.

Note that  $\overline{BB}$  is not negated until the second clock cycle after  $\overline{ARETRY}$  assertion.

#### CAUTION

$\overline{TA}$  or  $\overline{TEA}$  must not be asserted during a cycle in which  $\overline{ARETRY}$  is asserted. If  $\overline{TA}$  is asserted for any part of a burst cycle,  $\overline{ARETRY}$

must not be asserted at any time during the cycle; if  $\overline{\text{ARETRY}}$  is asserted during a burst cycle, it must be asserted before the first beat is terminated with  $\overline{\text{TA}}$ .



#### 5.4.11 Transfer Error Acknowledge Cycles

A bus timer or system address protection mechanism can assert transfer error acknowledge ( $\overline{\text{TEA}}$ ) to terminate the data phase when one of the following types of bus error conditions is encountered:

- Write to a read-only address space
- Access to a non-existent address

$\overline{\text{TEA}}$  assertion overrides the assertion of  $\overline{\text{TA}}$ . Assertion of  $\overline{\text{TEA}}$  causes the processor to enter the checkstop state, enter debug mode, or process a machine check exception. Refer to the [RCPU Reference Manual](#) (RCPURM/AD) for details.

#### CAUTION

$\overline{\text{TEA}}$  must not be asserted during a cycle in which  $\overline{\text{ARETRY}}$  is asserted.

If the address phase corresponding to the current data phase is still outstanding ( $\overline{\text{AACK}}$  has not yet been asserted),  $\overline{\text{TEA}}$  terminates both the address and the data phase. That is, the EBI generates  $\overline{\text{AACK}}$  and  $\overline{\text{TA}}$  internally and generates an internal error signal for that cycle. If  $\overline{\text{AACK}}$  has already been asserted externally, the EBI generates  $\overline{\text{TA}}$  but not  $\overline{\text{AACK}}$  internally and generates an internal error signal for that cycle.

All transfer errors that occur during an access to a 16-bit port terminate the cycle currently in progress. If an error occurs during any part of a word access to a 16-bit port, the current access is terminated. Subsequent bus cycles of the small port access will continue, but  $\overline{\text{TEA}}$  will be asserted internally with each beat.

All illegal accesses to internal registers are terminated with a data error, causing the bus monitor to assert the internal  $\overline{\text{TEA}}$  signal. Accesses to unimplemented internal memory locations and privilege violations (user access to supervisor register or write to read-only location or a write to register which is locked) also cause the bus monitor to assert the internal  $\overline{\text{TEA}}$  signal.

Note that the chip-select module can also assert the internal  $\overline{\text{TEA}}$  signal. Refer to [5.5.7 Access Protection](#) for more information.

#### 5.4.12 Cycle Types

The cycle type pins (CT[0:3]) are address-phase signals that provide information about the type of internal or external bus cycle in progress. These pins can be used by an external development system to construct a program trace.

**Table 5-14** summarizes the cycle type encodings. Refer to the [RCPU Reference Manual](#) (RCPURM/AD) for details on how a development system can use the information provided by these pins.



**Table 5-14 Cycle Type Encodings**

CT[0:3]	Cycle Type	Description
0000	Normal bus cycle	This is a normal external bus cycle. Both the address and data phase are seen on the external bus. This cycle requires an $\overline{AACK}$ and a $\overline{TA}$ signal. This cycle type is used for sequential fetches and for prefetches of predicted branch targets where the branch condition has not been evaluated before the prefetch. It is also used for all non-reservation type load/store cycles.
0001	Reservation start if address type is data  OR Instruction fetch marked as indirect change-of-flow if address type is instruction	If the address type is data ( $AT1 = 0$ ), then this is a data access to the external bus. Both the address and the data phase are seen on the external bus. This cycle requires an $\overline{AACK}$ and a $\overline{TA}$ signal. When this cycle starts, external snooping logic should latch the address to track the reservation.  If the address type is instruction ( $AT1 = 1$ ), then this is an instruction fetch cycle marked as an indirect change-of-flow cycle. Both the address and the data phase are seen on the external bus. This cycle requires an $\overline{AACK}$ and a $\overline{TA}$ signal. This cycle type is used when an external address is the destination of a branch instruction or the destination of an exception or VSYNC cycle.
0010	Emulation memory select (not supported in MPC509)	This is a special external bus cycle to emulation memory replacing internal I-mem or L-mem (and not resulting in a cache hit). The MPC509 does not support this cycle type.
0011	PRU select (not supported in MPC509)	This is a normal external bus cycle access to a port replacement chip used for emulation support. Both the address and the data phase are seen on the external bus. This cycle requires an $\overline{AACK}$ and a $\overline{TA}$ signal. It indicates that an access was made which would have gone to an internal port control register if the chip were not operating in PRU mode.
0100	I-mem (not supported in MPC509)	These are internal visibility cycles. This cycle is self-terminating and does not require $\overline{AACK}$ and $\overline{TA}$ signals. These encodings indicate that an access or aborted fetch (resulting from either a cache hit or a speculative load that is subsequently discarded) was made to an address on the internal I-bus or L-bus. An instruction access ( $AT1 = 1$ ) with an address which is an indirect branch target is indicated as a write on the $\overline{WR}$ signal.  The I-Mem cycle type is not supported in the MPC509.
0101	L-mem	
0110	E-Mem (external memory) cache hit, not using a chip select	This is an internal visibility cycle. It always has an address phase and includes a data phase for data accesses. This cycle is self-terminating and does not require $\overline{AACK}$ and $\overline{TA}$ signals. It indicates that an access was made to an address on the external bus and that a cache hit or aborted fetch occurred. An instruction access with an address that is an indirect branch target is indicated as a write on the $\overline{WR}$ signal.
0111	Internal register	This is an internal visibility cycle. It always has an address phase and a data phase. This cycle is self-terminating and does not require $\overline{AACK}$ and $\overline{TA}$ signals. It indicates that an access was made to a control register or internal IMB2 address. These accesses are always cache-inhibited.

**Table 5-14 Cycle Type Encodings (Continued)**



CT[0:3]	Cycle Type	Description
1000	E-Mem cache hit to $\overline{\text{CSBOOT}}$ region	These are internal visibility cycles. They always have an address phase and include a data phase for data accesses. These cycles are self-terminating and do not require $\overline{\text{AACK}}$ and $\overline{\text{TA}}$ signals. These encodings indicate that an access was made to an address on the external bus and that a cache hit or aborted fetch occurred. An instruction access with an address that is an indirect branch target is indicated as a write on the $\overline{\text{WR}}$ signal.  The region indicated is the main chip-select region, not the sub-region.
1001	E-Mem cache hit to $\overline{\text{CS1}}$ region	
1010	E-Mem cache hit to $\overline{\text{CS2}}$ region	
1011	E-Mem cache hit to $\overline{\text{CS3}}$ region	
1100	E-Mem cache hit to $\overline{\text{CS4}}$ region	
1101	E-Mem cache hit to $\overline{\text{CS5}}$ region	
1110	Reserved	—
1111	Reserved	—

#### 5.4.13 Show Cycles

Internal bus cycles that are echoed on the external bus are referred to as show cycles. By providing access to bus cycles that are not visible externally during normal operation, show cycles allow a development support system to trace the flow of a program.

The LSHOW field in the SIUMCR can be programmed to cause the EBI to echo certain or all internal L-bus cycles on the external bus. Likewise, the ISCTL field in the ICTRL register (instruction bus control register, SPR 158) in the RCPU can be programmed to cause the EBI to echo certain or all internal I-bus cycles on the external bus.

The I-bus show cycles are always address-only cycles. They do not wait for the internal transaction to complete. L-bus show cycles have both address and data and appear on the external bus after the internal cycle is completed.

Aborted L-bus cycles do not result in a show cycle. (The load/store unit of the processor may abort the cycle when the previous cycle terminates with a transfer error, or when an exception occurs during the current cycle.)

Aborted I-bus cycles do result in a show cycle. (The processor may abort an I-bus cycle when it encounters a branch; it aborts the fetch just starting on a wrong path. In addition, the processor aborts the cycle on a cache hit.)

Note that I-bus show cycles are not burst.

A show cycle involves transfer start ( $\overline{\text{TS}}$ ), address (ADDR), cycle type (CT), address type (AT), burst ( $\overline{\text{BURST}}$ ) and read/write ( $\overline{\text{WR}}$ ) pins. The data phase of an L-bus show cycle looks like a write cycle going out on the external bus. The address and data phases of a show cycle last one clock cycle each. No termination is needed for either phase, as all show cycles are automatically terminated inside the SIU. For the L-bus

show cycles (I-bus show cycles are address only), the data phase always follows the address phase by one clock cycle. The L-bus show cycle does not start until the internal cycle completes. This allows all show cycles to complete in two clock cycles.



Show cycles require several holding registers in the SIU to hold address and data of an L-bus cycle and address of an I-bus cycle until the E-bus is available and the show cycle is run. When these holding registers are full, the internal bus (or buses) are held up by the SIU while it waits for the show cycle to complete.

During cross-bus accesses, the show cycle is associated with the bus initiating the transaction. For example, if I-bus show cycles are enabled and L-bus show cycles are disabled, then an instruction fetch from L-RAM will show up as an address-only I-bus show cycle, and an L-bus access to I-memory would not have a show cycle.

Refer to **SECTION 8 DEVELOPMENT SUPPORT** for more information on show cycles.

#### 5.4.14 Storage Reservation Support

The PowerPC **lwarx** (load word and reserve indexed) and **stwcx.** (store word conditional indexed) instructions in combination permit the atomic update of a storage location. Refer to the **RCPU Reference Manual** (RCPURM/AD) for details on these instructions.

The storage reservation protocol supports a multi-level bus structure like the one shown in **Figure 5-10**. In this figure, the E-bus is a PowerPC bus interfaced to a non-local bus, such as a PC/AT or VME bus, through a non-local bus interface. For each local bus, storage reservation is handled by the local reservation logic.

The protocol tries to optimize reservation cancellation such that a PowerPC processor is notified of the loss of a storage reservation on a remote bus only when it has issued a **stwcx.** instruction to that address. That is, the reservation loss indication comes as part of the **stwcx.** cycle. This method eliminates the need to have very fast storage reservation loss indication signals routed from every remote bus to every PowerPC master.



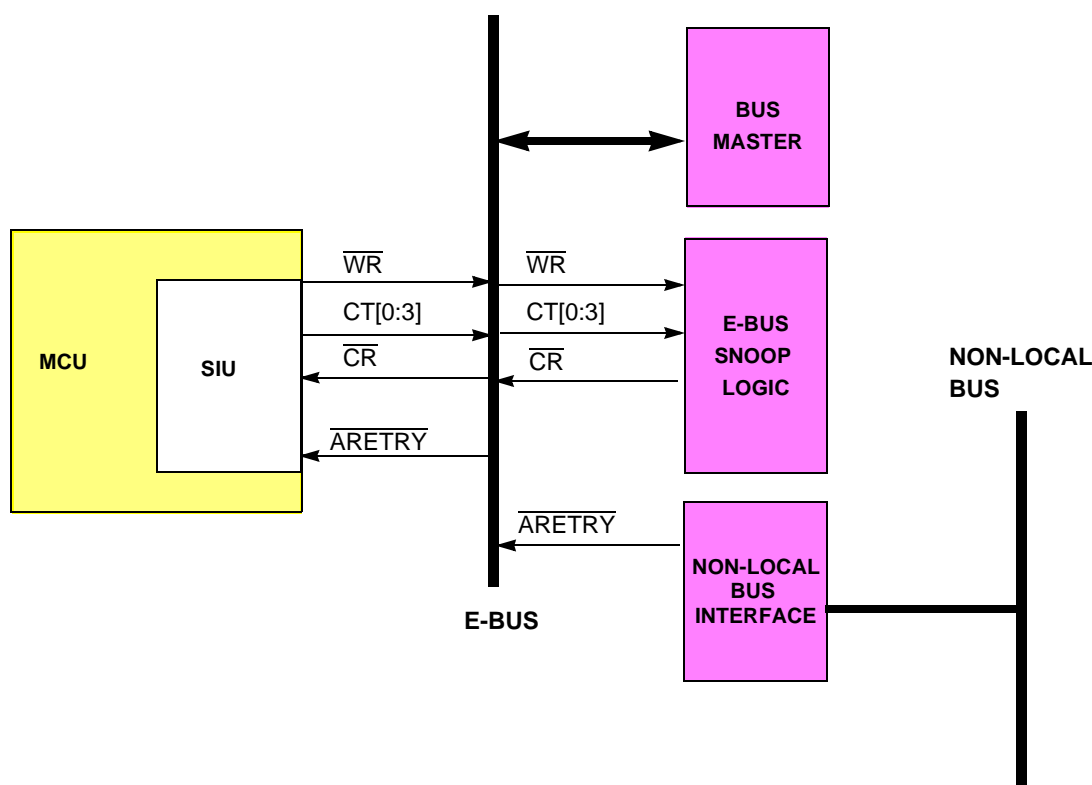


Figure 5-10 Storage Reservation Signaling

#### 5.4.14.1 PowerPC Architecture Reservation Requirements

The PowerPC architecture requires that the reservation protocol meets the following requirements:

- Each PowerPC processor has at most one reservation.
- The **lwarx** instruction establishes a reservation.
- The **lwarx** instruction by the same processor clears the first reservation and establishes a new one.
- The **stwcx.** instruction by the same processor clears the reservation.
- A normal store by the same processor does not clear the reservation.
- A normal store by some other processor (or other mechanism, such as a DMA) to an address with an existing reservation clears the reservation.
- If the storage reservation is lost, it is guaranteed that **stwcx.** instruction will not modify storage.
- The granularity of the address compare is a multiple of the coherent block size (which should be a multiple of four bytes).

#### 5.4.14.2 E-bus Storage Reservation Implementation

The E-bus reservation protocol requires local (external) bus reservation logic, if needed, to:



- Snoop accesses to all local bus slaves.
- Hold one reservation for each local master capable of storage reservations.
- Set the reservation when that master issues a load with reservation.
- Clear the reservation when some other master issues a store to the reservation address.
- Indicate the current status of the local bus reservation such that it may be sampled prior to the address phase of the **stwcx.** bus cycle. (The reservation must be set in time to enable a store to the reservation address, and must be cleared fast enough to disable a store to the reservation address).

The EBI samples the  $\overline{CR}$  pin prior to starting an external **stwcx.** cycle. If the reservation is cancelled ( $\overline{CR}$  is asserted), no cycle starts. If the reservation is not cancelled, the SIU begins the bus cycle.

If  $\overline{ARETRY}$  is asserted, the SIU must re-sample the  $\overline{CR}$  and  $\overline{BG}$  pins prior to performing the external retry.

If a reservation exists on a non-local bus, and the SIU begins a **stwcx.** cycle to that address on the local bus while the non-local bus reservation is cleared, the  $\overline{ARETRY}$  signal should be asserted to the SIU, and the reservation signal should be cleared before  $\overline{BG}$  is asserted to the SIU. This means that  $\overline{AACK}$  should not be returned until successful coherent completion of the **stwcx.** is ensured. The non-local bus interface must not perform the non-local write (or abort it if the bus supports aborted cycles) if it asserts  $\overline{ARETRY}$ .

#### NOTE

Single-master systems do not require an external reservation tracking logic. In these systems, the  $\overline{CR}$  pin should be tied by resistor to the reservation valid (high) state. Alternatively, the reservation pin may be configured as a port. If the reservation pin is configured as a port, the SIU will always consider the reservation to be valid.

#### 5.4.14.3 Reservation Storage Signals

Reservation storage signals used by the EBI are summarized in [Table 5-15](#).



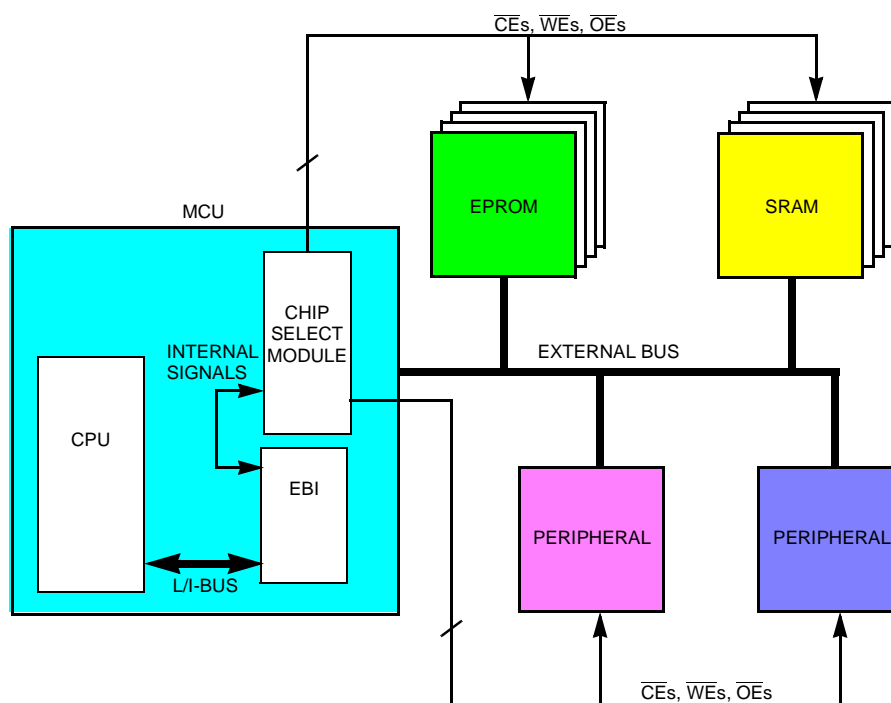
**Table 5-15 EBI Storage Reservation Interface Signals**

Name	Direction	Description
$\overline{CR}$	Snoop logic $\Rightarrow$ SIU	Cancel reservation. Each PowerPC CPU has its own $\overline{CR}$ signal. This signal shows the status of any outstanding reservation on the external bus. When asserted, $\overline{CR}$ indicates that there is no outstanding reservation. This is a level signal.
$\overline{ARETRY}$	Non-local bus interface $\Rightarrow$ SIU	Address retry. When asserted, indicates that the master needs to retry its address phase.  In case of an <b>stwcx.</b> cycle to a non-local bus on which the storage reservation has been lost, this signal is used by the non-local bus interface to back off the cycle.

## 5.5 Chip Selects

Typical microcontrollers require additional hardware to provide external chip-select signals. In the MPC500 family, the chip-select logic controls the slaves of typical uniprocessor systems. This allows the user to implement simple systems without the need to design any external glue logic.

**Figure 5-11** is an example of a typical uniprocessor system. This kind of system usually consists of a CPU, some memories, and some peripherals. In single-master systems the CPU is the only device that can be a bus master on the E-bus; memories and peripherals are slaves.



**Figure 5-11 Simplified Uniprocessor System with Chip-Select Logic**

The chip-select module provides the necessary control signals, such as the chip enable ( $\overline{CE}$ ), write enable ( $\overline{WE}$ ), and output enable ( $\overline{OE}$ ), for the external memory and peripheral devices. In addition, the chip-select module provides some handshakes for the external bus and some limited protection mechanisms for the system.

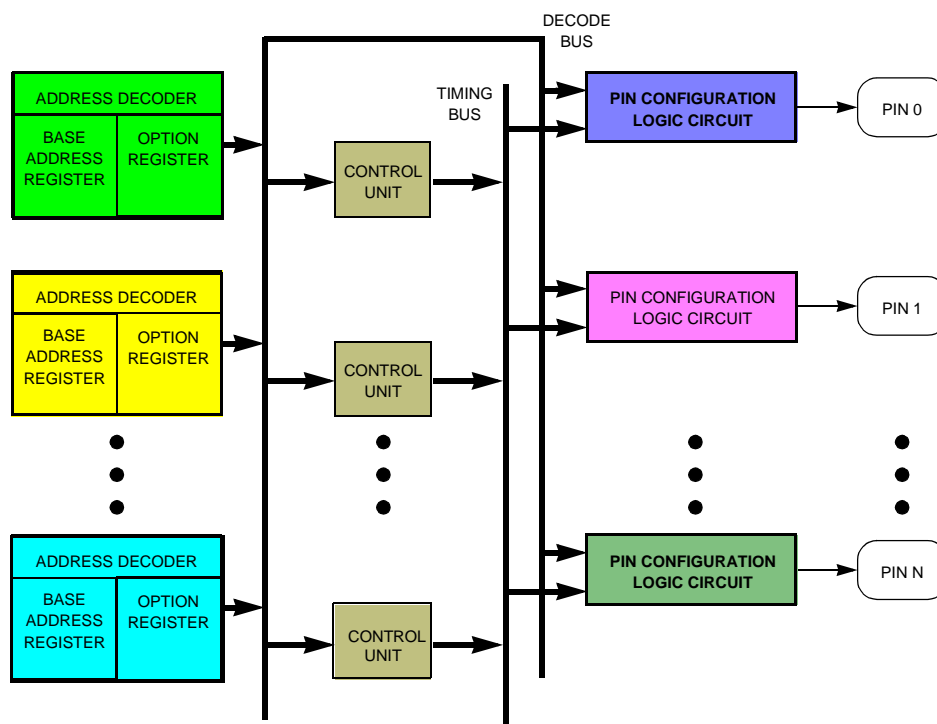


### 5.5.1 Chip-Select Features

- No external glue logic required for typical systems if the chip-select module is used
- Modular architecture for ease of expansion
- Twelve chip-select pins plus one  $\overline{CSBOOT}$  pin
- Pins can be programmed as  $\overline{CE}$ s (six maximum),  $\overline{OE}$ s, or  $\overline{WE}$ s
- Capable of supporting pipelineable, burstable devices
- Returns bus handshake signals for the selected address regions
- Provides up to seven programmable wait states for slave devices
- Controls the clocking of data to the slaves during write cycles
- Keeps slave sequentially consistent (data in the same order as addresses)
- Programmability for:
  - Latching and non-latching device types
  - Burstable and non-burstable device types
- Programmable address range and block size
- Programmable burst features:
  - Interruptible burst on any burstable device
  - Pipelineable with other devices during burst cycle
  - Supports two different burst protocols
- Supports pipelineable accesses
  - Up to two concurrent accesses can be outstanding to two different regions (one access to each region)
  - For two consecutive accesses to the same region, overlaps the address phase of the second access with the data phase of the first access
- Allows multi-level protection within a region. The  $\overline{CSBOOT}$  region can have up to two sub-levels of protection
- Supports both 16-bit and 32-bit port sizes

### 5.5.2 Chip-Select Block Diagram

**Figure 5-12** shows the functional block diagram of the chip-select module.



**Figure 5-12 Chip-Select Functional Block Diagram**

### 5.5.3 Chip-Select Pins

The pin configuration (PCON) field in each chip-select option register configures the associated pin to function as a chip enable ( $\overline{CE}$ ), write enable ( $\overline{WE}$ ), output enable ( $\overline{OE}$ ), or alternate-function pin. For pins configured for their alternate function, the port A pin assignment register configures the pin as either an address bus signal (ADDR[0:11]) or a port A or B output signal (PA[0:7] and PB[0:3]). Notice that the  $\overline{CSBOOT}$  pin has no alternate function.

**Table 5-16** describes the chip-select pins.

**Table 5-16 Chip-Select Pin Functions**

Chip-Select Function	Alternate Function	Pin Function in Chip-Select Mode
$\overline{CSBOOT}$	—	Can be the $\overline{CE}$ of the system boot memory (power-on default). In systems with no external boot device, this pin can be configured as $\overline{WE}$ or $\overline{OE}$ of EPROMs or SRAMs.
$\overline{CS0}/\overline{CSBTOE}$	ADDR0/PA0	Can be $\overline{WE}$ or $\overline{OE}$ of EPROMs or SRAMs. When configured as a chip select, this pin is assigned to be the $\overline{OE}$ of the $\overline{CSBOOT}$ pin following reset.
$\overline{CS1}$	ADDR1/PA1	Can be $\overline{CE}$ , $\overline{WE}$ , or $\overline{OE}$ of EPROMs or SRAMs.
$\overline{CS2}$	ADDR2/PA2	Can be $\overline{CE}$ , $\overline{WE}$ , or $\overline{OE}$ of EPROMs or SRAMs.

**Table 5-16 Chip-Select Pin Functions (Continued)**



Chip-Select Function	Alternate Function	Pin Function in Chip-Select Mode
$\overline{CS3}$	ADDR3/PA3	Can be $\overline{CE}$ , $\overline{WE}$ , or $\overline{OE}$ of EPROMs or SRAMs.
$\overline{CS4}$	ADDR4/PA4	Can be $\overline{CE}$ , $\overline{WE}$ , or $\overline{OE}$ of EPROMs or SRAMs.
$\overline{CS5}$	ADDR5/PA5	Can be $\overline{CE}$ , $\overline{WE}$ , or $\overline{OE}$ of EPROMs or SRAMs.
$\overline{CS6}$	ADDR6/PA6	Can be $\overline{WE}$ or $\overline{OE}$ of EPROMs or SRAMs.
$\overline{CS7}$	ADDR7/PA7	Can be $\overline{WE}$ or $\overline{OE}$ of EPROMs or SRAMs.
$\overline{CS8}$	ADDR8/PB0	Can be $\overline{WE}$ or $\overline{OE}$ of EPROMs or SRAMs.
$\overline{CS9}$	ADDR9/PB1	Can be $\overline{WE}$ or $\overline{OE}$ of EPROMs or SRAMs.
$\overline{CS10}$	ADDR10/PB2	Can be $\overline{WE}$ or $\overline{OE}$ of EPROMs or SRAMs.
$\overline{CS11}$	ADDR11/PB3	Can be $\overline{WE}$ or $\overline{OE}$ of EPROMs or SRAMs.

**NOTE**

During the first two clock cycles of power-on reset, the state of the pins listed in [Table 5-16](#) is unknown.

When a chip select is configured as a chip enable of a memory or I/O device, the MCU asserts the chip select when it drives the address onto the external bus. For non-pipelineable devices, the  $\overline{CE}$  is asserted until the access is completed. For pipelineable devices, when  $\overline{CE}$  is asserted the device should clock in the address at the rising edge of the clock. (Note that devices that the chip-select unit regards as pipelineable are always synchronous.)

The  $\overline{WE}$  signal is used during write accesses. When a chip select is configured as a write enable signal of a memory or I/O device, the MCU asserts the chip select as it drives data onto the external bus to signal the external device to strobe in the data. For synchronous devices, if  $\overline{WE}$  is asserted the device should clock in the data at the rising edge of the clock.

The  $\overline{OE}$  signal is used during read accesses. When the MCU asserts a chip-select signal that is configured as an output enable of a memory or I/O device, the device can drive its data onto the E-bus.

#### 5.5.4 Chip-Select Registers and Address Map

Chip-select registers are 32 bits wide. Reads of unimplemented bits in these registers return zero, and writes have no effect.

One base address register and one option register are associated with each chip-select pin that can function as a chip enable. The  $\overline{CSBOOT}$  pin has a dedicated sub-block for multi-level protection. It has two base address registers and two option registers. One option register is associated with each pin that can function as a write enable or output enable but not as a chip enable.

**Table 5-17** is an address map of the chip-select module. As the entries in the Access column indicate, all chip-select registers are accessible at the supervisor privilege level only.



When set, the LOK bit in the SIU module configuration register (SIUMCR) locks all chip-select registers to prevent software from changing the chip-select configuration inadvertently. Before changing the chip-select configuration, the user needs to ensure that this bit is cleared.

Note that if the processor is modifying the chip-select registers of a region and it needs the instructions from that region (a region that it is reconfiguring), software needs to ensure that the code is accessible elsewhere.

### EXAMPLE

If the processor is configuring the  $\overline{\text{CSBOOT}}$  registers and simultaneously executing instructions out of the boot region, software can re-locate the necessary code to the instruction cache, internal SRAM, or to another external region such as external SRAM, before modifying the chip-select control registers.



**Table 5-17 Chip-Select Module Address Map**

Access	Address	Register
—	0x8007 FD00 – 0x8007 FD90	Reserved
S	0x8007 FD94	$\overline{\text{CS11}}$ option register (CSOR11)
S	0x8007 FD98	Reserved
S	0x8007 FD9C	$\overline{\text{CS10}}$ option register (CSOR10)
S	0x8007 FDA0	Reserved
S	0x8007 FDA4	$\overline{\text{CS9}}$ option register (CSOR9)
S	0x8007 FDA8	Reserved
S	0x8007 FDAC	$\overline{\text{CS8}}$ option register (CSOR8)
S	0x8007 FDB0	Reserved
S	0x8007 FDB4	$\overline{\text{CS7}}$ option register (CSOR7)
S	0x8007 FDB8	Reserved
S	0x8007 FDBC	$\overline{\text{CS6}}$ option register (CSOR6)
S	0x8007 FDC0	$\overline{\text{CS5}}$ base address register (CSBAR5)
S	0x8007 FDC4	$\overline{\text{CS5}}$ option register (CSOR5)
S	0x8007 FDC8	$\overline{\text{CS4}}$ base address register (CSBAR4)
S	0x8007 FDCC	$\overline{\text{CS4}}$ option register (CSOR4)
S	0x8007 FDD0	$\overline{\text{CS3}}$ base address register (CSBAR3)
S	0x8007 FDD4	$\overline{\text{CS3}}$ option register (CSOR3)
S	0x8007 FDD8	$\overline{\text{CS2}}$ base address register (CSBAR2)
S	0x8007 FDDC	$\overline{\text{CS2}}$ option register (CSOR2)
S	0x8007 FDE0	$\overline{\text{CS1}}$ base address register (CSBAR1)
S	0x8007 FDE4	$\overline{\text{CS1}}$ option register (CSOR1)
S	0x8007 FDE8	Reserved
S	0x8007 FDEC	$\overline{\text{CS0}}$ option register (CSOR0)
S	0x8007 FDF0	$\overline{\text{CSBOOT}}$ sub-block base address register (CSBTSBBAR)
S	0x8007 FDF4	$\overline{\text{CSBOOT}}$ sub-block option register (CSBTSBOR)
S	0x8007 FDF8	$\overline{\text{CSBOOT}}$ base address register (CSBTBAR)
S	0x8007 FDFC	$\overline{\text{CSBOOT}}$ option register (CSBTOR)



### 5.5.4.1 Chip-Select Base Address Registers

Base address registers contain the base address of the range of memory to which the chip select circuit responds. All base address registers contain the same fields but have different reset values.



#### CSBTBAR — $\overline{\text{CSBOOT}}$ Base Address Register

**0x8007 FDF8**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
BA																				RESERVED											

RESET:

IP IP IP IP IP IP IP IP IP IP IP IP IP 0

The reset value of the BA field in the CSBTBAR equals 0x00000 if the exception prefix (IP) bit in the MSR is zero (default), and 0xFFF00 if IP equals one.

#### CSBTSBBAR — $\overline{\text{CSBOOT}}$ Sub-Block Base Address Register

**0x8007 FDF0**

#### CSBAR1 — $\overline{\text{CS1}}$ Base Address Register

**0x8007 FDE0**

#### CSBAR2 — $\overline{\text{CS2}}$ Base Address Register

**0x8007 FDD8**

#### CSBAR3 — $\overline{\text{CS3}}$ Base Address Register

**0x8007 FDD0**

#### CSBAR4 — $\overline{\text{CS4}}$ Base Address Register

**0x8007 FDC8**

#### CSBAR5 — $\overline{\text{CS5}}$ Base Address Register

**0x8007 FDC0**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
BA																				RESERVED											

RESET:

0 0

**Table 5-18 Chip-Select Base Address Registers Bit Settings**

Bit(s)	Name	Description
0:19	BA	Base address. Bits 0 through 19 of the base address of the block to which the chip select responds. Register bit 0 corresponds to address bit 0; register bit 19 corresponds to address bit 19.
20:31	—	Reserved.

### 5.5.4.2 Chip-Select Option Registers

CSBTOR, the option register for  $\overline{\text{CSBOOT}}$ , has the same field definitions as the option registers for  $\overline{\text{CS}}[1:5]$  but has different reset values. The  $\overline{\text{CS0}}$  and  $\overline{\text{CS}}[6:10]$  option registers contain a subset of the fields in the CSBTOR. The  $\overline{\text{CSBOOT}}$  sub-block option register contains a different subset of the fields in the CSBTOR.

The reset values of several bits in the chip-select option registers depend on the data bus configuration word (the state of the internal data bus) at reset. The TADLY field in the  $\overline{\text{CSBOOT}}$  option register is read from the internal DATA[6:8] bits, and the PS field is determined from DATA4. In addition, the reset value of the PCON field in the option

registers for  $\overline{CS}[0:11]$  depends on the value of internal DATA0 at reset. If DATA0 = 1, the  $\overline{CS}[0:11]/\text{ADDR}[0:11]$  pins are configured as chip selects, and the PCON field at reset is 0b10 (output enable) for  $\overline{CS}_0$  and 0b00 (chip enable) for  $\overline{CS}[0:11]$ . If internal DATA0 = 0 at reset, the pins are configured as address pins, and the PCON field values for all option registers are 0b11 (non-chip-select function). Refer to [5.8.3 Configuration During Reset](#) for more information on the data bus configuration word.



### CSBTOR — $\overline{CSBOOT}$ Option Register

0x8007 FDFC

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
BSIZE				SBLK	SUPV	DSP	WP	CI	RESERVED				ACK-EN	TADLY	

RESET:

1 0 0 1 0 1 0 1 0 0 0 0 0 1 \* \*

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
TAD-LY	PS		PCON		BYTE		REGION			RESERVED		ITYPE			

RESET:

\* \* \* 0 0 0 0 0 0 0 0 0 0 \* \* \*

\*From data bus reset configuration word

### CSBTSBOR — $\overline{CSBOOT}$ Sub-Block Option Register

0x8007 FDF4

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
BSIZE				SBLK	SUPV	DSP	WP	CI	RESERVED						

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED															

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

**CSOR0 —  $\overline{\text{CS0}}$  Option Register****0x8007 FDEC**

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

RESET:

0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED			PCON		BYTE		REGION			RESERVED					

RESET:

0   0   0   \*   \*   0   0   0   0   0   0   0   0   0   0   0

\*0b10 if pins are configured as chip selects at reset, otherwise 0b11

**CSOR1 —  $\overline{\text{CS1}}$  Option Register****0x8007 FDE4****CSOR2 —  $\overline{\text{CS2}}$  Option Register****0x8007 FDDC****CSOR3 —  $\overline{\text{CS3}}$  Option Register****0x8007 FDD4****CSOR4 —  $\overline{\text{CS4}}$  Option Register****0x8007 FDCC****CSOR5 —  $\overline{\text{CS5}}$  Option Register****0x8007 FDC4**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
BSIZE				SBLK	SUPV	DSP	WP	CI	RESERVED				ACK-EN	TADLY	

RESET:

0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
TADLY	PS		PCON		BYTE		REGION			RESERVED		ITYPE			

RESET:

0   0   0   \*   \*   0   0   0   0   0   0   0   0   0   0   0

\*0b00 if pins are configured as chip selects at reset, otherwise 0b11



**CSOR6** —  $\overline{\text{CS6}}$  Option Register  
**CSOR7** —  $\overline{\text{CS7}}$  Option Register  
**CSOR8** —  $\overline{\text{CS8}}$  Option Register  
**CSOR9** —  $\overline{\text{CS9}}$  Option Register  
**CSOR10** —  $\overline{\text{CS10}}$  Option Register  
**CSOR11** —  $\overline{\text{CS11}}$  Option Register

**0x8007 FDBC**  
**0x8007 FDB4**  
**0x8007 FDAC**  
**0x8007 FDA4**  
**0x8007 FD9C**  
**0x8007 FD94**

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

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED			PCON		BYTE		REGION			RESERVED					

RESET:

0 0 0 \* \* 0 0 0 0 0 0 0 0 0 0 0

\*0b00 if pins are configured as chip selects at reset, otherwise 0b11

**Table 5-19** describes the fields in the chip-select option registers.



**Table 5-19 Chip-Select Option Register Bit Settings**

Bit(s)	Name	Description
0:3	BSIZE	<p>Block size. This field determines the size of the block associated with the base address.</p> <p>0000 = Disables corresponding region            0001 = 4 Kbytes            0010 = 8 Kbytes            0011 = 16 Kbytes            0100 = 32 Kbytes            0101 = 64 Kbytes            0110 = 128 Kbytes            0111 = 256 Kbytes            1000 = 512 Kbytes            1001 = 1 Mbyte            1010 = 2 Mbytes            1011 = 4 Mbytes            1100 = 8 Mbytes            1101 = 16 Mbytes            1110 = 32 Mbytes            1111 = 64 Mbytes</p> <p>Refer to <a href="#">5.5.5 Chip-Select Regions</a> for more information.</p>
4	SBLK	<p>Sub-block</p> <p>0 = Address space is a main block            1 = Address space specified by the BA and BSIZE fields of the corresponding base address and option registers, respectively, is a sub-block within a larger main block. Pairing of main blocks and sub-blocks is as follows:  <math>\overline{\text{CSBOOT}}</math> and <math>\overline{\text{CS1}}</math>  <math>\overline{\text{CS2}}</math> and <math>\overline{\text{CS3}}</math>  <math>\overline{\text{CS4}}</math> and <math>\overline{\text{CS5}}</math></p> <p>Refer to <a href="#">5.5.6 Multi-Level Protection</a> for more information.</p>
5	SUPV	<p>Supervisor mode</p> <p>0 = Access is permitted in supervisor or user mode            1 = Access is permitted in supervisor mode only</p> <p>Refer to <a href="#">5.5.7.1 Supervisor Space Protection</a> for more information.</p>
6	DSP	<p>Data space only</p> <p>0 = Address block may contain both instructions and data.            1 = Address block contains data only.</p> <p>Refer to <a href="#">5.5.7.2 Data Space Protection</a> for more information.</p>
7	WP	<p>Write protect</p> <p>0 = Block is available for both read and write operations            1 = Block is read only</p> <p>Refer to <a href="#">5.5.7.3 Write Protection</a> for more information.</p>
8	CI	<p>Cache inhibit</p> <p>0 = Information in this block can be cached.            1 = Information in this block should not be cached.</p> <p>Refer to <a href="#">5.5.8 Cache Inhibit Control</a> for more information.</p>
9:12	—	Reserved
13	ACKEN	<p>Acknowledge enable.</p> <p>0 = Chip-select logic will not return <math>\overline{\text{TA}}</math> and <math>\overline{\text{AACK}}</math> signals            1 = Chip-select logic will return <math>\overline{\text{TA}}</math> and <math>\overline{\text{AACK}}</math> signals</p> <p>Refer to <a href="#">5.5.9 Handshaking Control</a> for more information.</p>

**Table 5-19 Chip-Select Option Register Bit Settings (Continued)**



Bit(s)	Name	Description
14:16	TADLY	<p><math>\overline{TA}</math> delay. Indicates the latency of the device for the first <math>\overline{TA}</math> returned. Up to seven wait states are allowed.</p> <p>000 = 0 wait states  001 = 1 wait state  010 = 2 wait states  011 = 3 wait states  100 = 4 wait states  101 = 5 wait states  110 = 6 wait states  111 = 7 wait states</p> <p>Refer to <a href="#">5.5.10 Wait State Control</a> for more information.</p>
17:18	PS	<p>Port size</p> <p>00 = Reserved  01 = 16-bit port  10 = 32-bit port  11 = Reserved</p> <p>Refer to <a href="#">5.5.11 Port Size</a> for more information.</p>
19:20	PCON	<p>Pin configuration. Note that only pins <math>\overline{CSBOOT}</math> and <math>\overline{CS}[1:5]</math> can be <math>\overline{CE}</math> pins.</p> <p>00 = Chip enable (<math>\overline{CE}</math>)  01 = Write enable (<math>\overline{WE}</math>)  10 = Output enable (<math>\overline{OE}</math>)  11 = Alternate function (address bus or discrete output)</p> <p>Refer to <a href="#">5.5.12.1 Pin Configuration</a> for more information.</p>
21:22	BYTE	<p>Byte enable. This field applies to pins configured as <math>\overline{WE}</math>s only. Specifies for which of the four bytes in a word the <math>\overline{WE}</math> is asserted. If the region can always be written in 32-bit quantity, this field can be programmed to any value.</p> <p>00 = Byte enable 0  01 = Byte enable 1  10 = Byte enable 2  11 = Byte enable 3</p> <p>Refer to <a href="#">5.5.12.2 Byte Enable Control</a> for more information.</p>
23:25	REGION	<p>Memory region (only applicable when pin is configured to be a <math>\overline{WE}</math> or <math>\overline{OE}</math> pin). These bits indicate the memory region with which the pin is associated.</p> <p>000 = <math>\overline{CSBOOT}</math>  001 = <math>\overline{CS1}</math>  010 = <math>\overline{CS2}</math>  011 = <math>\overline{CS3}</math>  100 = <math>\overline{CS4}</math>  101 = <math>\overline{CS5}</math>  110 = Reserved  111 = Reserved</p> <p>Refer to <a href="#">5.5.5 Chip-Select Regions</a> for more information.</p>
26:27	—	Reserved
28:31	ITYPE	<p>Interface type. Indicates the type of memory or peripheral device being controlled. Refer to <a href="#">5.5.13 Interface Types</a> for details.</p>

### 5.5.5 Chip-Select Regions

The SIU supports an address space of four gigabytes ( $2^{32}$  bytes). This space can be divided into regions. Each region can be occupied by one or more chips, depending on the output width of each chip.

Each chip-select pin that is programmed as a chip enable defines a separate region. Only the  $\overline{\text{CSBOOT}}$  and  $\text{CS}[1:5]$  pins can serve as chip enables. All chips within a region have a common chip enable signal.



Each chip select that can be programmed as a chip enable has an associated base address register. In addition, the  $\overline{\text{CSBOOT}}$  sub-block circuit has a base address register. The base address register specifies the base address of the memory or peripheral controlled by the chip select.

The base address and block size together determine the range of addresses controlled by a chip select. Block size is the extent of the address block above the base address. Block size is specified in the BSIZE field of the chip-select option register.

The BA (base address) field in the chip-select base address register contains the high-order bits (bits 0 through 19) of the address block to which the associated chip select responds. Register bit 0 corresponds to ADDR0; register bit 19 corresponds to ADDR19. The BSIZE field determines how many of these bits are actually compared. For the smallest block size encoding (four Kbytes), bits 0 through 19 are compared with ADDR[0:19]. For larger block sizes, not all of these bits are compared. **Table 5-20** shows the block size and address lines compared for each BSIZE encoding.

**Table 5-20 Block Size Encoding**

BSIZE Field (Binary)	Block Size (Bytes)	Address Lines Compared
0000	Invalid	Chip select is not asserted until BSIZE field is assigned a non-zero value.
0001	4 Kbytes	ADDR[0:19]
0010	8 Kbytes	ADDR[0:18]
0011	16 Kbytes	ADDR[0:17]
0100	32 Kbytes	ADDR[0:16]
0101	64 Kbytes	ADDR[0:15]
0110	128 Kbytes	ADDR[0:14]
0111	256 Kbytes	ADDR[0:13]
1000	512 Kbytes	ADDR[0:12]
1001	1 Mbyte	ADDR[0:11]
1010	2 Mbytes	ADDR[0:10]
1011	4 Mbytes	ADDR[0:9]
1100	8 Mbytes	ADDR[0:8]
1101	16 Mbytes	ADDR[0:7]
1110	32 Mbytes	ADDR[0:6]
1111	64 Mbytes	ADDR[0:5]

Since the address decode logic of the chip select uses only the most significant address bits to determine an address match within its block size, the value of the base address must be a multiple of the corresponding block size.

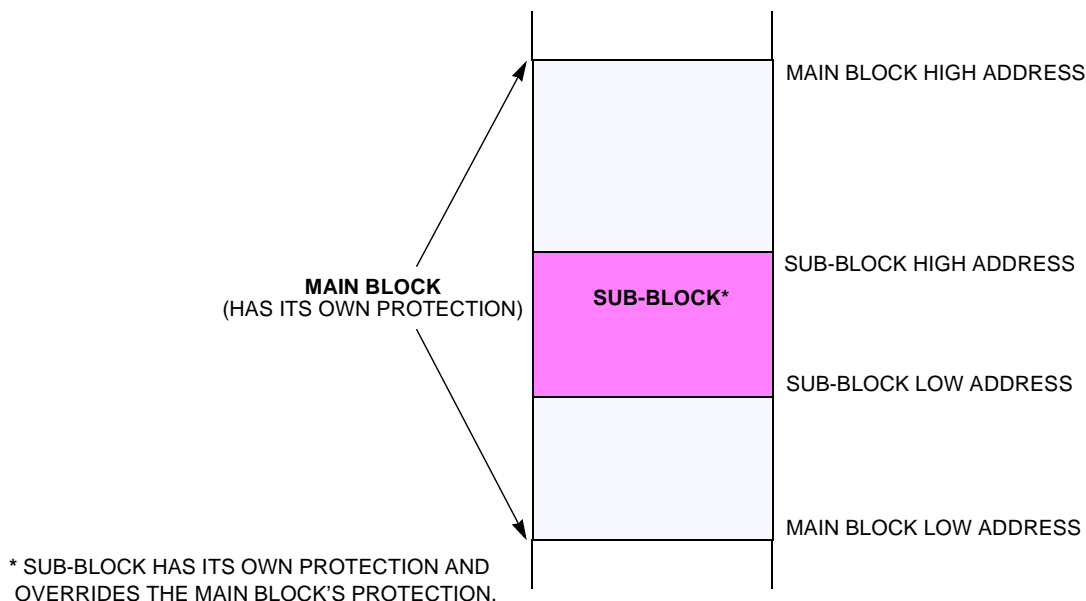


Although the base address registers can be programmed to be any address within the address map, the user must avoid programming these registers to values that overlap the addresses of internal modules. At power-on time, the address of the boot device may match that of an internal module, because a system can have on-chip EPROM for instructions. If this occurs, the internal access overrides the external access. That is, the internal access provides the boot instructions, and the chip-select unit does not run an external cycle.

The reset value of the BA field in the  $\overline{\text{CSBOOT}}$  base address register depends on the value of the exception prefix (IP) bit in the machine state register. Refer to the [RCPU Reference Manual](#) (RCPURM/AD) for a description of the machine state register.

### 5.5.6 Multi-Level Protection

The chip-select unit allows protection for an address space within another address space. [Figure 5-13](#) illustrates this concept.



**Figure 5-13 Multi-Level Protection**

[Figure 5-13](#) shows a sub-block contained within a main block. The main block and the sub-block can have different protection mechanisms programmed. For example, the user can have a separate data space and instruction space within a single chip-select region.

The block size of the sub-block should be less than the block size of the main block. The protection of the smaller block overrides the protection of the main block.



### 5.5.6.1 Main Block and Sub-Block Pairings

Multi-level protection is accomplished using a paired set of chip-select decoding circuits. The decoding pairs are specified in [Table 5-21](#).



**Table 5-21 Main Block and Sub-Block Pairings**

Main Block	Sub-Block
$\overline{\text{CSBOOT}}$	$\overline{\text{CS1}}$
$\overline{\text{CS2}}$	$\overline{\text{CS3}}$
$\overline{\text{CS4}}$	$\overline{\text{CS5}}$

If the address of an access falls within a sub-block, the protection of the sub-block overrides that of the main block. (The sub-block decoding logic overrides the decoding logic of the main block.) If all match conditions are met, the chip-select pin of the main block is asserted.

Notice that only  $\overline{\text{CSBOOT}}$  and  $\overline{\text{CS}}[1:5]$  are involved in the sub-block protection scheme. These are the chip selects with address decoding logic (i.e., they can act as chip enables).

### 5.5.6.2 Programming the Sub-Block Option Register

When the SBLK bit in CSOR1, CSOR3, or CSOR5 is set, the corresponding address block (defined by the BA and BSIZE fields) is designated a sub-block. [Table 5-21](#) indicates the main block to which the sub-block is assigned.

When the SBLK bit in one of these registers is set, the following fields in the sub-block option register must be programmed to the same values as in the option register for the corresponding main block: ITYPE, ACKEN, TADLY, and PS. If there is a discrepancy in the encoding of any of these bits in the two option registers, the chip-select unit uses the bits that are set in either register (i.e., it performs a logical OR on the associated bits in the two registers).

When the SBLK bit in CSOR1, CSOR3, or CSOR5 is set, the corresponding chip-select pin cannot act as a  $\overline{\text{CE}}$  pin, since its decoder is used for multi-level protection. The pin, however, can still be configured (by programming the PCON field) to function as an  $\overline{\text{OE}}$ ,  $\overline{\text{WE}}$ , or non-chip-select pin. If the pin is configured as a  $\overline{\text{WE}}$  or  $\overline{\text{OE}}$ , it can be assigned to any region (not just the region associated with the sub-block).

The  $\overline{\text{CSBOOT}}$ ,  $\overline{\text{CS2}}$ , and  $\overline{\text{CS4}}$  regions cannot be sub-blocks. They can only be the main blocks. Setting the SBLOCK bit in any of these registers has no effect.

### 5.5.6.3 Multi-Level Protection for $\overline{\text{CSBOOT}}$

The  $\overline{\text{CSBOOT}}$  region has a dedicated sub-block decoder in addition to its paired sub-block decoder ( $\overline{\text{CS1}}$ ). If both sub-block decoders are used, the  $\overline{\text{CS1}}$  decoder has higher priority than the dedicated sub-block decoder. That is, if an address is contained in both sub-blocks, the protections specified in the  $\overline{\text{CS1}}$  option register are used.

If an address is contained in the dedicated sub-block (and the  $\overline{\text{CSBOOT}}$  main block) but not the CS1 sub-block, the protections specified in the  $\overline{\text{CSBOOT}}$  sub-block option register are used.



The SBLK bit of the dedicated sub-block option register is cleared at power-on. The bit can be modified after reset if needed. The boot region would need to contain enough instructions to reconfigure the chip-select registers to provide multi-level protection shortly after power-on.

### 5.5.7 Access Protection

The SUPV, DSP, and WP bits in the option registers for  $\overline{\text{CSBOOT}}$ , the  $\overline{\text{CSBOOT}}$  sub-block, and  $\overline{\text{CS}}[1:5]$  control access to the address block assigned to the chip select. These bits are present in the option registers for chip selects with address decoding logic only; they are not present in the option registers for  $\overline{\text{CS}}0$  or  $\overline{\text{CS}}[6:11]$ . In addition, the bits take effect only if the chip select is programmed either as a  $\overline{\text{CE}}$  or as a sub-block.

If the chip-select unit detects a protection violation, it asserts the internal  $\overline{\text{TEA}}$  signal and does not assert the external chip enable signal. Assertion of  $\overline{\text{TEA}}$  causes the processor to enter the checkstop state, enter debug mode, or process a machine check exception. Refer to the [RCPU Reference Manual](#) (RCPURM/AD) for details.

#### 5.5.7.1 Supervisor Space Protection

The SUPV bit in the option registers for  $\overline{\text{CSBOOT}}$ , the  $\overline{\text{CSBOOT}}$  sub-block, and  $\overline{\text{CS}}[1:5]$  controls user-level access to the associated region. If the bit is set, access is permitted at the supervisor privilege level only. If the bit is cleared, both supervisor- and user-level accesses are permitted.

When an access is made to the region assigned to the chip select, the chip-select logic compares the SUPV bit with the internal AT0 signal, which indicates whether the access is at the user (AT0 = 0) or supervisor (AT0 = 1) privilege level. If the chip-select logic detects a protection violation (SUPV = 1 and AT0 = 0), it asserts the internal  $\overline{\text{TEA}}$  signal and does not assert the external chip enable signal.

This protection applies to data address space only. The chip-select logic does not check for supervisor access protection on instruction accesses.

#### 5.5.7.2 Data Space Protection

The DSP bit in the option registers for  $\overline{\text{CSBOOT}}$ , the  $\overline{\text{CSBOOT}}$  sub-block, and  $\overline{\text{CS}}[1:5]$  controls whether instruction access is allowed to the address block associated with the chip select. If DSP is set, the address block is designated as data space; no instruction access is allowed. This feature can be used to prevent the system from inadvertently executing instructions out of data space.

When an access is made to the region controlled by the chip select, the chip-select logic compares the DSP bit with the internal AT1 signal, which indicates whether the access is to instruction or data space. If the chip-select logic detects a protection vio-

lation ( $DSP = 1$  and  $AT1 = 1$ ), it asserts the internal  $\overline{TEA}$  signal and does not assert the external chip enable signal.



### 5.5.7.3 Write Protection

The WP bit in the option registers for  $\overline{CSBOOT}$ , the  $\overline{CSBOOT}$  sub-block, and  $\overline{CS}[1:5]$  controls whether the address block is write-protected. If WP is set, read accesses only are permitted. If WP is cleared, both read and write accesses are allowed. This feature permits the user to protect certain regions, such as ROM regions, from being inadvertently written.

When an access is made to the region controlled by the chip select, the chip-select logic compares the WP bit with the internal  $\overline{WR}$  signal, which indicates whether the access is a read or a write. If the chip-select logic detects a protection violation ( $WP = 1$  and  $\overline{WR} = 0$ ), it asserts the internal  $\overline{TEA}$  signal and does not assert the external chip enable signal.

### 5.5.8 Cache Inhibit Control

The CI (cache inhibit) bit in the option registers for  $\overline{CSBOOT}$ , the  $\overline{CSBOOT}$  sub-block, and  $\overline{CS}[1:5]$  controls whether the information in the address block can be cached. The chip-select logic provides the status of this bit to the cache during the data phase of an access. If CI is set, the data in the region is not cached.

### 5.5.9 Handshaking Control

The acknowledge enable (ACKEN) bit in the option registers for  $\overline{CSBOOT}$  and  $\overline{CS}[1:5]$  determines whether the chip-select logic returns address acknowledge ( $\overline{AACK}$ ) and transfer acknowledge ( $\overline{TA}$ ) signals for the region. When ACKEN is set, the chip-select logic returns these signals. (When ACKEN is set, external logic can still return these signals. If it does, it must assert them before the chip-select logic asserts the signals internally.) When ACKEN is cleared, the external device must return them.

When ACKEN is cleared, the chip-select logic still returns the  $\overline{BI}$  and  $PS[0:1]$  signals. Since the chip-select logic does not return the  $\overline{TA}$  signal, the TADLY field, indicating the number of wait states before  $\overline{TA}$  assertion, is not used.

After power-on, the  $\overline{CSBOOT}$  circuit is enabled to return  $\overline{AACK}$  and  $\overline{TA}$ . If the external boot device returns the  $\overline{TA}$  signal, however, before the chip-select logic asserts  $\overline{TA}$  internally, the external  $\overline{TA}$  assertion terminates the access.

### 5.5.10 Wait State Control

The TADLY field in the option registers for  $\overline{CSBOOT}$  and  $\overline{CS}[1:5]$  indicates the number of wait states for the chip-select logic to insert before returning  $\overline{TA}$ . If this field is encoded for zero wait states,  $\overline{TA}$  is asserted one clock cycle after  $\overline{TS}$  is asserted. An encoding of one wait state means that  $\overline{TA}$  is asserted two clock cycles after  $\overline{TS}$ , and so on. Up to seven wait states are allowed. The encodings are shown in [Table 5-22](#).



**Table 5-22 TADLY and Wait State Control**

TADLY	Wait States
0b000	0
0b001	1
0b010	2
0b011	3
0b100	4
0b101	5
0b110	6
0b111	7

Note this field is used only when the chip-select logic returns the handshaking signals (ACKEN = 1).

Note that the user does not program the number of wait states prior to  $\overline{\text{AACK}}$  assertion. The chip-select logic uses the following scheme to determine when to assert  $\overline{\text{AACK}}$ :

- If the region is an asynchronous type,  $\overline{\text{AACK}}$  is asserted at the end of access to the region.
- If the region is pipelineable and the region is not busy with a pending access,  $\overline{\text{AACK}}$  is asserted after the address is latched by the region (i.e., at the next rising clock edge).
- If the region is pipelineable and the region is busy with a pending access,  $\overline{\text{AACK}}$  is asserted for the next access to the region at the end of the pending access to the region (i.e., when  $\overline{\text{TA}}$  is asserted for that access).

#### 5.5.11 Port Size

The PS field indicates the port size of the region. The chip-select logic always returns PS[0:1] for regions under its control. Port size encoding is shown in [Table 5-23](#). The 0b00 and 0b11 encodings are reserved; if one of these encodings is used, the port size defaults to 32 bits.

**Table 5-23 Port Size**

PS Field	Port Size
0b00	Reserved
0b01	16 bits
0b10	32 bits
0b11	Reserved

## 5.5.12 Chip-Select Pin Control

The PCON, BYTE, and REGION fields of each chip-select option register control how the associated pin is used. The PCON field determines pin function ( $\overline{CE}$ ,  $\overline{OE}$ ,  $\overline{WE}$ , or alternate function). The BYTE field determines which byte enable a  $\overline{WE}$  pin corresponds to. The REGION field assigns a  $\overline{WE}$  or  $\overline{OE}$  pin to one of six chip-select regions.



### 5.5.12.1 Pin Configuration

The PCON (pin configuration) field in the chip-select option register configures the associated pin to be a  $\overline{CE}$ ,  $\overline{WE}$ ,  $\overline{OE}$ , or non-chip-select function pin. The encodings are shown in [Table 5-24](#).

**Table 5-24 Pin Configuration Encodings**

PCON	Pin Assignment
0b00	Chip enable ( $\overline{CE}$ )
0b01	Write enable ( $\overline{WE}$ )
0b10	Output enable ( $\overline{OE}$ )
0b11	Address pin or discrete output

Note that only the  $\overline{CSBOOT}$  and  $\overline{CS}[1:5]$  pins can be  $\overline{CE}$  pins. If the pin is a  $\overline{CE}$  pin, the REGION field does not affect it, since each  $\overline{CE}$  pin has its own base address register and decoding logic.

The  $\overline{CS0}$  and  $\overline{CS}[6:11]$  pins cannot be  $\overline{CE}$  pins. If one of these pins is configured as a chip enable, the pin is never asserted.

A PCON encoding of 0b11 assigns the pin to its alternate function. In this case, the value in the port A/B pin assignment register (PABPAR) determines whether the pin operates as an address pin or discrete output pin. Refer to [5.9.3 Ports A and B](#) for more information.

### 5.5.12.2 Byte Enable Control

The BYTE field is applicable only for pins configured as  $\overline{WE}$  pins. This field is used to determine to which of the four E-bus byte enables the pin corresponds. That is, the  $\overline{WE}$  pin will be asserted only when the corresponding E-bus byte enable is asserted. The encoding is shown in [Table 5-25](#). If the region can always be written in 32-bit quantity, this field can be programmed to any value.



**Table 5-25 BYTE Field Encodings**

BYTE	Byte Enabled
0b00	Byte enable 0
0b01	Byte enable 1
0b10	Byte enable 2
0b11	Byte enable 3

If the pin is configured as an  $\overline{OE}$ , this field is not used. (It is assumed the  $\overline{OE}$  pin enables the outputs of all four bytes of the region onto the 32-bit E-bus.) Thus, typically a writable region would have multiple  $\overline{WE}$ s, one  $\overline{OE}$ , and one  $\overline{CE}$ .

### 5.5.12.3 Region Control

The REGION field indicates which memory region the pin is assigned to. This field is used only when the pin is configured to be a  $\overline{WE}$  or  $\overline{OE}$  pin. For example, a PCON encoding of 0b10 and a REGION encoding of 0b001 configures the pin as an  $\overline{OE}$  of the memory region defined by  $\overline{CS1}$ .

REGION field encodings are shown in [Table 5-26](#).

**Table 5-26 REGION Field Encodings**

REGION	Memory Region Defined by
0b000	$\overline{CSBOOT}$
0b001	$\overline{CS1}$
0b010	$\overline{CS2}$
0b011	$\overline{CS3}$
0b100	$\overline{CS4}$
0b101	$\overline{CS5}$
0b110	Reserved
0b111	Reserved

### 5.5.13 Interface Types

The chip-select module supports a wide variety of devices. The interface type (ITYPE) field in the option registers for  $\overline{CSBOOT}$  and  $\overline{CS[1:5]}$  identifies the characteristics of the device interface. These characteristics include whether the external device interface:

- Is synchronous or asynchronous
- Supports pipelined accesses
- Can hold off its internal data

- Has a synchronous  $\overline{OE}$ , an asynchronous  $\overline{OE}$ , or no  $\overline{OE}$
- Is burstable or non-burstable
- Uses the  $\overline{LAST}$  or  $\overline{BDIP}$  protocol for ending a burst transmission



The following paragraphs define these concepts.

A burstable device can accept one address and drive out multiple data beats. A burstable device must be synchronous. (Note that devices with fast static column access are not considered burstable. This class of devices is considered asynchronous.)

Two accesses are overlapped if they are aligned such that the address of the second access is on the external bus at the same time as the data of the first access.

Two accesses are pipelined if they are aligned such that the address of the second access is on the external bus before the data of the first access.

A device is pipelineable if it can latch the address presented to it and does not require the address to be valid on its address pins for the duration of the access to the device. The pipelineable device should latch the address at the rising edge of the clock when its  $\overline{CE}$  is asserted. Note that only synchronous devices are treated as pipelineable by the chip-select logic.

$\overline{BDIP}$  and  $\overline{LAST}$  are the early termination control signals for burst cycles. A memory device with a type 1 burst interface may have a  $\overline{BDIP}$  signal as one of its inputs. A memory device with a type 2 burst interface has a  $\overline{LAST}$  signal as one of its inputs. Refer to [5.5.16.6 Synchronous Burst Interface](#) for a description of these interface types.

A device may or may not have the ability to hold off its data output until the data bus is available to the device. To be able to hold off its data the device needs an  $\overline{OE}$  control input, and if the device is burstable it also needs the ability to suspend its internal state machine from advancing to the next data beat until the data bus has been granted to it. An example of this is a memory device with burst address advance control such as  $\overline{BDIP}$  to control the incrementing of its internal address counter.

### 5.5.13.1 Interface Type Descriptions

**Table 5-27** lists the characteristics of each interface type. Note that if software programs the ITYPE field to one of the reserved values, the chip-select signal will never be asserted.



**Table 5-27 Interface Types**

ITYPE (Binary)	Interface Type
0000	Generic asynchronous region with output buffer turn-off time of less than or equal to one clock period (see <a href="#">5.5.13.2 Turn-Off Times for Different Interface Types</a> ). A device of this type cannot be pipelined. Refer to <a href="#">Figure 5-17</a> and <a href="#">Figure 5-18</a> .
0001	Generic asynchronous region with output buffer turn-off time of two clock periods (see <a href="#">5.5.13.2 Turn-Off Times for Different Interface Types</a> ). A device of this type cannot be pipelined. The chip-select logic inserts a dead clock between two subsequent accesses to the same region of this type in order to satisfy the high time required by the $\overline{CE}$ and $\overline{WE}$ of some memory types.
0010	Synchronous region (no burst) with asynchronous $\overline{OE}$ . Refer to <a href="#">Figure 5-19</a> and <a href="#">Figure 5-20</a> . A device with this type of interface is pipelineable, can function as an asynchronous device, and has the ability to hold off its internal data on a read access until $\overline{OE}$ is asserted. Note that with this interface type, if the MCU receives $\overline{TA}$ before asserting $\overline{OE}$ , $\overline{OE}$ may still be asserted and may remain asserted.
0011	Synchronous region (no burst) with synchronous $\overline{OE}$ . Refer to <a href="#">Figure 5-21</a> . A device with this type of interface is pipelineable, can function as an asynchronous device, and has the ability to hold off its internal data on a read access until $\overline{OE}$ is asserted. The chip-select logic asserts $\overline{OE}$ for one clock cycle on accesses to devices with this interface type. A device with synchronous $\overline{OE}$ must be programmed for one or more wait states. If the region is programmed for zero wait states with synchronous $\overline{OE}$ , the chip-select logic still generates the $\overline{OE}$ as if the region were programmed for one wait state.
0100	Reserved.
0101	Region with fixed burst access capability (burst type 1) and asynchronous $\overline{OE}$ . Refer to <a href="#">Figure 5-23</a> and <a href="#">Figure 5-24</a> . A device of this type is pipelineable and can hold off its internal data until $\overline{OE}$ is asserted. The interface keeps the first data beat valid until the $\overline{BDIP}$ signal indicates that it should send out the next data. This interface type can function as an asynchronous interface. That is, a device with this ITYPE can be assigned to the $\overline{CSBOOT}$ region, which comes out of reset configured as an asynchronous region with seven wait states. In this case, the MCU doesn't latch the data to be read until the assigned number of wait states have elapsed and $\overline{OE}$ is asserted.
0110	Reserved.
0111	Region with fixed burst access capability (burst type 1) and synchronous $\overline{OE}$ . Refer to <a href="#">Figure 5-23</a> (but with a synchronous, not asynchronous, $\overline{OE}$ ) and to <a href="#">Figure 5-24</a> . Devices with this type of interface are pipelineable and can hold off internal data until $\overline{OE}$ is asserted. The interface keeps the first data beat valid until the $\overline{BDIP}$ signal indicates that it should send out the next data. This interface type can function as an asynchronous interface. That is, a device with this ITYPE can be assigned to the $\overline{CSBOOT}$ region, which comes out of reset configured as an asynchronous region with seven wait states. In this case, the MCU doesn't latch the data to be read until the assigned number of wait states have elapsed and $\overline{OE}$ is asserted.



**Table 5-27 Interface Types (Continued)**



ITYPE (Binary)	Interface Type
1000	<p>Region with fixed burst access capability (burst type 2). Refer to <a href="#">Figure 5-25</a>. This interface type uses the <math>\overline{\text{LAST}}</math> timing protocol. Typically, this ITYPE is used for burst accesses to DRAM.</p> <p>This interface type may have an <math>\overline{\text{OE}}</math> and may have a wait state counter, but the chip-select logic does not expect the device to have either and will never assert the <math>\overline{\text{OE}}</math> signal. (<math>\overline{\text{OE}}</math> can be provided by external logic if required, or a different ITYPE can be selected.) The device will drive out the data after the number of wait states it requires. The interface keeps the first data beat valid for only one clock.</p> <p>Any access to a device with this type of interface must be made using chip selects, and the ACKEN bit in the option register for the chip select must be set. Because this type cannot hold off its internal data until the data bus is available, an access to a region of this type cannot be pipelined with a previous access to the same or a different region. (That is, the address of an access to this region cannot appear on the external bus before the data for the previous access.) The address for the second access can overlap the data for the first access, however. In addition, if an access to this region is followed by an access to a pipelineable region, the second access is pipelined.</p> <p>This interface type can function as an asynchronous interface. That is, a device with this ITYPE can be assigned to the <math>\overline{\text{CSBOOT}}</math> region, which comes out of reset configured as an asynchronous region with seven wait states. In this case, the MCU doesn't latch the data to be read until the assigned number of wait states have elapsed and <math>\overline{\text{OE}}</math> is asserted.</p>
1001	<p>Synchronous region (no burst) with synchronous <math>\overline{\text{OE}}</math>, as with ITYPE 3, but with early overlapping of accesses to the region. Refer to <a href="#">Figure 5-21</a>. This type of interface must be able to pipeline another access to it one clock cycle before it drives valid data out on a read or receives data on a write for the previous access.</p>
1010–1111	Reserved.

### 5.5.13.2 Turn-Off Times for Different Interface Types

The turn-off time for asynchronous devices is equal to the time for  $\overline{\text{OE}}$  to negate plus the time for device's outputs to go to a high-impedance state. For devices with an ITYPE of zero, turn-off time is less than or equal to one clock cycle. For devices with an ITYPE of one, turn-off time is two clock cycles.

The turn-off time of asynchronous devices must be taken into account in systems that pipeline accesses to devices controlled by chip selects with accesses to devices that are not under chip select control. Otherwise, external bus contention can result.

The turn off time for synchronous devices is equal to the time from the rising edge of the device's clock to the time the device's outputs are in a high-impedance state. This turn off time must be less than or equal to one clock period.

### 5.5.13.3 Interface Type and $\overline{\text{BI}}$ Generation

During a burst access to a region under chip-select control that does not support burst accesses, the chip-select unit asserts the  $\overline{\text{BI}}$  signal internally. Only regions with an ITYPE of five, seven, and eight support burst accesses.

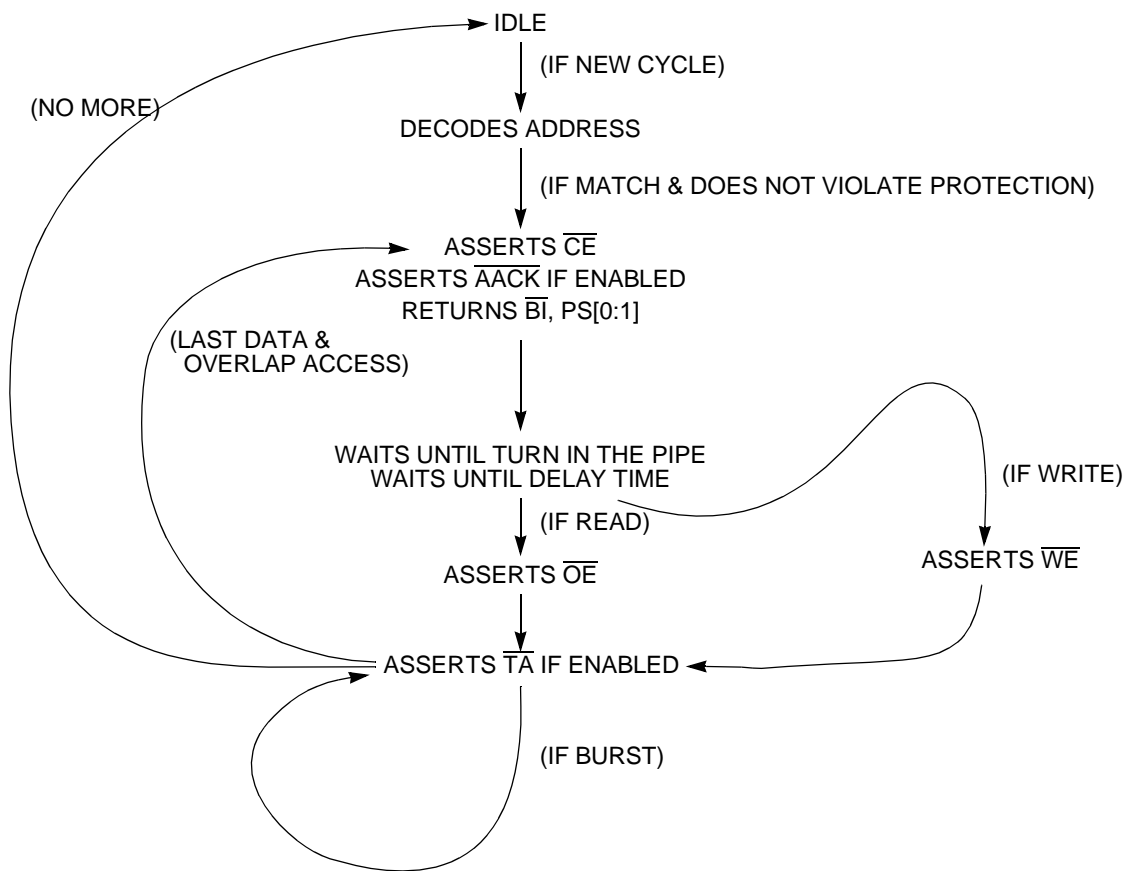
## CAUTION

It is recommended that the  $\overline{\text{BI}}$  pin not be asserted during accesses to memory regions controlled by chip selects; instead, the chip-select unit will generate the  $\overline{\text{BI}}$  signal internally when appropriate.



### 5.5.14 Chip-Select Operation Flowchart

Figure 5-14 illustrates the operation of the chip-select logic for external accesses.



MPC500 CS FLOW

Figure 5-14 Chip-Select Operation Flowchart

### 5.5.15 Pipe Tracking

The chip-select module supports pipelined accesses to external devices. Up to two cycles can be pending in the chip-select module.

The chip-select unit supports pipelined reads for certain types of interfaces. Pipelined writes are not supported. Table 5-28 summarizes the chip-select pipelining of read and write accesses.



**Table 5-28 Pipelined Reads and Writes**

First Access	Second Access	Pipelining Supported
Read	Read	Yes
Write	Read	Yes
Read	Write	No
Write	Write	No

The following subsections explain which types of interfaces permit pipelining of read accesses. Pipelining of consecutive accesses to the same region is discussed first, followed by pipelining of consecutive accesses to different regions.

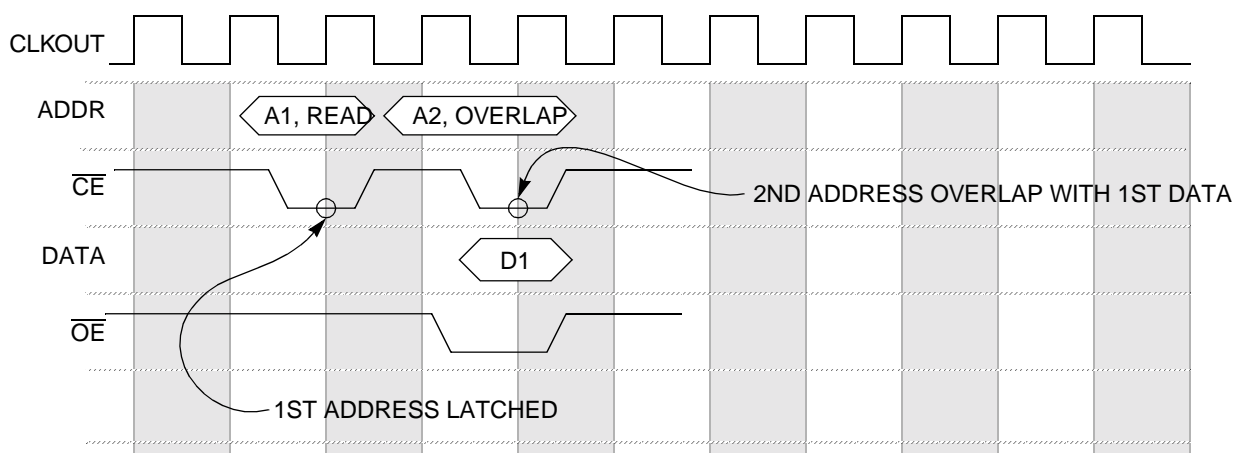
#### 5.5.15.1 Pipelined Accesses to the Same Region

The chip-select unit overlaps consecutive accesses to the same region, provided the following conditions are met:

- The second access is a read
- The region is pipelineable (as determined by its ITYPE)
- The  $\overline{TA}$  signal is generated by the chip-select logic ( $ACKEN = 1$ ).

When these conditions are met, the address (and  $\overline{CE}$  assertion) for the second access can overlap the data phase of the first access.

**Figure 5-15** illustrates the concept of overlapped accesses to the same region. (Note that the diagram cannot be assumed to be accurate in timing.)



**Figure 5-15 Overlapped Accesses to the Same Region**

## NOTE

If the region is programmed to return its own handshaking signals ( $\overline{\text{ACKEN}} = 0$ ), the chip-select logic does not know whether the device has an address latch (hence, whether the device is programmable). The chip-select control logic takes this into account and asserts the  $\overline{\text{CE}}$  of the second access only after  $\overline{\text{AACK}}$  has been asserted for the first access.

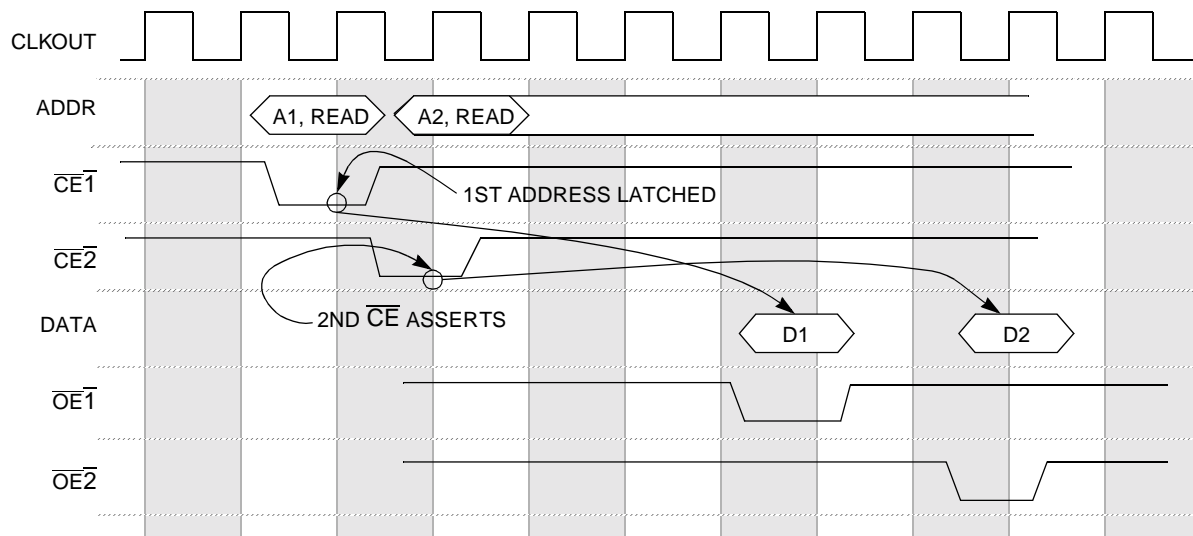


### 5.5.15.2 Pipelined Accesses to Different Regions

The chip-select unit supports pipelined accesses to different memory regions, depending on the properties of the two regions. The chip-select module tracks the incoming cycles and uses the information in the option registers to control the assertion of the  $\overline{\text{CE}}$ ,  $\overline{\text{WE}}$ , and  $\overline{\text{OE}}$  signals.

For the chip-select module to pipeline accesses to two different regions, the first region must be pipelineable; otherwise, the chip-select unit waits for the first access to complete ( $\overline{\text{TA}}$  asserted) before beginning the second access.

**Figure 5-16** uses two synchronous devices ( $\text{ITYPE} = 2$ ) to illustrate this pipelining case. In this example, the first access is to a four-wait-state region, and the second access is to a region with zero wait states. (For a second region with wait states, the pipelining is similar except the data of the second region takes more time to be available on the bus if the second region cannot hold off its internal data.) The example is intended to show when the  $\overline{\text{CE}}$  (or address phase) and the data phase of the second access can be given to the region. It assumes the device(s) in the first region is pipelineable, and both accesses are initiated by the same bus master.



**Figure 5-16 Pipelined Accesses to Two Different Regions**



1. If both regions are under chip-select control, the delays of both regions are known to the chip-select logic, and the interface type of the first region supports pipelining, then the second access (if a read) can be pipelined with the first.
2. For any two consecutive accesses, if the latency of either region is not known to the chip-select logic, the two accesses are pipelined only if the second access is a read access to a region with an interface type that can hold off the data until the data bus is available (See [Table 5-27](#)).

For example, suppose the first access is to a region that supplies its own  $\overline{TA}$  signal, the second access is to another region with  $ITYPE = 8$ , and  $\overline{TA}$  is returned by the chip-select logic for the second access. In this case, the chip-select logic must hold off the second access until the first access is completed because the second region may not be able to hold off its data without an  $\overline{OE}$ . On the other hand, suppose the first access is to a region that supplies its own  $\overline{TA}$  signal, the second access is to another region with  $ITYPE = 3$  (synchronous  $\overline{OE}$ ) and  $\overline{TA}$  is returned by the chip-select logic for the second access. In this case, the second region can hold off its data until its  $\overline{OE}$  is asserted. The chip-select module can pipeline the second access (if a read) with the first access after it has received the  $\overline{AACK}$  signal for the first access. The chip-select logic asserts the  $\overline{CE}$  of the second access while the data phase of the first access is still in progress, if the second access is issued before the first access is completed.

3. If the first access is to a region that is not under chip-select control (external glue logic generates all control and handshake signals for the region, as for a DRAM controller, for example), and the second access is to a region that is under chip-select control, the chip-select module does *not* pipeline the second access with the first.
4. If the first access is to a region under chip-select control and the second access is a read access to a region that is not under chip-select control, the external glue logic designer must decide whether to pipeline the second access with the first. The decision depends on system requirements and on the interface type of the region that is not under chip-select control.
5. If the first access is a burst read access to a burstable region and the second is a read access to another region, the chip-select module pipelines the second read if the second access is to a region with an interface type that is pipelineable and can hold off its data. If  $ITYPE = 8$  for the second region, the chip-select module does not pipeline the second access with the first.
6. If the first access is to a synchronous region, and the second access is to an asynchronous region, the chip-select module does *not* pipeline the accesses.
7. If the first access is to an asynchronous region, the chip-select module does *not* pipeline the second access with the first, since both the external address and data bus must be available for the first access until it is completed. If the first region requires an extra clock to turn off its buffer, the chip-select logic allows an extra clock for the region.

## 5.5.16 Chip-Select Timing Diagrams

The diagrams in this section show the different device interfaces that the chip-select module supports. Where applicable, the diagrams indicate how the various signals (address, data, and chip-select signals) are correlated.



### CAUTION

The user must not assume that  $\overline{CE}$  is always asserted simultaneously with  $\overline{TS}$ . Depending on the state of the pipeline (which depends on the interface types of the devices being accessed), the chip-select unit may delay asserting  $\overline{CE}$  until one or more clock cycles after  $\overline{TS}$  is asserted.

### 5.5.16.1 Asynchronous Interface

An external device with an asynchronous interface requires the address and the chip select signals ( $\overline{CE}$ ,  $\overline{OE}$ , and  $\overline{WE}$ ) to be valid until the end of the access. The next access to the same device must wait for the previous access to complete. No overlap of accesses is allowed. [Figure 5-17](#) and [Figure 5-18](#) illustrate the asynchronous interface for read and write accesses. For the asynchronous write, the external memory latches the data when  $\overline{WE}$  is asserted.

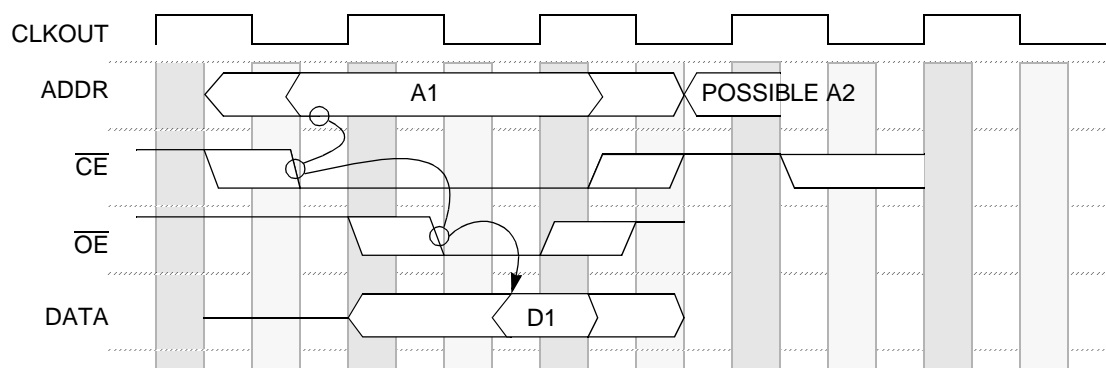
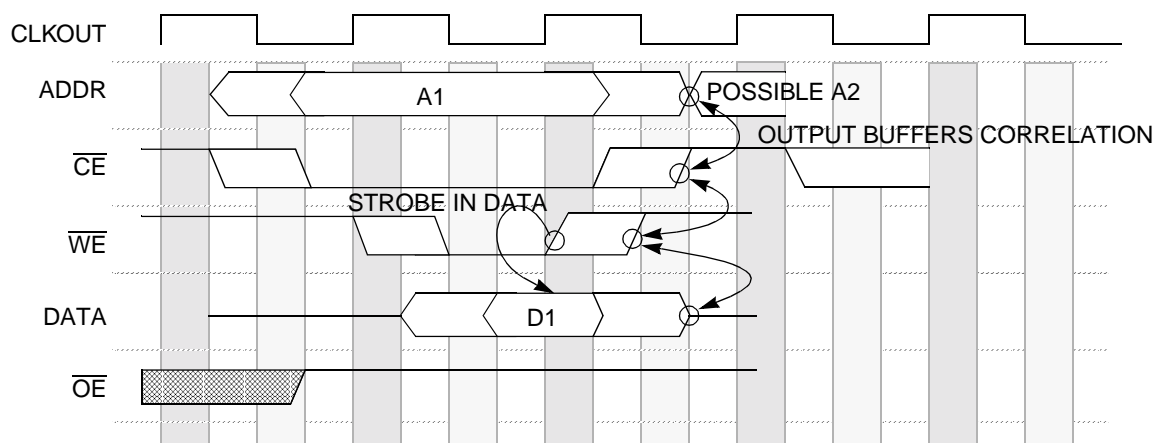


Figure 5-17 Asynchronous Read (Zero Wait States)



**Figure 5-18 Asynchronous Write (Zero Wait States)**

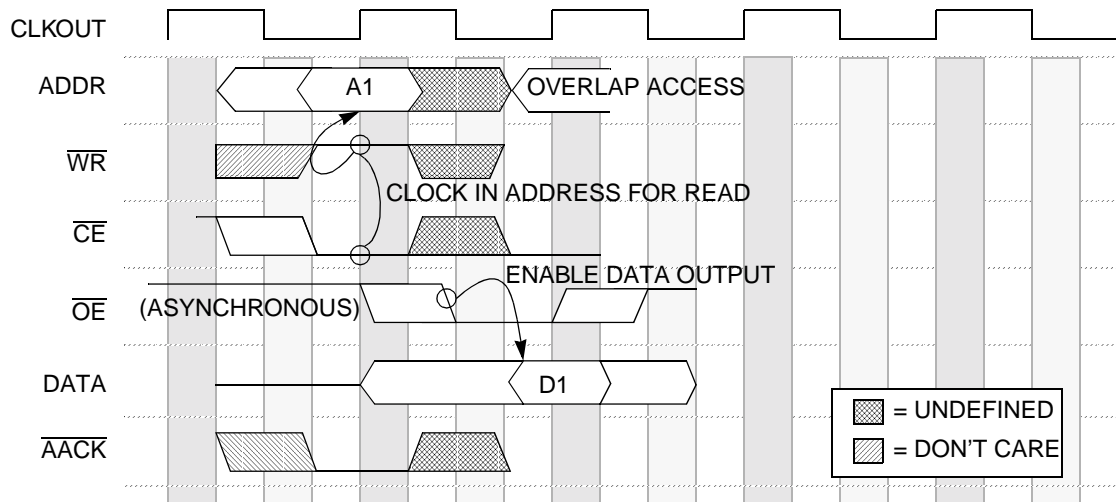
### 5.5.16.2 Asynchronous Interface with Latch Enable

Devices with an address latch enable signal, such as the Motorola MCM62995A memory chip, also support unlatched asynchronous read and write interfaces as shown in [Figure 5-17](#) and [Figure 5-18](#). The chip-select module supports this type of device in the unlatched asynchronous mode only.

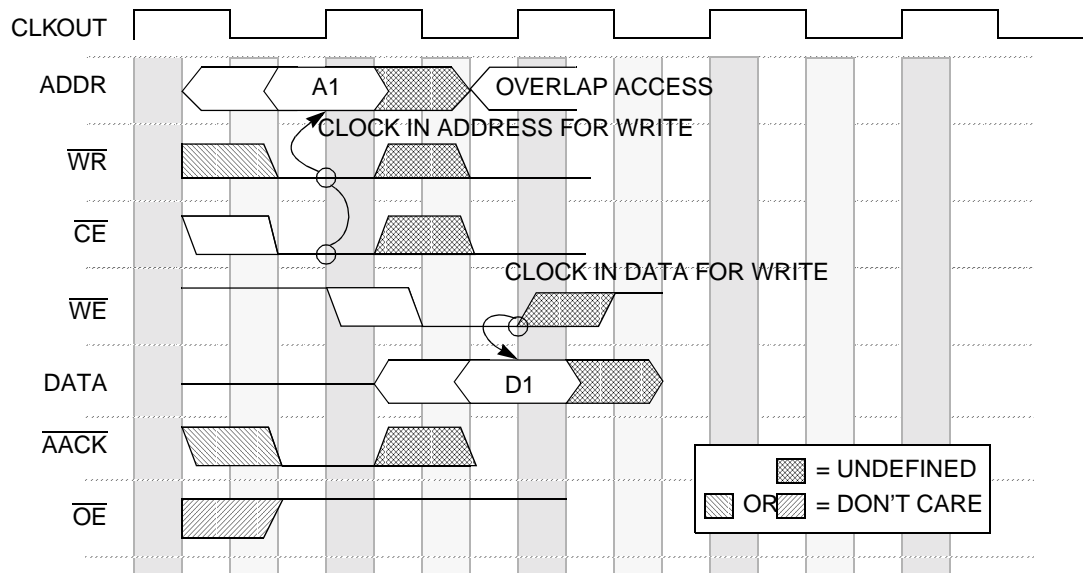
### 5.5.16.3 Synchronous Interface with Asynchronous $\overline{OE}$

Devices with  $ITYPE = 2$  have a synchronous interface with an asynchronous output enable. Devices of this type clock the address and the data on the rising edge of CLK-OUT. On a read access, these devices drive the data out as soon as the  $\overline{OE}$  is asserted. In addition, the interface has the ability to latch the address so the next access to the same device can be overlapped with the previous access.

[Figure 5-19](#) and [Figure 5-20](#) illustrate reads and writes for devices with this type of interface.



**Figure 5-19 Synchronous Read with Asynchronous  $\overline{OE}$  (Zero Wait States)**



**Figure 5-20 Synchronous Write (Zero Wait States)**

#### 5.5.16.4 Synchronous Interface with Early Synchronous $\overline{OE}$

Devices with  $ITYPE = 3$  have a synchronous interface with a synchronous output enable.  $\overline{OE}$  is asserted, at the earliest, one clock cycle after  $\overline{CE}$ . The synchronous  $\overline{OE}$  should be sampled by the external device using the rising edge of its clock signal.

For read accesses, the early  $\overline{OE}$  signal allows the responding device to prepare for the next data cycle. If  $\overline{OE}$  is asserted, the device can prepare to drive the next data or refill its internal data queue. If  $\overline{OE}$  is not asserted, the device can place the data lines in a high-impedance state for fast relinquishing of the data bus.



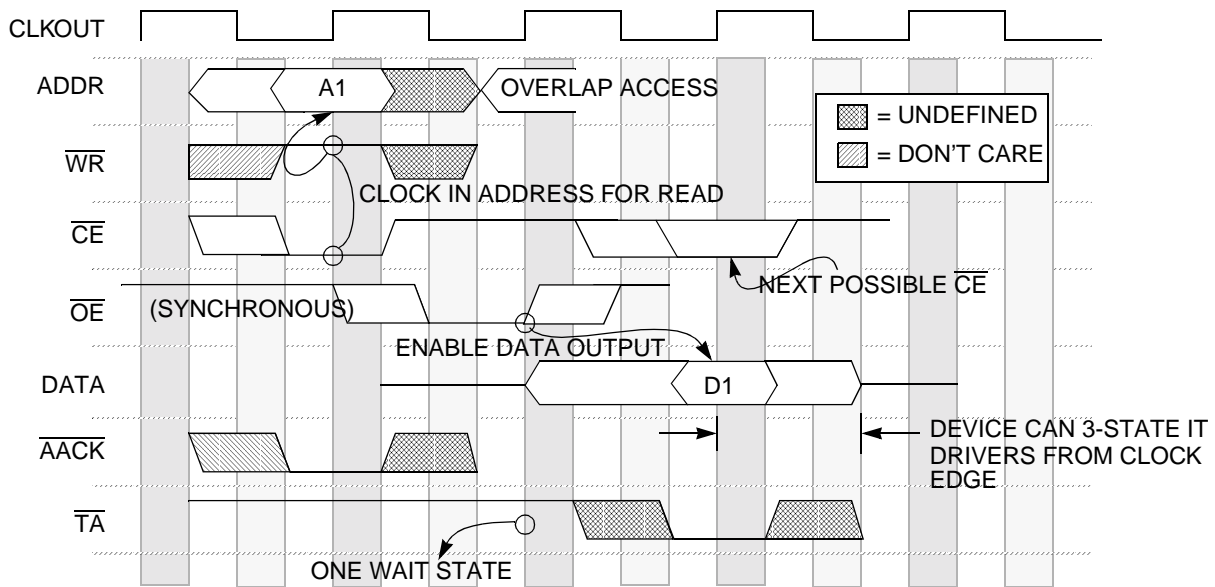


Figure 5-21 Synchronous Read with Early  $\overline{OE}$  (One Wait State)

#### 5.5.16.5 Synchronous Interface with Synchronous $\overline{OE}$ , Early Overlap

Devices with ITYPE = 9 are synchronous with a synchronous output enable. They are different from devices with ITYPE = 3 in that they support early overlapping of accesses. That is, the region is capable of accepting a second address one clock cycle before the data phase of the first access terminates. Notice in [Figure 5-22](#) that  $\overline{CE}$  is asserted one clock cycle earlier than in the previous example ([Figure 5-21](#)).

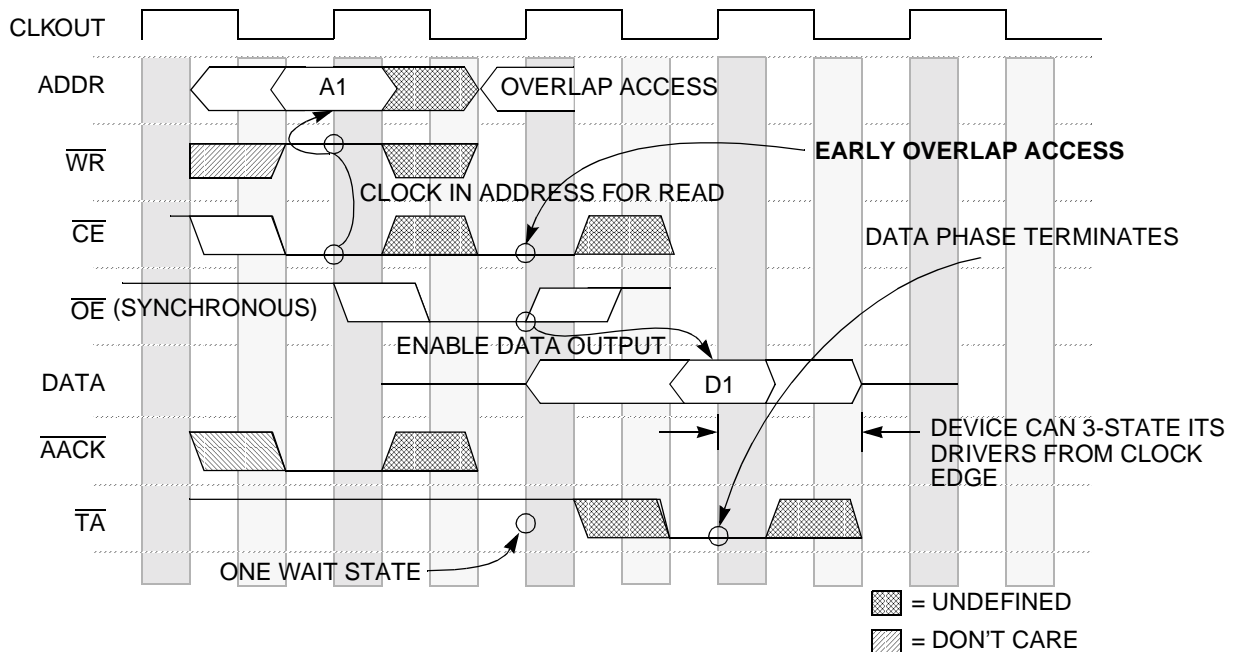


Figure 5-22 Synchronous Read with Early Overlap (One Wait State)

### 5.5.16.6 Synchronous Burst Interface



The chip-select module supports two types of burst interfaces. The type 1 burst interface uses the output enable and the write enable to control the data being driven out or received. The type 1 burst interface also requires a  $\overline{\text{BDIP}}$  signal to control when the region should output the next beat of the burst.

For the read case, the type 2 burst interface does not require an output enable signal. Instead, it uses a  $\overline{\text{LAST}}$  signal. When this signal is asserted at the rising edge of the clock, the type 2 burst device places its output buffers in a high-impedance state following the clock edge. The  $\overline{\text{CE}}$  of the type 2 burst must be valid for the duration of the device's access latency or wait states. This type of device also requires a signal with timing similar to that of the  $\overline{\text{TS}}$  signal. The interface may or may not contain an  $\overline{\text{OE}}$  signal.

Any access to a device with type 2 burst interface must be made using chip selects, and the ACKEN bit in the option register for the chip select must be set.

#### NOTE

The  $\overline{\text{LAST}}$  and  $\overline{\text{BDIP}}$  signals share the same pin. The LST bit in the SIU module configuration register (SIUMCR) specifies whether the pin uses timing for the  $\overline{\text{LAST}}$  signal (LST = 1) or the  $\overline{\text{BDIP}}$  signal (LST = 0).

Type 1 and type 2 burst interfaces both have address latches, so the address of the next access to the device can be overlapped with the previous access. That is, the address of an access does not need to be valid after the address has been latched at the rising edge of the clock.

For type 1 burst interfaces with an asynchronous  $\overline{\text{OE}}$ , the ITYPE field in the appropriate chip-select option register should be programmed to 0b0101. For type 1 burst interfaces with a synchronous  $\overline{\text{OE}}$ , this field should be programmed to 0b0111. For type 2 burst interfaces, ITYPE should be programmed to 0b1000.

**Figure 5-23** and **Figure 5-24** show a read and write access, respectively, to a type 1 burst interface.

#### NOTE

In **Figure 5-23** that the  $\overline{\text{OE}}$  is asynchronous (ITYPE = 5).

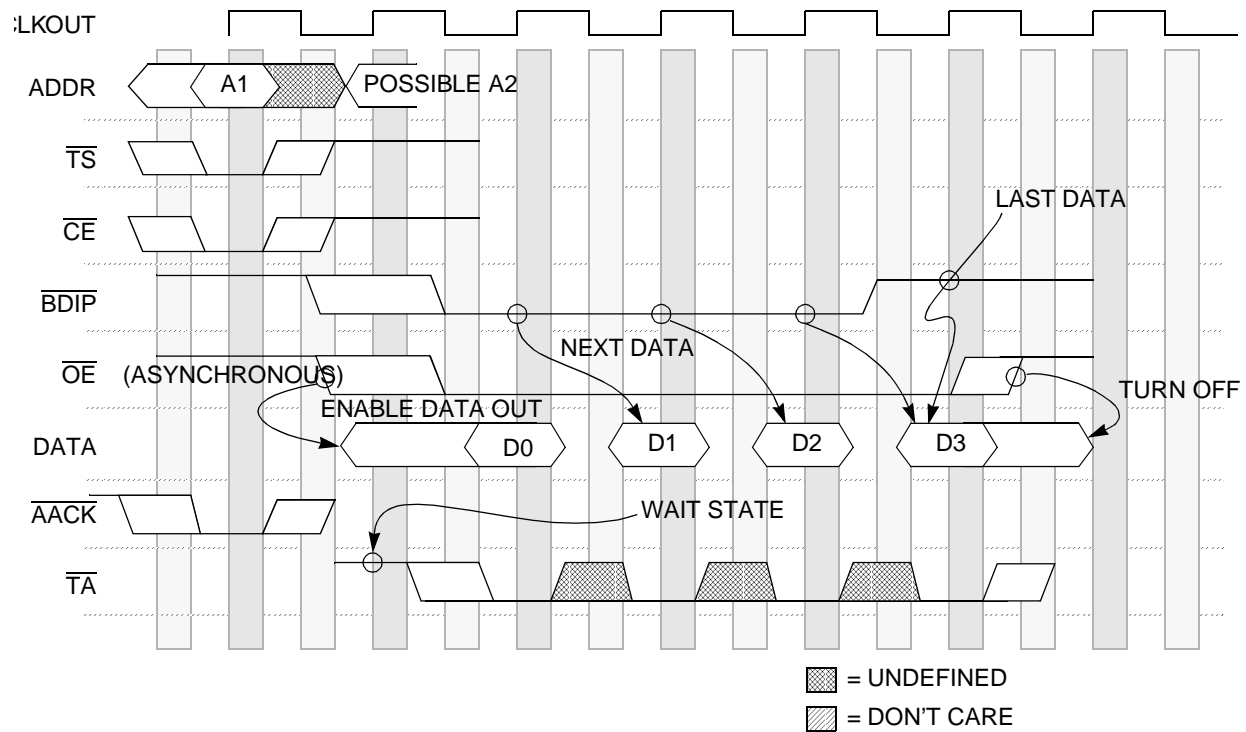


Figure 5-23 Type 1 Synchronous Burst Read Interface

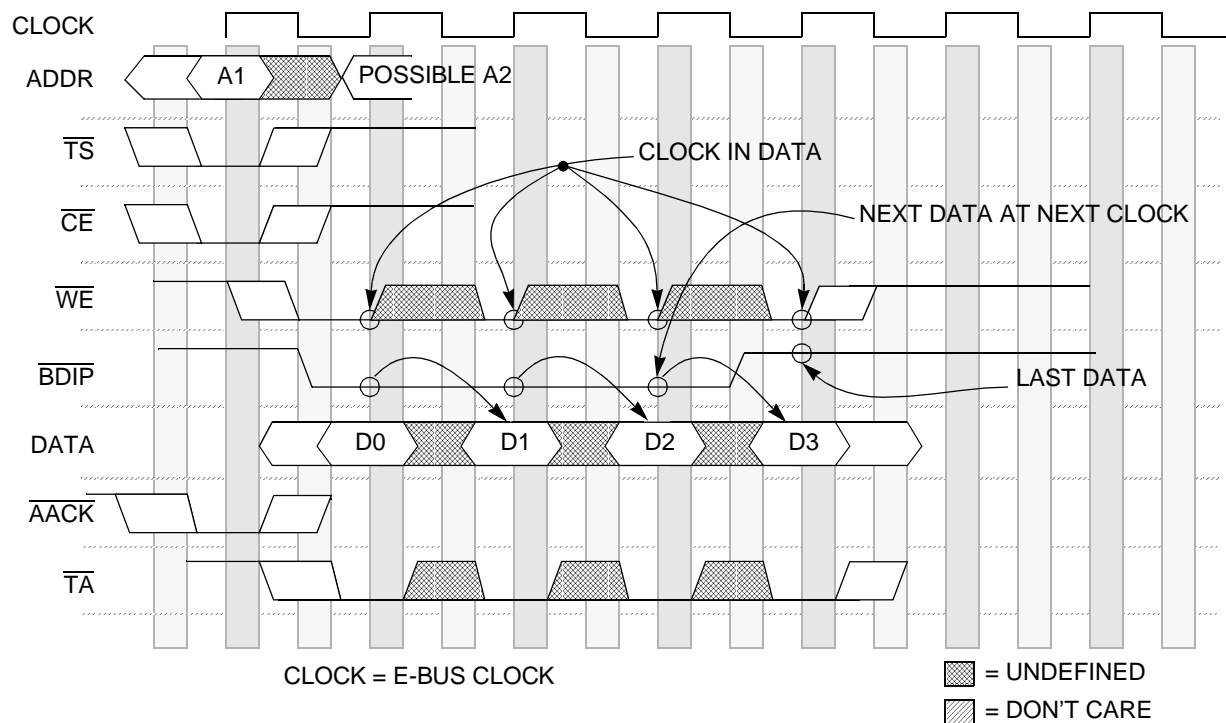
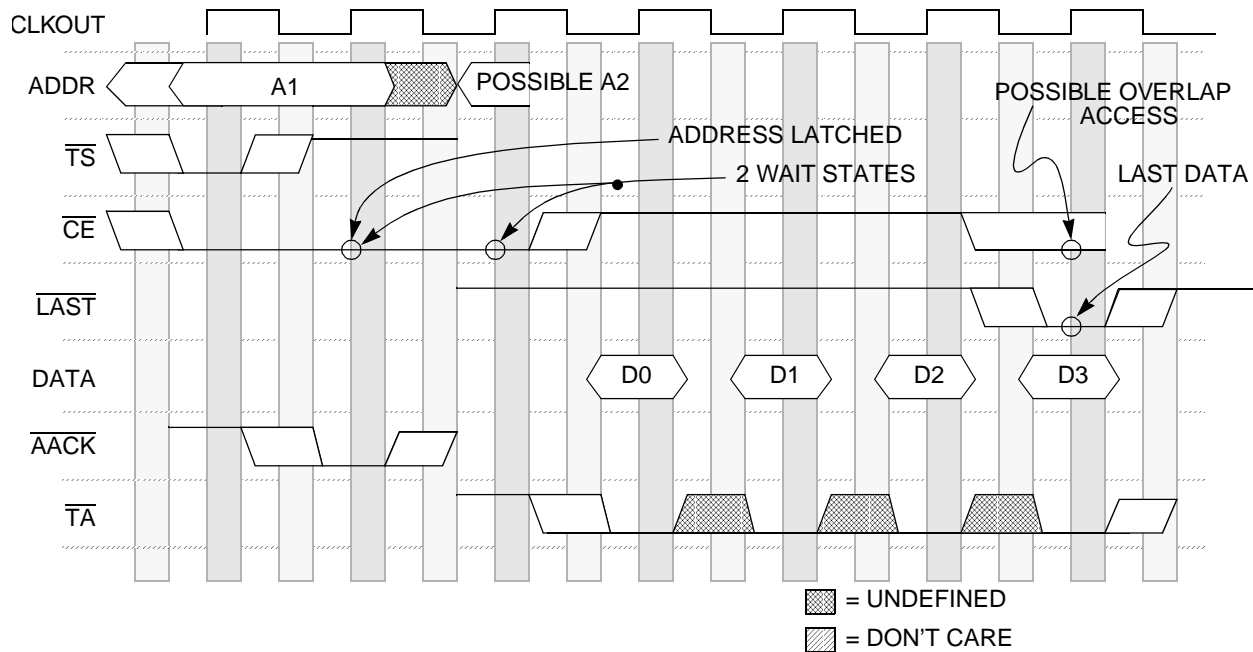


Figure 5-24 Type 1 Synchronous Burst Write Interface

**Figure 5-25** shows a read access to a type 2 burst interface (ITYPE = 8). Note that an output enable signal is not required for this type of interface. Instead, the interface uses the  $\overline{\text{LAST}}$  signal.



**Figure 5-25 Type 2 Synchronous Burst Read Interface**

### 5.5.17 Burst Handling

The chip-select module supports burst accesses, with four data beats per burst. The following paragraphs describe how the chip-select module handles some of the more complex cases.

- For a single-word access to a burstable region, the chip-select module asserts  $\overline{\text{OE}}$  (on a read) or  $\overline{\text{WE}}$  (on a write) for only one word. The burstable region may require an early termination signal such as  $\overline{\text{LAST}}$ . The EBI is expected to provide the early termination indication to the region.
- For fixed burst access to a burstable region, since all burstable types supported by the chip-select module allow fixed burst accesses, the chip-select module keeps the  $\overline{\text{OE}}$  or  $\overline{\text{WE}}$  asserted for the length of four words unless the cycle is terminated early.
- For a burst access to a non-burstable region, the chip-select module asserts the burst inhibit indication to the EBI and treats the access as a single-word access.
- For a fixed-burst access to a burstable small port (16-bit) device, the chip-select module keeps the  $\overline{\text{OE}}$  or  $\overline{\text{WE}}$  valid until the EBI terminates the burst. Depending on the starting address of the burst, the EBI breaks the access into two or more cycles and increments the address appropriately. The small port device is expected to wrap as specified in [5.4 External Bus Interface](#).
- For a single-word access to a device with a small port, the chip-select module always performs a single access to the small port device and indicates to the EBI

that the device has a 16-bit port. If more data is needed, the EBI requests the chip selects to perform another access to the device to complete the transfer.



### 5.5.18 Chip-Select Reset Operation

The data bus configuration word specifies how the MCU is configured at reset. [Table 5-29](#) summarizes the data bus configuration bits that affect chip selects.

**Table 5-29 Data Bus Configuration Word Settings for Chip Selects**

Bit(s)	Configuration Function Affected	Description
0	Address bus/chip selects	0 = $\overline{CS}[0:11]/ADDR[0:11]$ configured as address pins 1 = $\overline{CS}[0:11]/ADDR[0:11]$ configured as chip-select pins (default value)
1	Exception prefix (vector table location)	0 = Vector table begins at 0x0000 0000 1 = Vector table begins at 0xFFFF 0000 (default value)
2	Burst mode type	0 = Type 1 burst mode — uses $\overline{BDIP}$ timing (default value) 1 = Type 2 burst mode — uses $\overline{LAST}$ timing
3	ITYPE of boot device	0 = Boot device ITYPE = 0x08 (Synchronous burst) 1 = Boot device ITYPE = 0x01 (Asynchronous — default value)
4	Port size of boot device	0 = Boot device has 16-bit port (default value) 1 = Boot device has 32-bit port
6:8	$\overline{TA}$ delay for $\overline{CSBOOT}$	000 = 0 wait states 001 = 1 wait state 010 = 2 wait states 011 = 3 wait states 100 = 4 wait states 101 = 5 wait states 110 = 6 wait states 111 = 7 wait states (default value)

The boot region can be a ROM or flash EPROM. At power-on, it is assumed that no writing to the region is needed until the chip-select logic has been configured. Thus, no  $\overline{WE}$  is needed at power-on time. The boot device can be internal or external memory. If internal memory, the boot device is not under chip-select control. If the boot device is located externally, the chip-select logic decodes the address of the access and enables the  $\overline{CE}$  and  $\overline{OE}$  of the boot device appropriately. If the external boot device has a type 2 burst interface, the  $\overline{LAST}$  signal must be supplied by the EBI.

#### NOTE

At power-on, the boot region may be located at the uppermost or lowermost 1 Mbyte of the address range. The  $\overline{CSBOOT}$  base address (specified in the BA field of CSBTBAR) can be reset to 0xFFFF0 0000 or 0x0000 0000 accordingly. The chip-select logic asserts the  $\overline{CSBOOT}$  signal for the entire 1-Mbyte range if the access is to external boot memory.

While  $\overline{RESET}$  is asserted, the MCU drives the chip-select pins high (negated) to avoid a possible data bus conflict.

## 5.6 Clock Submodule



The system clock provides timing signals for the IMB2 and for an external peripheral bus. The MCU drives the system clock onto the external bus on the CLKOUT pin. The main timing reference for the MPC500 family is a 4-MHz crystal. The system operating frequency is generated through a programmable phase-locked loop. The PLL is programmable in integer multiples of 4 MHz to generate operating frequencies of 16 MHz to 44 MHz. These frequencies can be divided by powers of two to generate other frequencies.

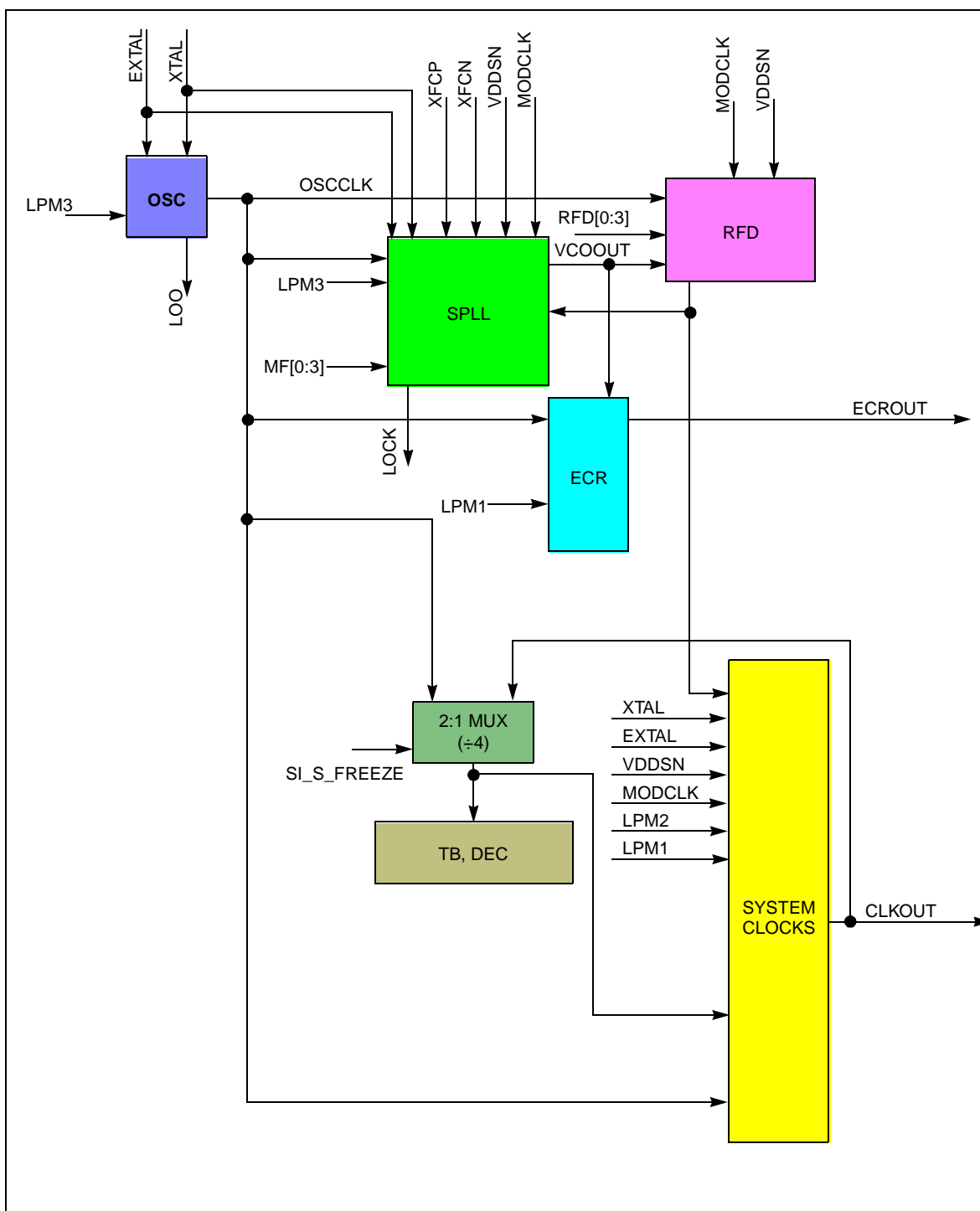
If the crystal ceases to function, the loss of oscillator (LOO) bit is set and the PLL is forced to operate in the self-clocked mode (SCM). This mode provides a system clock frequency of approximately 4 MHz. The exact frequency depends on the voltage and temperature of the CPU, but is optimized for nominal operating conditions.

The PLL can be bypassed by grounding the  $V_{DDSN}$  pin. Note that in this case, the input frequency needs to be twice the desired operating system frequency. With  $V_{DDSN}$  grounded, the multiplication factor (MF) bits in the system clock control register (SCCR) no longer have any effect on the system frequency, but the reduced frequency (RFD) bits and the low-power mode (LPM) bits do have an effect.

Three different low-power modes are available to minimize standby power usage. Normal operation or one of the three low-power modes is selected by programming the LPM bits in the SCCR.

The clock submodule also provides a clock source for the PowerPC time base and decremter. The oscillator, time base, and decremter are powered from the keep alive power supply (VDDKAP1). This allows the time base to continue incrementing even when the main power to the MCU is off. While the power is off, the decremter also continues to count. The power-down wakeup pin (PDWU) can be programmed to signal an external power-on reset circuit to enable power to the system whenever the MSB of the decremter changes from a zero to a one.

**Figure 5-26** is a block diagram of the SIU clock module.



SIU CLOCK BLOCK

**Figure 5-26 SIU Clock Module Block Diagram**

### 5.6.1 Clock Submodule Signal Descriptions

**Table 5-30** describes the signals used by the clock module.

**Table 5-30 Clocks Module Signal Descriptions**

Mnemonic	Name	Direction	Description
CLKOUT	System clock out	O	System clock. Used as the bus timing reference by external devices.
EXTAL, XTAL	Crystal oscillator	I/O	Connections for external crystal to the internal oscillator circuit. An external oscillator should serve as input to the EXTAL pin, when used.
XFCN, XFCP	External filter capacitor	I	These pins are used to add an external capacitor to the filter circuit of the phase-locked loop.
MODCLK	Clock mode select	I	The state of this input signal during reset selects the source of the system clock. Refer to <a href="#">5.6.3 System Clock Sources</a> .
V <sub>DDSN</sub> , V <sub>SSSN</sub>	Synthesizer power	I	These pins supply a quiet power source to the VCO.
ECROUT	Engineering clock reference output	O	Buffered output of the crystal oscillator. The ECROUT output frequency is equal to the crystal oscillator frequency.
PLLL/DSDO	PLL lock status or debug output	O	Phase-locked loop status output or debug output.
PDWU	Power-down wakeup	O	Asserted or negated, respectively, by software setting or clearing the WUR bit in the SCLSR. Also asserted when decremter counts down to zero. Can be used as power-down wakeup to external power-on reset circuit.

### 5.6.2 Clock Power Supplies

The power supply for each block of the clock submodule is shown in [Table 5-31](#).

**Table 5-31 Clock Module Power Supplies**

Power Supply	Blocks
V <sub>DDI</sub>	CLKOUT ECR PIT Clock RFD PLL (Digital)
V <sub>DDKAP1</sub>	Decrementer/time base clock Oscillator SCCR SCCSR
V <sub>DDSN</sub>	PLL (Analog)

To improve noise immunity, the PLL has its own set of power supply pins, V<sub>DDSYN</sub> and GND<sub>SYN</sub>. Only the charge pump and the VCO are powered by these pins.





The oscillator, system clock control register, and system clock control and status register are powered from the keep alive power supply (VDDKAP1) and  $V_{SS1}$ . In addition, VDDKAP1 powers the PowerPC time base and decremter. This allows the time base to continue incrementing at 1 MHz even when the main power to the MCU is off. While the power is off, the decremter also continues to count and may be used to signal the external power supply to enable power to the system at specific intervals. This is the power-down wakeup feature.

### 5.6.3 System Clock Sources

The  $V_{DDSN}$  and MODCLK pins are used to configure the clock source for the MCU. The configuration modes are shown in [Table 5-32](#).

**Table 5-32 System Clock Sources**

$V_{DDSN}$	MODCLK	PLL Options
1	1	Normal operation
1	0	1:1 mode (CLKOUT frequency is equal to oscillator frequency)
0	1	PLL bypass mode (CLKOUT frequency is equal to one half the oscillator frequency)
0	0	Special test mode

When both pins are high, the CPU clocks are configured for normal operation and the PLL is fully programmable.

If MODCLK = 0 and  $V_{DDSN}$  = 1, then the PLL enters 1:1 frequency mode. In this mode, CLKOUT frequency is equal to the oscillator frequency and is not affected by the RFD bits. The oscillator can be driven by either an external crystal or an external clock source.

If  $V_{DDSN}$  = 0 and MODCLK = 1, then the PLL is disabled and bypassed. In this case, CLKOUT frequency is equal to one half the oscillator frequency. In this mode, the oscillator source must have a 50% duty cycle.

In each clock mode except bypass mode, CLKOUT frequency can be reduced by programming the RFD field to a non-zero value. Refer to [5.6.5 CLKOUT Frequency Control](#) for details.

If  $V_{DDSN}$  = 0 and MODCLK = 0, then the CPU clocks are configured in special test mode. In this mode, the PLL and most of the clock generation circuitry are bypassed. This mode is intended for factory test only.

#### CAUTION

When the clock is in PLL bypass mode or special test mode, setting the LOLRE bit in the SCCR generates a loss-of-lock reset request (since the PLL is off). The LOLRE bit must not be set when the clock is in PLL bypass or special test mode.

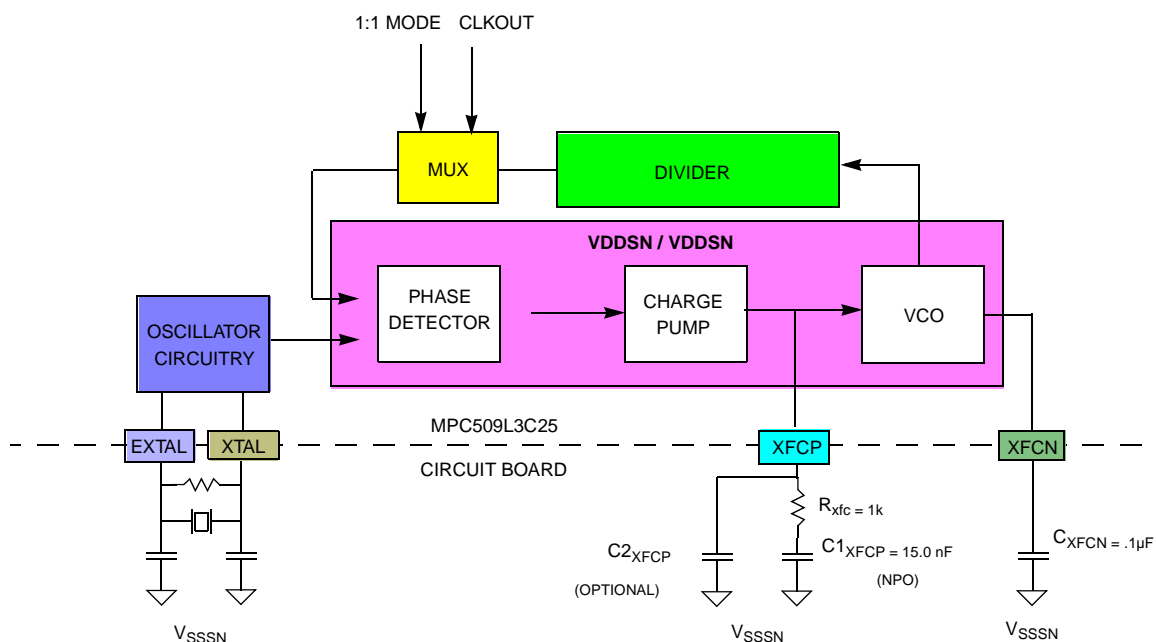
## 5.6.4 Phase-Locked Loop

The phase-locked loop (PLL) is a frequency synthesis PLL that can multiply the reference clock frequency by a factor from 4 to 11, provided the system clock (CLKOUT) frequency (when RFD = 0b000) remains within the specified limits. With a reference frequency of 4 MHz, the PLL can synthesize frequencies from 16 MHz to 44 MHz.

The output of the PLL can be divided down to reduce the system frequency with the reduced frequency divider (RFD). The RFD is not contained in the feedback loop of the PLL, so changing the RFD bits does not affect PLL operation.

Note that the system frequency programmed should not exceed the operating frequency of any of the parts in the target system.

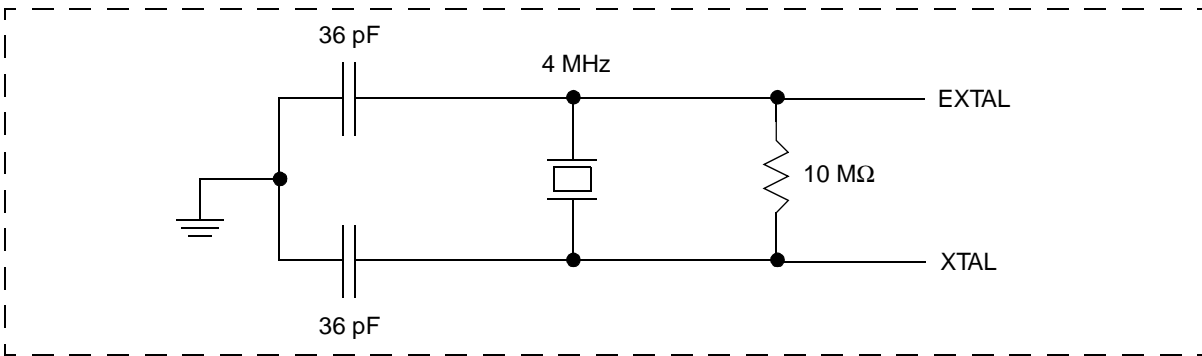
**Figure 5-27** shows the overall block diagram for the PLL. Each of the major blocks shown is discussed briefly below.



**Figure 5-27 Phase-Locked Loop Block Diagram**

### 5.6.4.1 Crystal Oscillator

The crystal oscillator has one 10-M $\Omega$  resistor and two external 36-pf capacitors connected to the EXTAL and XTAL pins, as shown in **Figure 5-28**. The internal oscillator is designed to work best with a 4-MHz crystal. Crystal start-up times depend on the value of the resistor. The start-up time can be reduced by reducing the value of the resistor.



**Figure 5-28 Crystal Oscillator**

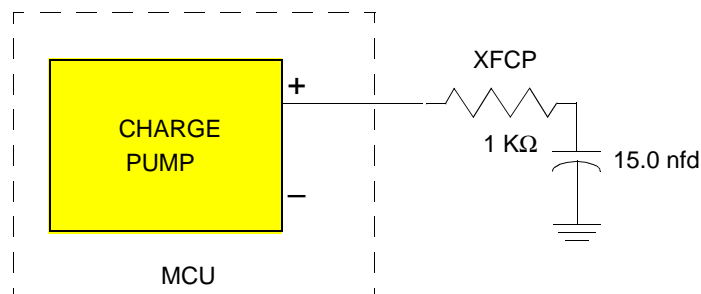
#### 5.6.4.2 Phase Detector

The phase detector compares both the phase and frequency of the reference clock (*oscclk* in [Figure 5-27](#)) 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 (the system clock) in 1:1 mode or the VCO output divided down by the MF divider in normal mode.

The phase detector pulses either the UP or DOWN signal, 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.

#### 5.6.4.3 Charge Pump and Loop Filter

The UP and DOWN signals from the phase detector control whether the charge pump applies or removes charge, respectively, from the loop filter. The loop filter is shown in [Figure 5-29](#). The filter network is external to the chip and can be replaced if necessary.



**Figure 5-29 Charge Pump with Loop Filter Schematic**

#### 5.6.4.4 VCO

The VCO uses a single-ended design with an external capacitor to increase noise immunity. The voltage on XFCP controls the VCO output frequency. The frequency-to-voltage relationship (VCO gain) is positive, and the output frequency is twice the maximum target system frequency.



#### 5.6.4.5 Multiplication Factor Divider

The multiplication factor divider (MFD) divides down the output of the VCO and feeds it back to the phase detector (when the PLL is not operating in 1:1 mode). The phase detector 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. When the PLL is operating in 1:1 mode, the MFD is bypassed and the effective multiplication factor is one. Refer to [5.6.5 CLKOUT Frequency Control](#) for details on setting system clock frequency with the MF and RFD bits.

#### 5.6.4.6 Clock Delay

Besides frequency synthesis, the PLL must also align the phase of (i.e., phase lock) the reference and system clocks to ensure proper system timing. Since the purpose of the RFD is to allow the user to change the system frequency without forcing the PLL to re-lock, the feedback clock must originate before the RFD (i.e., the output of the VCO).

The clock delay is a chain of gates that approximates the delay through the RFD, clock generation circuits, metal routing and the CLKOUT driver. This approach does not allow for precise phase alignment. System applications must not rely on precise phase alignment between the reference and system clocks when the PLL is operating in normal (frequency synthesis) mode. In 1:1 mode, however, the RFD is disabled. The feedback clock comes directly from the CLKOUT pin, and true phase lock is achieved.

#### 5.6.5 CLKOUT Frequency Control

The multiplication factor (MF) and reduced frequency divide (RFD) fields in the SCCR determine the system clock (CLKOUT) frequency. [Table 5-33](#) summarizes the available CLKOUT frequencies with a 4-MHz crystal.



**Table 5-33 CLKOUT Frequencies with a 4-MHz Crystal<sup>1</sup>**

RFD[0:3]	CLKOUT (Hz)							
	MF = X000 (x4)	MF = X001 (x5)	MF = X010 (x6)	MF = X011 (x7)	MF = X100 (x8)	MF = X101 (x9)	MF = X110 (x10)	MF = X111 (x11)
0 = 0000 (÷ 1)	16.000 M	20.000 M	24.000M	28.000 M	32.000 M	36.000 M	40.000 M	44.000 M
1 = 0001 (÷ 2)	8.000 M	10.000 M	12.000 M	14.000 M	16.000 M	18.000 M	20.000 M	22.000 M
2 = 0010 (÷ 4)	4.000 M	5.000 M	6.000 M	7.000 M	8.000 M	9.000 M	10.000 M	11.000 M
3 = 0011 (÷ 8)	2.000 M	2.500 M	3.000 M <sup>2</sup>	3.500 M	4.000 M	4.500 M	5.000 M	5.500 M
4 = 0100 (÷ 16)	1.000 M	1.250 M	1.500 M	1.750 M	2.000 M	2.250 M	2.500 M	2.750 M
5 = 0101 (÷ 32)	0.500 M	0.625 M	0.750 M	0.875 M	1.000 M	1.125 M	1.250 M	1.3750 M
6 = 0110 (÷ 64)	0.250 M	0.313 M	0.375 M	0.438 M	0.500 M	0.563 M	0.625 M	0.688 M
7 = 0111 (÷ 128)	0.125 M	0.156 M	0.188 M	0.219 M	0.250M	0.281 M	0.313 M	0.344 M
8 = 1000 (÷ 256)	62.500 K	78.125 K	93.750 K	0.109 M	0.125 M	0.141 M	0.156 M	0.172 M
9 = 1001 (÷ 512)	31.250 K	39.063 K	46.875 K	54.688 K	62.500 K	70.313 K	78.125 K	85.938 K
10 = 1010 (÷ 1024)	15.625 K	19.531 K	23.438 K	27.344 K	31.250 K	35.156 K	39.063 K	42.969 K
11 = 1010 (÷ 1024)	15.625 K	19.531 K	23.438 K	27.344 K	31.250 K	35.156 K	39.063 K	42.969 K
12 = 1010 (÷ 1024)	15.625 K	19.531 K	23.438 K	27.344 K	31.250 K	35.156 K	39.063 K	42.969 K
13 = 1010 (÷ 1024)	15.625 K	19.531 K	23.438 K	27.344 K	31.250 K	35.156 K	39.063 K	42.969 K
14 = 1010 (÷ 1024)	15.625 K	19.531 K	23.438 K	27.344 K	31.250 K	35.156 K	39.063 K	42.969 K
15 = 1010 (÷ 1024)	15.625 K	19.531 K	23.438 K	27.344 K	31.250 K	35.156 K	39.063 K	42.969 K

**NOTES:**

1. Settings resulting in CLKOUT frequencies in the shaded areas should not be used in the MPC509.
2. Default setting.

Whenever clock reset is asserted, the MF bits are set to 0x2 (multiply by six), and the RFD bits are set to 0x3 (divide by eight). These values program the PLL to generate the default system frequency of 3 MHz when a 4-MHz crystal is used.

### 5.6.5.1 Multiplication Factor (MF) Bits

The MF bits determine the operating frequency of the PLL. The 4-MHz crystal reference frequency is multiplied by an integer from 4 to 11, depending on the value of the MF bits, resulting in a PLL frequency of 16 MHz to 44 MHz. [Table 5-34](#) summarizes the effect of the MF bits.



**Table 5-34 Multiplication Factor Bits**

MF Field (Binary)	Multiplication Factor	PLL Frequency (with 4-MHz Reference)
x000	4	16 MHz
x001	5	20 MHz
x010	6	24 MHz
x011	7	28 MHz
x100	8	32 MHz
x101	9	36 MHz
x110	10	40 MHz
x111	11	44 MHz

**NOTE 1**

The PLL frequency is equal to the CLKOUT frequency when the RFD bits are programmed to 0b0000 (divide by one).

**NOTE 2**

That the value of MF0 is ignored. However, this bit should be written to zero in case it is used in future implementations.

**CAUTION**

The MF bits must not be programmed to a value that requires the VCO to operate at greater than 180 MHz. ( $F_{VCO} = 4 \times \text{MF factor} \times F_{REF}$ .) Specifically, if the reference crystal frequency is 5 MHz, the MF settings of X110 (times 10) and X111 (times 11) must be avoided.

The MF bits can be read and written at any time. However, the MF bit field can be write-protected by setting the MF lock (MPL) bit in the SCSLR.

Changing the MF bits causes the PLL to lose lock. If the loss-of-lock reset enable bit (LOLRE) is set, the loss-of-lock condition causes the clock module to signal a reset condition to the reset controller. The reset controller may wait for the PLL to lock before negating reset. Thus, the PLL can still be out of lock when  $\overline{\text{RESETOUT}}$  is negated. After changing the MF bits, software should monitor the system PLL lock status (SPLS) bit in the SCSLR to determine lock status.

The RFD bits should always be written to a value of 0x1 or greater before changing the MF bits to ensure the system frequency does not exceed the system's design margin, since the VCO overshoots in frequency as it tries to compensate for the change in frequency. For example, to change from the default system frequency to 16 MHz, write the RFD bits to 0x1, then write the MF bits to 0x0. After the PLL locks, write the RFD bits to 0x0.

When the PLL is operating in one-to-one mode, the multiplication factor is set to one and MF is ignored.



**Figure 5-27** shows how the PLL uses the MF bits to multiply the input crystal frequency. The output of the VCO is divided down to generate the feedback signal to the phase comparator. The MF bits control the value of the divider in the PLL feedback loop. The phase comparator determines the phase shift between the feedback signal and the reference clock. This difference results in either an increase or decrease in the VCO output frequency.

#### 5.6.5.2 Reduced Frequency Divider (RFD[0:3])

The RFD bits control a prescaler at the output of the PLL. The reset state of the RFD bits is 0x3, which divides the output of the VCO by eight.

These bits can be changed without affecting the PLL's VCO, (i.e., no re-lock delay is incurred). All changes in frequency are synchronized to the next falling edge of the current system clock.

**Table 5-35** summarizes the possible values for the RFD bits.

**Table 5-35 Reduced Frequency Divider Bits**

RFD Field (Binary)	Divider
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	512
1010	1024
1011	1024
1100	1024
1101	1024
1110	1024
1111	1024

These bits can be read at any time. They should be written only when the system PLL lock status bit (SPLS) is set. Writing the RFD bits, especially to 0x0, when the PLL is not locked can cause the clock frequency to surpass the system operating frequency. Software is responsible for monitoring the SPLS bit and preventing a write to RFD[0:3] while the PLL is out of lock.



The RFD bits should always be written to a value of 0x1 or greater before changing the MF bits to ensure the system frequency does not exceed the system's design margin since the VCO overshoots in frequency as it tries to compensate for the change in frequency. The RFD bits should be changed to their final value only after the MF bits have been written to their final value and PLL lock at the new frequency has been established. For example, to change from the default system frequency to 16 MHz, write the RFD bits to 0x1, then write the MF bits to 0x0. After the PLL locks, write the RFD bits to 0x0.

The RFD bits can be protected against further writes by setting the RFD lock (RFDL) bit in the SCSLR register.

#### **NOTE**

The RFD bits do not affect clock frequency when the system clock is operating in 1:1 mode.

### **5.6.6 Low-Power Modes**

The clock module provides one normal operating mode and three low-power modes. The low-power mode (LPM) bits in the SCCR select one of these four modes. When one of the three low-power modes is selected, the EBI prevents the CPU from starting any more bus cycles, but allows the current bus cycle to terminate. At the end of the current bus cycle, the appropriate clocks are stopped and the EBI continues operation as defined for the low-power mode selected.

#### **NOTE**

In debug mode, the crystal and the PLL are not shut down, but continue to run and provide clocks to the debug module.

The LPM bits can be protected against further writes by setting the LPM lock (LPML) bit in the SCLSR register.

#### **5.6.6.1 Normal Mode**

The normal operating mode, state 0x0, is the state out of clock reset. This is also the state the bits go to when the low-power mode exit signal arrives.

#### **5.6.6.2 Single-Chip Mode**

Mode 0x1 is single-chip mode. In this mode, CLKOUT is turned off. This mode can be selected when the MCU is used by itself and does not need to provide a system clock. Turning off CLKOUT saves power and improves electromagnetic compatibility. The low-power mode exit signal is the logical OR of the external reset pin, the  $\overline{\text{IRQ}}[0:1]$  pins (if LPMM = 1), the decremter interrupt, and the PIT interrupt.



Since the oscillator and PLL are still running and locked, the low-power mode exit signal must be a minimum of two system clock cycles and exiting this state does not incur a PLL lock time.



### 5.6.6.3 Doze Mode

Mode 0x2 is doze mode. In this state, not only is CLKOUT turned off, but also all internal clocks are turned off. However, the oscillator and the PLL continue to operate normally. The low-power mode exit signal is the logical OR of the external reset pin, the  $\overline{\text{IRQ}}[0:1]$  pins (if LPMM = 1), the decremter interrupt, and the PIT interrupt. Since the oscillator and PLL are still running and locked, the low power mode exit signal must be a minimum of two system clock cycles and exiting this state does not incur a PLL lock time.

### 5.6.6.4 Sleep Mode

Mode 0x3 is sleep mode. In this state, *all* clocks are turned off, including the oscillator and the PLL. The low-power mode exit signal is the logical OR of the external reset pin and the  $\overline{\text{IRQ}}[0:1]$  pins (if LPMM = 1). The decremter interrupt and the PIT interrupt are not active in this mode. Since all clocks are stopped, this signal is asynchronous. Exiting state 0x3 requires the normal crystal start-up time plus PLL lock time or time-out.

### 5.6.6.5 Exiting Low-Power Mode

**Table 5-36** summarizes the events that cause the MCU to exit from each of the three low-power modes.

**Table 5-36 Exiting Low-Power Mode**

Event Causing Exit from Low-Power Mode	Mode 0x1	Mode 0x2	Mode 0x3
$\overline{\text{RESETOUT}}$ assertion	Yes	Yes	Yes
$\overline{\text{IRQ}}$ pin assertion (if LPMM = 1)	Yes	Yes	Yes
Decrementer interrupt	Yes	Yes	No
PIT interrupt	Yes	Yes	No

The low-power mode mask (LPMM) bit in the SCCR is used to mask the  $\overline{\text{IRQ}}[0:1]$  pins to the low-power mode exit logic. When the LPMM bit is zero, the  $\overline{\text{IRQ}}[0:1]$  pins are disabled from causing an exit from any of the low-power modes. When the LPMM bit is a one, the  $\overline{\text{IRQ}}[0:1]$  pins are enabled to cause an exit from any of the low-power modes.

When a low level occurs at the  $\overline{\text{IRQ}}[0:1]$  pins and the LPMM bit is a one, the LPM bits are cleared and the low-power mode exit sequence begins. If the LPMM bit = 1, then the low-power mode exit sequence is started even if a low-power mode is not selected.

The time required to exit the low-power modes depends on which mode was selected. For modes 1 and 2, the delay from the exit signal being asserted to the clocks starting

up is two clock cycles of the frequency that the VCO was programmed to generate. The delay for mode 3 is the crystal start-up time plus the VCO lock time.



The LPMM bit can be read or written any time.

### 5.6.7 System Clock Lock Bits

The system clock lock and status register (SCLSR) contains several bits that lock the corresponding bits or fields in the system clock control register (SCCR). [Table 5-37](#) summarizes the lock bits and the fields that they control.

**Table 5-37 System Clock Lock Bits**

SCLSR Lock Bit	SCCR Field Affected
MPL	MF
LPML	LPM
RFDL	RFD

When a lock bit (MPL, LPML, or RFDL) is cleared, writes to the corresponding bit or field in the SCCR (MF, LPM or RFD, respectively) take effect. When the lock bit is set, however, writes to the corresponding bit or field in the SCCR have no effect. All other bits in the SCCR are unaffected.

The MPL, LPML, and RFDL bits can be written to zero as many times as required. Only a clock reset can clear one of these bits, however, once it is written to a one. In freeze mode, the lock bits can be written to a one or to a zero at any time.

### 5.6.8 Power-Down Wake Up

PDWU (power-down wake up) is an output signal used to signal an external power supply to enable power to the system. PDWU can be asserted by the decremter counting down to zero or by the CPU setting the wake-up request (WUR) bit in the SCLSR.

The WUR bit controls the state of the PDWU pin directly. The WUR bit is set when the CPU writes a one to it or when the MSB of the decremter changes from a zero to a one. WUR is cleared when the CPU writes a zero to it. (Note that to clear the bit, it is *not* necessary to read the bit as a one before writing it as a zero, as is required to clear most status bits.)

#### NOTE

The WUR bit is not affected by resets and its value should remain as long as the keep-alive power is valid. At keep-alive reset, the WUR bit is in an unknown state.

### 5.6.9 Time Base and Decrementer Support



The time base is a timer facility defined by the PowerPC architecture. It is a 64-bit free-running binary counter which is incremented at a frequency determined by each implementation of the time base. There is no interrupt or other indication generated when the count rolls over. The period of the time base depends on the driving frequency. The time base is not affected by any resets and should be initialized by software.

The decrementer is a 32-bit decrementing counter defined by the PowerPC architecture. The decrementer causes an interrupt, unless masked by MSR[EE], when it passes through zero.

The time base and decrementer use the stand-by power supply, VDDKAP1. This allows them to be used while normal power is off. The time base and decrementer also continue to function in all low-power modes except sleep mode (LPM = 0b11), in which the oscillator is turned off.

The state of the decrementer and time base after standby power is restored is indeterminate. The decrementer runs continuously after power-up (unless the decrementer clock enable bit is cleared). System software is necessary to perform any initialization. The decrementer is not affected by reset and continues counting while reset is asserted.

Reads and writes of the time base and decrementer are restricted to special instructions. Refer to **SECTION 3 CENTRAL PROCESSING UNIT** and to the *RCPU Reference Manual* (RCPURM/AD) for instructions on reading and writing the time base and decrementer.

#### 5.6.9.1 Time Base and Decrementer Clock Source

The frequency source is the crystal oscillator divided by four

The clock source for the time base and decrementer is the crystal oscillator divided by four. With a 4-MHz oscillator frequency, the period for the time base is:

$$T_{TB} = 2^{64} / 1 \text{ MHz} = 1.8 \times 10^{13} \text{ seconds}$$

which is approximately 585,000 years.

With the same clock source, the period for the decrementer is:

$$T_{DEC} = 2^{32} / 1 \text{ MHz} = 4295 \text{ seconds}$$

which is approximately 71.6 minutes.

#### 5.6.9.2 Time Base/Decrementer and Freeze Assertion

The assertion of the global freeze signal can stop the clock to the time base and decrementer if the SIUFRZ bit in the SIUMCR (0x8007 FC00) is set. The actual freezing of the clock source occurs at the falling edge of the clock.

### 5.6.9.3 Decrementer Clock Enable (DCE) Bit

The decrementer clock enable (DCE) bit in the SCCR enables or disables the clock source to the decrementer. The default state is to have the clock enabled. The actual clock source is determined by the TBS bit. The DCE bit does not affect the decrementer until after the next increment time, as determined by the clock source.



### 5.6.10 Clock Resets

The following reset conditions cause the internal clock reset signal to be asserted: external reset, loss-of-oscillator (when LOORE is set), and loss-of-lock (when LOLRE is set). Clock reset causes the clock circuitry, including the PLL, oscillator, SCCR, and SCLSR to be reset.

Note that all reset sources cause normal reset processing to occur, as described in [5.8 Reset Operation](#). However, only the reset sources mentioned above (external reset, loss-of-oscillator, and loss-of-lock) result in clock reset.

#### 5.6.10.1 Loss of PLL Lock

The system PLL lock status (SPLS) in the SCLSR indicates the current lock status of the PLL. When the SPLS bit is clear, the PLL is not locked. When the SPLS bit is set, the PLL is locked. The SPLS bit can be read anytime. It can be written only during special test mode.

In PLL bypass mode and special test mode, this bit is forced high to indicate a lock condition.

The loss-of-lock reset enable (LOLRE) bit in the SCCR indicates how the clocks should handle a loss of lock indication (SPLS asserted). When LOLRE is clear, clock reset is not asserted if a loss of lock indication occurs. When LOLRE is set, clock reset is asserted when a loss of lock indication occurs. The reset module may wait for the PLL to lock before negating reset (see [5.8 Reset Operation](#)). The LOLRE bit is cleared whenever clock reset is asserted and may be re-initialized by software.

#### CAUTION

When the clock is in PLL bypass mode or special test mode, setting the LOLRE bit in the SCCR generates a loss-of-lock reset request (since the PLL is off). The LOLRE bit must not be set when the clock is in PLL bypass or special test mode.

The system PLL lock status sticky bit (SPLSS) in the SCLSR must be initialized by software. After the bit is set by software, any out-of-lock indication clears the SPLSS bit, even if the out-of-lock indication is active while the setting takes place. The bit remains clear until software again sets it. At clock reset, the state of the SPLSS bit is zero, since the PLL has not achieved lock.

#### 5.6.10.2 Loss of Oscillator

The loss-of-oscillator (LOO) status bit in the SCLSR indicates the absence of an input frequency on the EXTAL pin. A frequency below 125 kHz causes the loss-of-oscillator

circuitry to assert the LOO bit and force the PLL into self-clocked mode. A frequency above 500 kHz causes the loss-of-oscillator circuitry to negate the LOO bit, and the PLL operates normally. The LOO bit can be read any time. It can be written only in special test mode.



The loss-of-oscillator reset enable (LOORE) bit in the SCCR indicates how the clock module should handle a loss-of-oscillator condition (LOO asserted). When LOORE is clear, clock reset is not asserted if a loss-of-oscillator indication occurs. When LOORE is set, clock reset is asserted when a loss-of-oscillator indication occurs. The reset module may wait for the PLL to lock before negating reset (see [5.8 Reset Operation](#)). The LOORE bit is cleared when clock reset is asserted and may be re-initialized by software.

Note that a loss of oscillator forces the PLL to go out of lock and into the self-clocked mode (SCM), regardless of the state of the LOORE bit.

### 5.6.11 System Clock Control Register (SCCR)

The SCCR controls the operation of the PLL. It is powered by VDDKAP1. The SCCR is not affected by reset conditions that do not cause clock reset. Clock reset (caused by loss-of-oscillator, loss-of-lock, or external reset) causes the register to be reset as indicated in the diagram.

#### SCCR — System Clock Control Register

0x8007 FC50

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RESERVED			LPMM	0	DCE	LOL-RE	LOORE	0	MF			RESERVED			

CLOCK RESET:

0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED						LPM		RESERVED				RFD			

CLOCK RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1



**Table 5-38 SCCR Bit Settings**

Bit(s)	Name	Description
0:2	—	Reserved
3	LPMM	Low-power mode mask 0 = $\overline{\text{IRQ}}[0:1]$ pins cannot be used to wake up from LPM 1 = $\overline{\text{IRQ}}[0:1]$ pins can be used to wake up from LPM
4	—	Reserved
5	DCE	Decrementer clock enable 0 = Clock to decrementer is disabled 1 = Clock to decrementer is enabled
6	LOLRE	Loss-of-lock reset enable 0 = Loss of lock does not cause reset 1 = Loss of lock causes reset
7	LOORE	Loss-of-oscillator reset enable 0 = Loss of oscillator does not cause reset 1 = Loss of oscillator causes reset
8	—	Reserved
9:12	MF	Multiplication factor. In normal mode, the output of the VCO is divided down to generate the feedback signal to the phase comparator. The MF field controls the value of the divider in the PLL feedback loop.  The MF and RFD fields determine the CLKOUT frequency. Refer to <a href="#">Table 5-33</a> . X000 = x 4 X001 = x 5 X010 = x 6 X011 = x 7 X100 = x 8 X101 = x 9 X110 = x 10 X111 = x 11  Caution: The MF bits must not be programmed to a value that requires the VCO to operate at greater than 180 MHz. ( $F_{\text{VCO}} = 4 \times \text{MF factor} \times F_{\text{REF}}$ .) Specifically, if the reference crystal frequency is 5 MHz, the MF settings of X110 (times 10) and X111 (times 11) must be avoided.
13:21	—	Reserved
22:23	LPM	Low-power mode select bits. Refer to <a href="#">Table 5-35</a> and <a href="#">Table 5-36</a> . 00 = Normal operating mode 01 = Low-power mode 1 (single chip) 10 = Low-power mode 2 (doze) 11 = Low-power mode 3 (sleep)  Since all clocks are stopped in sleep mode, exiting this mode requires the normal crystal start-up time plus the PLL lock time. Minimum length of the exit signal is two clocks in single-chip mode, three clocks in doze mode, and until the PLL is stable in sleep mode.
24:27	—	Reserved

**Table 5-38 SCCR Bit Settings (Continued)**



Bit(s)	Name	Description
28:31	RFD	<p>Reduced-frequency divider. The RFD field controls a prescaler at the output of the PLL. In normal mode, the MF and RFD fields determine the CLKOUT frequency. (In bypass mode, only this field, and not the MF field, affects CLKOUT frequency. In one-to-one mode, this field has no effect.) Refer to <a href="#">Table 5-33</a>.</p> <p>0000 = ÷ 1  0001 = ÷ 2  0010 = ÷ 4  0011 = ÷ 8  0100 = ÷ 16  0101 = ÷ 32  0110 = ÷ 64  0111 = ÷ 128  1000 = ÷ 256  1001 = ÷ 512  1010 = ÷ 1024  1011 = ÷ 1024  1100 = ÷ 1024  1101 = ÷ 1024  1110 = ÷ 1024  1111 = ÷ 1024</p>

#### 5.6.12 System Clock Lock and Status Register (SCLSR)

The system clock lock and status register (SCLSR) contains lock and status bits for the PLL. It is powered by VDDKAP1. The SCLSR is not affected by reset conditions that do not cause clock reset. Clock reset (caused by loss-of-oscillator, loss-of-lock, or external reset) causes the register to be reset as indicated in the diagram.

#### SCLSR — System Clock Lock and Status Register

**0x8007 FC54**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RESERVED				STME	MPL	LPML	RFDL	RESERVED						STMS	
CLOCK RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED							WUR	RESERVED					SPLS S	SPLS	LOO
CLOCK RESET:															
0	0	0	0	0	0	0	U	0	0	0	0	0	U	U	U

U = Unaffected by clock reset



**Table 5-39 SCLSR Bit Settings**

Bit(s)	Name	Description
0:3	—	Reserved
4	STME	System PLL test mode enable 0 = Test mode disabled 1 = Test mode enabled
5	MPL	MF lock 0 = Writes to MF field (in SCCR) allowed 1 = Writes to MF field have no effect
6	LPML	Low-power mode lock 0 = Writes to LPM bits allowed 1 = Writes to LPM bits have no effect
7	RFDL	Reduced-frequency divide lock 0 = Writes to RF bits allowed 1 = Writes to RF bits have no effect
8:13	—	Reserved
14:15	STMS	System PLL test mode select 00 = Normal operation 00, 01, 11 = Special test modes (for factory test only)
16:22	—	Reserved
23	WUR	Wake-up request 0 = PDWU pin forced low (request power off) 1 = PDWU pin forced high (request power on)
24:28	—	Reserved
29	SPLSS	System PLL lock status sticky bit 0 = PLL has gone out of lock since software last set this bit 1 = PLL has remained in lock since software last set this bit
30	SPLS	System PLL lock status 0 = PLL has not locked 1 = PLL has locked
31	LOO	Loss-of-oscillator status 0 = Clock detected 1 = Crystal not detected

## 5.7 System Protection

SIU system protection features include a periodic interrupt timer and a bus monitor.

Additional MPC509 system protection features include the PowerPC decremter and time base, described in [SECTION 3 CENTRAL PROCESSING UNIT](#), and a software watchdog, described in [6.4 Software Watchdog](#).

### 5.7.1 System Protection Features

- The bus monitor monitors any internal-to-external bus accesses. Four selectable response time periods are available, ranging from 16 to 256 system clock cycles. An internal bus error signal is generated if a bus time-out occurs.



- The periodic interrupt timer generates an interrupt after a period specified by the user.



## 5.7.2 System Protection Registers

**Table 5-40** shows the SIU system protection registers.

**Table 5-40 System Protection Address Map**

Access	Address	Register
S/U	0x8007 FC40	Periodic interrupt control and select register (PICSR)
S/U	0x8007 FC44	Periodic interrupt timer register (PIT)
S	0x8007 FC48	Bus monitor control register (BMCR)

## 5.7.3 Periodic Interrupt Timer (PIT)

The periodic interrupt timer consists of a 16-bit counter clocked by the input oscillator signal divided by four. A 4-MHz system oscillator frequency results in a one-microsecond count interval. The input clock signal is supplied by the clock module. In order to ensure adequate range for the PIT, the input clock signal to the PIT must not exceed 4 MHz.

The 16-bit counter counts down to zero when loaded with a value from the PIT count (PITC) field in the PICSR. After the timer reaches zero, the PIT status (PS) bit is set and an interrupt is generated if the PIT interrupt enable (PIE) bit is set to one.

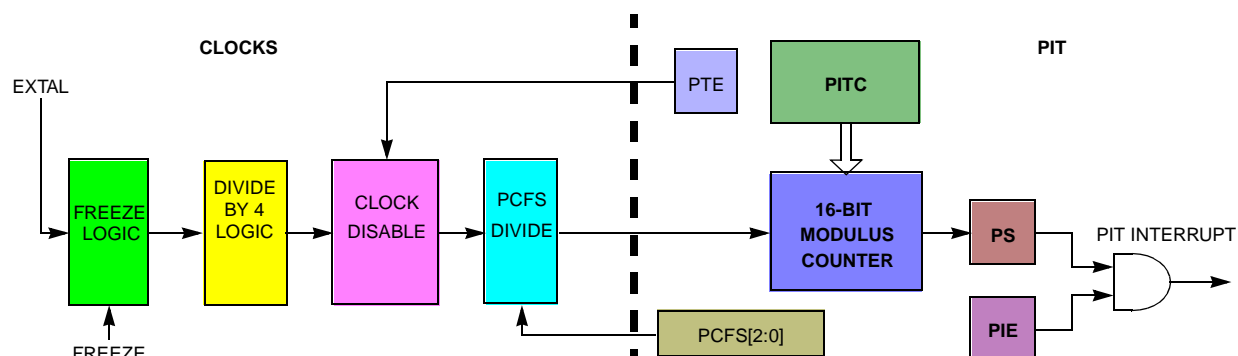
At the next input clock edge, the value in the PITC is loaded into the counter, and the process starts over. When a new value is loaded into the PITC field, the periodic timer is updated (i.e., the new value is loaded into the modulus counter), and the counter begins counting.

The software service routine should read the PS bit and then write it to zero to terminate the interrupt request. The interrupt request remains pending until the PS bit is cleared. If the counter reaches zero again before the interrupt service routine clears the PS bit, the interrupt request remains pending until PS is cleared.

Any write to the PITC stops the current countdown, and the count resumes with the new value in PITC. If the PITC is loaded with the value 0, the PIT counts for the maximum period.

If the PIT enable (PTE) bit is not set, the PIT is unable to count and retains the old count value. Reads of the PIT register have no effect on the value in the PIT.

**Figure 5-30** is a block diagram of the PIT.



**Figure 5-30 Periodic Interrupt Timer Block Diagram**

### 5.7.3.1 PIT Clock Frequency Selection

The PIT clock frequency select (PCFS) field in the PICSIR selects the appropriate frequency for the PIT clock source over a range of external clock or crystal frequencies. The bit encodings are shown in [Table 5-41](#).

**Table 5-41 PCFS Encodings**

PCFS Encoding	Divide Input Frequency by:
0b000	4
0b001	8
0b010	16
0b011	32
0b100	64
0b101	Reserved
0b110	Reserved
0b111	Reserved

To achieve a PIT setting of approximately 1 MHz, program the PCFS field as shown in [Table 5-42](#).

**Table 5-42 Recommended Settings for PCFS[0:2]**

Input Frequency Range	PCFS[0:2]
$1 \text{ MHz} < \text{FREQ} \leq 4 \text{ MHz}$	0b000
$4 \text{ MHz} < \text{FREQ} \leq 8 \text{ MHz}$	0b001
$8 \text{ MHz} < \text{FREQ} \leq 16 \text{ MHz}$	0b010

**Table 5-42 Recommended Settings for PCFS[0:2]**

Input Frequency Range	PCFS[0:2]
16 MHz < FREQ ≤ 32 MHz	0b011
32 MHz < FREQ ≤ 64 MHz	0b100
Reserved	0b101
Reserved	0b110
Reserved	0b111

### 5.7.3.2 PIT Time-Out Period Selection

The PIT time-out period is determined by the input clock frequency, the divider specified in the PCFS field, and the timing count specified in the PITC field of the PICSR.

The time-out period is calculated as follows:

$$\text{PIT period} = \text{PITC} / (\text{PIT frequency})$$

where the PIT frequency is equal to the PIT input clock frequency divided by a divisor determined by the PCFS bits, as specified in [5.7.3.1 PIT Clock Frequency Selection](#).

With a 4-MHz clock frequency and a PCFS value of 0b000 (divide by 4, reset value), this gives a range from 1 μs (PITC = 0x0001) to 65.5 ms (PITC = 0x0000).

**Table 5-43 Example PIT Time-Out Periods**

PITC Value	Time-Out Period <sup>1</sup>
1 (decimal)	1 μs
5 (decimal)	5 μs
10 (decimal)	10 μs
100 (decimal)	100 μs
1000 (decimal)	1.00 ms
10000 (decimal)	10.00 ms
50000 (decimal)	50.0 ms
FFFF (hex)	65.5 ms
0000 (hex) <sup>2</sup>	65.5 ms

**NOTES:**

1. After a time-out is signaled, some additional time may elapse prior to any observed action.
2. The count value associated with the maximum time-out is 0b0000.

5.7.3.3 PIT Enable Bits

The PIT enable (PTE) bit in the PICSR enables or disables the timer. When the timer is disabled, it retains its current value. When the timer is enabled, it resumes counting starting with the current value.

The periodic interrupt enable (PIE) bit in the PICSR enables or disables PIT interrupts. When this bit is cleared, the PIT does not generate any interrupts. The PIT continues to count even when interrupts are disabled.



5.7.3.4 PIT Interrupt Request Level and Status

The PIT interrupt request level (PITIRQL) field in the PIT/port Q interrupt level register (PITQIL) determines the level of PIT interrupt requests. Refer to [6.5.3.4 PIT/Port Q Interrupt Levels Register](#) for a description of this register.

The PIT status (PS) bit is set when the PIT issues an interrupt request. This occurs when the modulus counter counts to zero. The PS bit is cleared by writing it to zero after reading it as a one. Attempting to write this bit to one has no effect.

5.7.3.5 Periodic Interrupt Control and Select Register

The periodic interrupt control and select register (PICSR) contains the interrupt status bit as well as the controls for the 16 bits to be loaded into a modulus counter. Reserved bits in this register return zero when read. This register can be read or written at any time.

PICSR — Periodic Interrupt Control and Select Register 0x8007 FC40

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	PTE	PIE	RESERVED	PCFS				RESERVED							PS
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
PITC															
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**Table 5-44 PICSR Bit Settings**

Bit(s)	Name	Description
0	—	Reserved
1	PTE	Periodic timer enable 0 = Disable decremter counter 1 = Enable decremter counter
2	PIE	Periodic interrupt enable 0 = Disable periodic interrupt 1 = Enable periodic interrupt Caution: Be sure the EE (external interrupts enable) bit in the MSR is cleared before changing the value of this bit.
3:4	—	Reserved
5:7	PCFS	PIT clock frequency select. To achieve PIT setting of approximately 1 MHz, program the PCFS field as shown in <a href="#">Table 5-42</a> .
8:14	—	Reserved
15	PS	PIT status 0 = No PIT interrupt asserted 1 = Periodic interrupt asserted
16:31	PITC	Periodic interrupt timing count. Number of counts to load into the PIT.

### 5.7.3.6 Periodic Interrupt Timer Register

The periodic interrupt timer (PIT) register is a read-only register that shows the current value in the periodic interrupt down counter. Writes to this register have no effect. Reads of the register do not affect the counter.

#### PIT — Periodic Interrupt Timer Register

**0x8007 FC44**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED																PIT															

RESET: UNDEFINED

### 5.7.4 Hardware Bus Monitor

Typical bus systems require a bus monitor to detect excessively long data and address acknowledge response times. The MPC509 provides a bus monitor to monitor internal-to-external bus accesses on the E-bus. If the external bus pipeline depth is zero (all previous external bus cycles are complete), the monitor counts from transfer start to transfer acknowledge. Otherwise, the monitor counts from transfer acknowledge to transfer acknowledge.

If the monitor times out, transfer error acknowledge ( $\overline{\text{TEA}}$ ) is asserted internally.

The bus monitor is always enabled (regardless of the value of the BME bit in the BMCR) while the internal freeze signal is asserted and debug mode is enabled, or while debug mode is enabled and the debug non-maskable breakpoint is asserted.



### 5.7.4.1 Bus Monitor Timing

The bus monitor timing (BMT) field in the BMCR allows the user to select one of four selectable response time periods. Periods range from 16 to 256 system clock cycles. The programmability of the time-out allows for a variation in system peripheral response time. The timing mechanism is derived from taps off a divider chain which is clocked by the system clock.

The time-out period should be set for the maximum total cycle time (including all beats of a burst, i.e., until  $\overline{TA}$  is asserted for the final beat of a burst cycle), not for just the address phase or data phase of the cycle.

### 5.7.4.2 Bus Monitor Lock

The bus monitor lock (BMLK) bit in the BMCR is used to prevent inadvertent writes to the BMCR. Once BMLK is set, subsequent writes to the BMCR have no effect and result in a data error on the internal bus.

Writing a zero to BMLK after it has been set has no effect. A write to the BMCR before the lock bit is set can configure protected bits and set the BMLK in the same access.

The BMLK bit is cleared by reset. It can also be cleared by software while the internal FREEZE signal is asserted. Software can write BMLK to zero any number of times before writing it to one.

### 5.7.4.3 Bus Monitor Enable

The bus monitor enable (BME) bit in the BMCR enables or disables the operation of the bus monitor during internal-to-external bus cycles. Note that the bus monitor is always enabled while freeze is asserted and debug mode is enabled, or when debug mode is enabled and the debug non-maskable breakpoint is asserted, even if BME is cleared.

### 5.7.4.4 Bus Monitor Control Register

A diagram and description of the BMCR are provided below.

**BMCR — Bus Monitor Control Register**

**0x8007 FC48**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RESERVED				BMLK	BME	BMT		RESERVED							
RESET:															
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED															
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



**Table 5-45 BMCR Bit Settings**

Bit(s)	Name	Description
0:3	—	Reserved
4	BMLK	Bus monitor lock 0 = Enable changes to BMLK, BME, BMT 1 = Ignore writes to BMLK, BME, BMT
5	BME	Bus monitor enable 0 = Disable bus monitor 1 = Enable bus monitor
6:7	BMT	Bus monitor timing. These bits select the time-out period, in system clocks, for the bus monitor. 00 = 256 system clocks 01 = 64 system clocks 10 = 32 system clocks 11 = 16 system clocks
8:31	—	Reserved

## 5.8 Reset Operation

Reset procedures handle system initialization and recovery from catastrophic failure. The MPC509 performs reset with a combination of hardware and software. The SIU determines whether a reset is valid, asserts control signals, performs basic system configuration based on hardware mode-select inputs, and then passes control to the CPU.

Reset is the highest-priority CPU exception. Any processing in progress is aborted by the reset exception and cannot be restarted. Only essential tasks are performed during reset exception processing. Other initialization tasks must be accomplished by the exception handler routine.

### 5.8.1 Reset Sources

The following sources can cause reset:

- External reset pin ( $\overline{\text{RESET}}$ )
- Loss of oscillator
- Loss of PLL lock
- Software watchdog reset
- Checkstop reset
- JTAG reset (external  $\overline{\text{TRST}}$  pin)

All of these reset sources are fed into the reset controller. The reset status register (RSR) reflects the most recent source, or sources, of reset. (Simultaneous reset requests can cause more than one bit to be set at the same time.) This register contains one bit for each reset source. A bit set to logic one indicates the type of reset that last occurred.

Individual bits in the RSR can be cleared by writing them as zeros after reading them as ones. (Writing individual bits as ones has no effect.) The register can be read at all times. Assertion of the  $\overline{\text{RESET}}$  pin clears all bits except the RESET bit.



## RSR — Reset Status Register

0x8007 FC4C

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RE-SET	LOO	LOL	SW	CR	JTAG	RESERVED									

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED															

**Table 5-46 Reset Status Register Bit Settings**

Bit(s)	Name	Description
0	RESET	If set, source of reset is the external $\overline{\text{RESET}}$ input pin. This pin should be asserted whenever $V_{DD}$ is below $V_{DDmin}$ .
1	LOO	If set, source of reset is a loss of oscillator. The clock module asserts loss-of-oscillator reset when the MCU is in low-power mode 3 or no clock signal is present on the EXTAL pin. If the clock module detects a loss-of-oscillator condition, erroneous external bus operation will occur if synchronous external devices use the MCU input clock. Erroneous operation can also occur if devices with a PLL use the MCU CLKOUT signal. This source of reset is masked by the loss-of-oscillator reset enable (LOORE) bit in the system clock control register (SCCR).
2	LOL	If set, the cause of reset is the loss of PLL lock. The clock module asserts loss-of-lock reset when the PLL detects a loss of lock and the loss-of-lock reset enable bit is set in the system clock control register (SCCR). If the PLL detects a loss of lock condition, erroneous external bus operation will occur if synchronous external devices use the MCU input clock. Erroneous operation can also occur if devices with a PLL use the MCU CLKOUT signal. This source of reset is masked by the loss-of-lock reset enable (LOLRE) bit in the system clock control register.
3	SW	If set, source of reset is a software watchdog time-out. This occurs when the software watchdog counter reaches zero.
4	CR	If set, the source of reset is a checkstop. This occurs when the processor enters the checkstop state and the checkstop reset is enabled.
5	JTAG	If set, the source of reset is the JTAG module. This reset occurs only during production testing.
6:31	—	Reserved

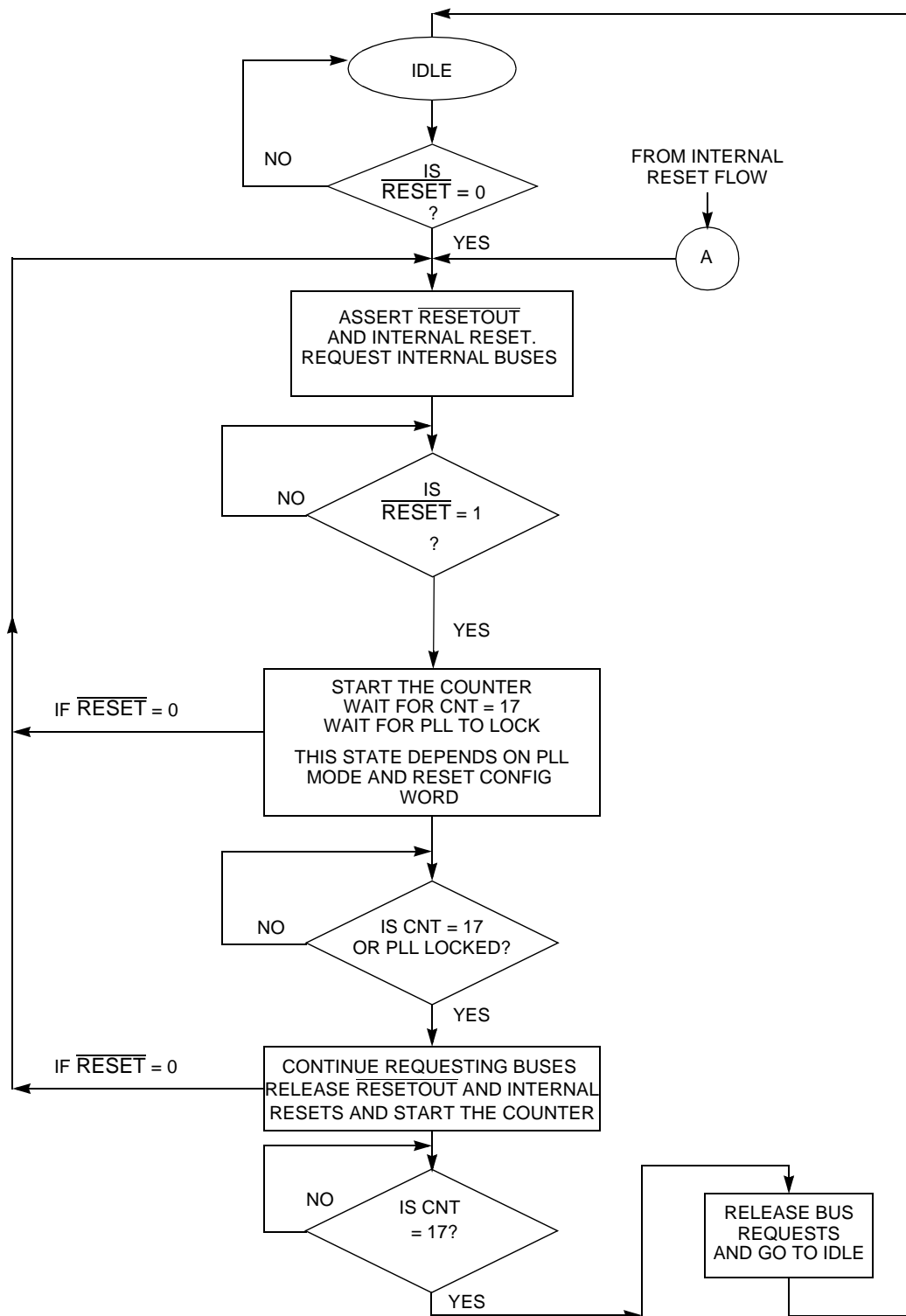
## 5.8.2 Reset Flow

The reset flow can be divided into two flows: external reset request flow and the internal reset request flow.

### 5.8.2.1 External Reset Request Flow

**Figure 5-31** is a flow diagram for external resets.





**Figure 5-31 External Reset Request Flow**

The external reset flow begins when the  $\overline{\text{RESET}}$  pin is asserted (low). The external reset request has a synchronization phase during which it takes one of the two paths

(synchronous or asynchronous) before getting to the reset control logic. The external reset request follows the asynchronous path in the case of power-on reset or in case of loss of oscillator.



Under all remaining conditions the reset request goes through the synchronous path, in which the reset request is synchronized with the system clock.  $\overline{\text{RESET}}$  must be asserted for at least two clock cycles to be recognized by the reset control block.

Once the reset request passes through the synchronization phase, the chip enters reset. The  $\overline{\text{RESETOUT}}$  pin and the internal reset signal are driven while the chip is in reset.

#### NOTE

This internal reset signal is an output from the reset control block that is sent to the internal MCU modules. This signal is different from internal reset request, which are inputs to the reset control block.

Six clock cycles after  $\overline{\text{RESET}}$  is negated, all mode select pins are sampled except for the VDDSN and MODCLK pins. These two pins are sampled at the rising edge of  $\overline{\text{RESETOUT}}$ .

After the  $\overline{\text{RESET}}$  pin is negated,  $\overline{\text{RESETOUT}}$  is held for a minimum of 17 clock cycles. After the 17 clock cycles, the state of data bus configuration bit 19 and the phase-locked loop (PLL) mode determine when  $\overline{\text{RESETOUT}}$  is released.

- If the PLL is operating in 1:1 mode or the data bus configuration bit 19 is cleared,  $\overline{\text{RESETOUT}}$  is released when the PLL is locked.
- If data bus configuration bit 19 is set and the PLL is not operating in 1:1 mode,  $\overline{\text{RESETOUT}}$  is released as soon as the 17 clock cycles have finished.

When the PLL is operating in 1:1 mode, the MCU waits until the PLL is locked before releasing  $\overline{\text{RESETOUT}}$ , since the clock which is an input to the MCU may also be used as an input to other bus devices. In addition, if other bus devices use the MCU CLKOUT signal to feed a PLL, the user must ensure that the PLL is locked before  $\overline{\text{RESETOUT}}$  is released. This is achieved by clearing data bus configuration bit 19 at reset.

While  $\overline{\text{RESETOUT}}$  is being asserted, the SIU requests control of the I-bus and L-bus. The IMB2 interface requests control of the IMB2 bus. Internal reset is released when  $\overline{\text{RESETOUT}}$  is released; however, the internal buses are not released until 16 clock cycles after  $\overline{\text{RESETOUT}}$  is negated.

If an external reset is asserted any time during this process, this process begins again.

#### 5.8.2.2 Internal Reset Request Flow

**Figure 5-32** is a flow diagram for internal reset requests.

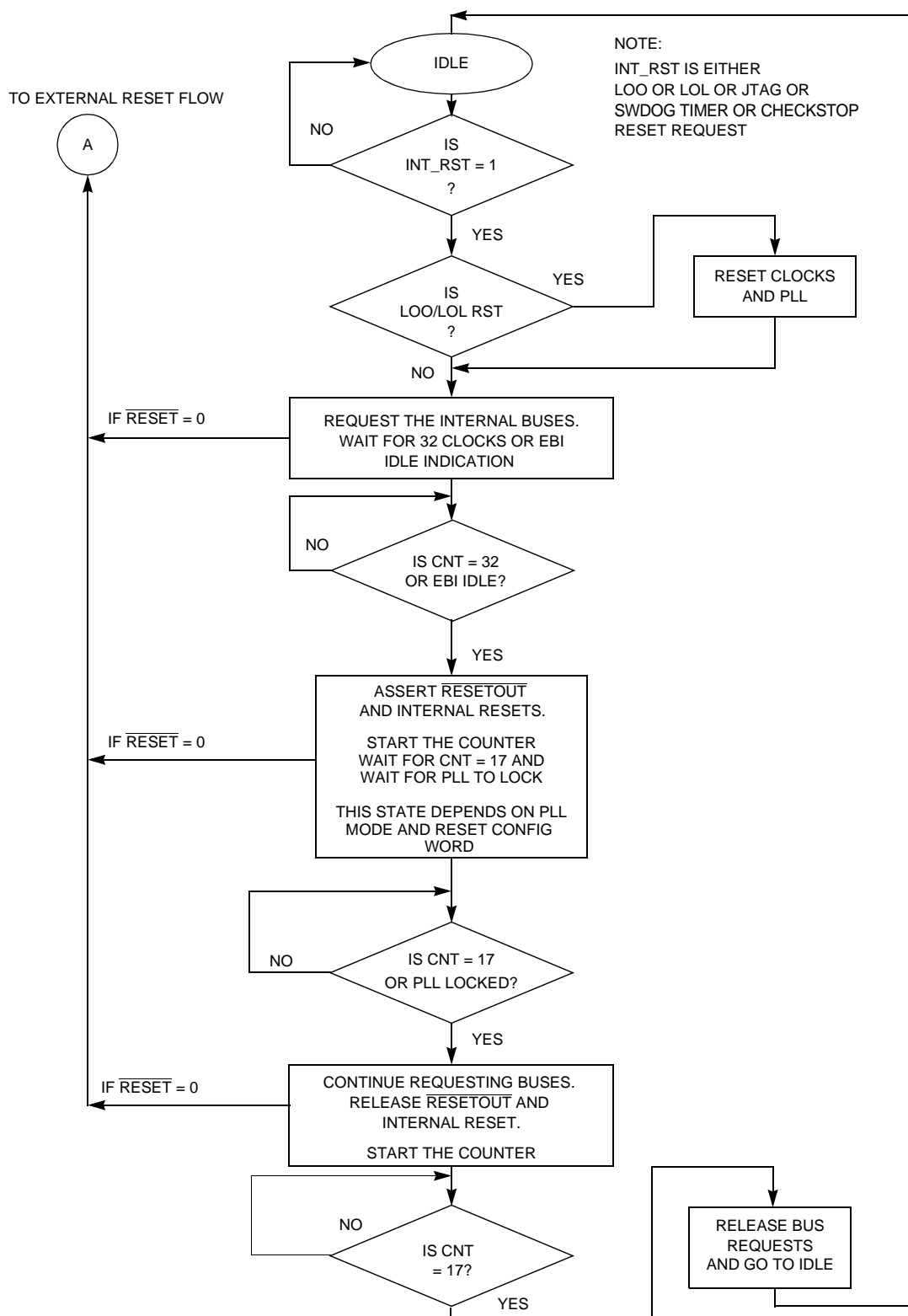


Figure 5-32 Internal Reset Request Flow



The SIU enters internal reset flow when an internal reset request is issued due to one of the following causes: loss of clock, loss of PLL lock, software watchdog time-out, entry into checkstop state, or assertion of a JTAG reset request. If the source of reset is either loss of oscillator or loss of clock, the SIU resets the clocks and the PLL immediately. For other reset sources, the SIU does not reset the clocks or the PLL.

When the internal reset request signal is asserted, the SIU attempts to complete the current transaction on the external bus before placing the chip (except clocks and PLL) in reset. The SIU requests the L-bus and I-bus and removes the qualified bus grant from the EBI to make sure that no new transaction is started.

The SIU waits for 32 clock cycles (after internal reset request is asserted) or for the EBI to indicate that the SIU is idle, whichever occurs first. Then the SIU asserts  $\overline{\text{RESETOUT}}$  and internal reset.  $\overline{\text{RESETOUT}}$  and internal reset will be driven out to put the chip into reset. Four clock cycles after the assertion of  $\overline{\text{RESETOUT}}$ , all mode select pins will be sampled except  $V_{\text{DDSN}}$ , DSCK and MODCLK pins which are sampled at the rising edge of  $\overline{\text{RESETOUT}}$ .

$\overline{\text{RESETOUT}}$  is held for a minimum of 17 clock cycles. After the 17 clock cycles, the state of data bus configuration bit 19 determines when  $\overline{\text{RESETOUT}}$  is released.

- If the PLL is operating in 1:1 mode or the data bus configuration bit 19 is cleared,  $\overline{\text{RESETOUT}}$  is released when the phase-locked loop (PLL) is locked.
- If data bus configuration bit 19 is set and the PLL is not operating in 1:1 mode,  $\overline{\text{RESETOUT}}$  is released as soon as the 17 clock cycles have finished.

Internal reset is released when  $\overline{\text{RESETOUT}}$  is released; however, the internal buses are not released until 17 clocks after  $\overline{\text{RESETOUT}}$  is negated.

If an external reset is asserted any time during this process, the external reset flow begins.

### 5.8.2.3 Reset Behavior for Different Clock Modes

**Table 5-47** summarizes the conditions under which internal reset is released for each clock mode.

**Table 5-47 Reset Behavior for Different Clock Modes**

Clock Mode	V <sub>DDSN</sub>	MODCLK	Internal DATA19 = 1 at Reset	Internal DATA19 = 0 at Reset
Normal operation	1	1	Release internal reset 17 clocks after $\overline{\text{RESET}}$ is negated	Release internal reset when PLL is locked and 17 clocks after $\overline{\text{RESET}}$ is negated OR when time-out value in the time base register has expired (whichever occurs first)
1:1 mode	1	0	Release internal reset when PLL is locked and 17 clocks after $\overline{\text{RESET}}$ is negated	
SPLL bypass mode	0	1	Release internal reset 17 clocks after $\overline{\text{RESET}}$ is negated	
Special test mode	0	0	Release internal reset 17 clocks after $\overline{\text{RESET}}$ is negated	

### 5.8.3 Configuration During Reset

Many SIU pins can have more than one function. The logic state of certain mode-select pins during reset determines which functions are assigned to pins with multiple functions. These mode-select pins determine other aspects of operating configuration as well.

Basic operating configuration is determined by the DSDI and DSCK pins, as shown in [Table 5-48](#).

**Table 5-48 Pin Configuration During Reset**

Pin During Reset	Function Affected
DSCK asserted (1)	Debug mode enabled
DSCK negated (0)	Debug mode disabled
DSDI asserted (1)	Data bus configuration mode
DSDI negated (0)	Internal default mode

The state of the DSDI pin is latched internally five clock cycles after  $\overline{\text{RESETOUT}}$  is asserted. The state of the DSCK pin is latched every clock cycle while  $\overline{\text{RESETOUT}}$  is asserted. The MCU is configured based on the values latched from these two pins. The user is responsible for ensuring a valid level on these pins five clock cycles after  $\overline{\text{RESETOUT}}$  is asserted. If DSDI is asserted (causing data bus configuration mode to be entered), the user must also drive DATA[0:5], at a minimum.

For any reset source other than external reset, the external data pins are latched five clock cycles after internal reset control logic asserts  $\overline{\text{RESETOUT}}$ . For external resets, the data pins are latched five clock cycles after  $\overline{\text{RESET}}$  is negated. The default reset configuration word is driven onto the internal buses until the external word is latched. If no external reset configuration word is latched, the default word continues to be driven on the internal buses.

This scheme allows users of the internal default mode to limit their required external configuration hardware to two pull-down resistors (DSDI and DSCK). It also allows many options to be configured with a single three-state octal buffer.



### 5.8.3.1 Data Bus Configuration Mode

If data bus configuration mode is selected (DSDI asserted), then the MCU is configured according to the values latched from the data bus pins.

#### NOTE

$\overline{BG}$  must be asserted in order for the reset configuration word latch to access the data bus.

The external data bus is divided into four groups:

- DSDI, DATA[0:5]
- DATA[6:13]
- DATA[14:21]
- DATA[22:31]

This grouping allows the user to use three-state octal buffers to only drive valid data on the pins for those reset configuration options that the user would want to change. The state of the last pin in each group (pins 5, 13, and 21) determines whether the next set of configuration options use the internal default values or are configured from the external data bus. The user is required to drive to a valid level all the pins in any of the groups that are to be changed. The functions selected by these pins are shown in [Table 5-49](#).

### 5.8.3.2 Internal Default Mode

If DSDI is held low during reset, internal default mode is selected. The internal default mode allows MCUs with on-board non-volatile memory modules (such as flash EEPROM) to provide a pin configuration word on the instruction or load/store data bus during reset. For MCUs without such a memory module, such as the MPC509, the SIU provides a mask-programmed default value.

### 5.8.3.3 Data Bus Reset Configuration Word

In either reset configuration mode (data bus configuration mode or internal default mode), the configuration is accomplished within the MCU by driving a configuration word on the internal data bus before the internal  $\overline{RESET}$  signal is negated. At the negation of internal  $\overline{RESET}$ , those functions that are configured at reset latch their configuration values from the assigned bits of the internal data bus. The format of the data bus reset configuration word is the same regardless of which configuration mode is selected, except that data bus bits 5, 13, and 21 have no meaning in internal default mode. [Table 5-49](#) describes the configuration options.



**Table 5-49 Data Bus Reset Configuration Word**

Data Bus Bit	Configuration Function Affected	Effect of Mode Select = 1 During Reset	Effect of Mode Select = 0 During Reset	Internal Default Mode	
				3 V I/O <sup>1</sup>	TTL I/O <sup>2</sup>
0	Address Bus	Minimum bus mode ADDR[0:11]/CS[0:11] configured as chip selects	Maximum bus mode ADDR[0:11]/CS[0:11] configured as address pins	1	1
1	Vector table location (IP Bit)	Vector table 0xFFFF 0000	Vector table 0x0000 0000	0	1
2	Burst type/indication	Type 2 (LAST Timing)	Type 1 (BDIP Timing)	0	0
3	Interface type for CSBOOT	ITYPE = 001 Asynchronous (Time to Hi-Z = 2 Clk)	ITYPE = 1000 Synchronous burst	1	1
4	CSBOOT port size	32-Bit	16-Bit	1	0
5	Reset configuration source For DATA[6:13]	Latch configuration from external pins	Latch configuration from internal defaults	0	0
6:8	TA Delay For CSBOOT	TA Delay Encoding 000 001 010 011 100 101 110 111	Number of Wait States 0 1 2 3 4 5 6 7	010	111
9:10	IMEMBASE[0:1]	IMEMBASE 00 01 10 11	I-Mem block placement Start Addr: 0x0000 0000 End Addr: 0x000F FFFF Start Addr: 0xFFFF 0000 End Addr: 0xFFFF FFFF Note: MPC509 does not contain I-Mem	01	10
11:12	LMEMBASE[0:1]	LMEMBASE 00 01 10 11	L-Mem (SRAM) block placement Start address: 0x0000 0000 End address: 0x000F FFFF Start address: 0xFFFF 0000 End address: 0xFFFF FFFF	11	11
13	Reset configuration source for DATA[14:21]	Latch configuration from external pins	Latch configuration from internal defaults	0	0
14	CT[0:3], AT[0:1], TS	CT[0:3], AT[0:1], TS	PJ[1:7]	1	1
15	WR, BDIP	WR, BDIP	PK[0:1]	1	1
16	PLLL/DSDO, VF[0:2], VFLS[0:1], WP[1:5]	DSDO, pipe tracking, watchpoints	PK[2:7], PL[2:7]	1	1

**Table 5-49 Data Bus Reset Configuration Word (Continued)**



Data Bus Bit	Configuration Function Affected	Effect of Mode Select = 1 During Reset	Effect of Mode Select = 0 During Reset	Internal Default Mode	
				3 V I/O <sup>1</sup>	TTL I/O <sup>2</sup>
17	BURST, TEA, AACK, TA, BE[0:3]	Handshake pins	PORTI[0:7]	1	1
18	CR, BI, BR, BB, BG, ARETRY	Bus arbitration pins	PM[2:7]	1	1
19	Release reset when PLL locked	Refer to <a href="#">Table 5-47</a>	Refer to <a href="#">Table 5-47</a>	0	1
20	Reserved			0	0
21	Reset configuration source For DATA[22:31]	Latch configuration from external pins	Latch configuration from internal defaults	0	0
22	Reserved			0	0
23	Reserved			0	0
24	LEN	L-bus memory modules are enabled.	L-bus memory modules are disabled and emulated externally	1	1
25	PRUMODE	Forces accesses to ports A, B, I, J, K, and L to go external	No effect	0	0
26	ADDR[12:15]	ADDR[12:15]	PB[4:7]	1	1
27	Reserved			0	0
28	Reserved			0	0
29	Reserved			0	0
30	Test slave mode enable	Test slave mode disabled	Test slave mode enabled	1	1
31	Test transparent mode enable	Test transparent mode disabled	Test transparent mode enabled	1	1

**NOTES:**

1. 3 V only I/O structure. The part number is MPC509LCFT25.
2. TTL compatible (5 V friendly input) I/O structure.

### 5.8.4 Power-On Reset

Power-on reset occurs when the VDDKAP1 pin is high and  $V_{DD}$  makes a transition from zero to one. The SIU does not have a power-on reset circuit. This function must be provided externally.

**NOTE**

The ADDR[0:11]/CS[0:11] pins contain unknown values during the first two clocks of power-on reset. The user must ensure that external EEPROM and standby memory interface logic do not allow inadvertent writes during this time.



## 5.9 General-Purpose I/O



Many of the pins associated with the SIU can be used for more than one function. The primary function of these pins is to provide an external bus interface. When not used for their primary function, many of these pins can be used as digital I/O pins.

SIU digital I/O pins are grouped into eight-bit ports. The following registers are associated with each I/O port. (Output-only ports do not have a data direction register.)

- Pin assignment register — allows the user to configure a pin for its primary function or digital I/O.
- Data direction register — configures individual pins as input or output pins.
- Data register — monitors or controls the state of its pins, depending on the state of the data direction register for that pin.

If a pin is not configured as an I/O port pin in the pin assignment register, the data direction and data registers have no effect on the pin.

Ports A through L can be used with a port replacement unit (PRU). These ports provide three-clock-cycle access. If PRU mode is enabled at reset, access to these registers is disabled, and an external bus cycle is initiated. Other (non-PRU) ports provide two-clock-cycle access.

In addition to the SIU ports described in this section, port Q in the peripherals controller unit provides edge- or level-sensitive I/O. Refer to [6.6 Port Q](#) for information on port Q.

[Table 5-50](#) is an address map of the SIU port registers.



**Table 5-50 SIU Port Registers Address Map**

Access	Address	Register
S	0x8007 FC60	Port M data direction (DDRM)
S	0x8007 FC64	Port M pin assignment (PMPAR)
S/U	0x8007 FC68	Port M data (PORTM)
—	0x8007 FC6C – 0x8007 FC80	Reserved
S	0x8007 FC84	Port A, B pin assignment (PAPAR, PBPAP)
S/U	0x8007 FC88	Port A, B data (PORTA, PORTB)
—	0x8007 FC8C – 0x8007 FC94	Reserved
S	0x8007 FC98	Port I, J, K, L data direction (DDRI, DDRJ, DDRK, DDRL)
S	0x8007 FC9C	Port I, J, K, L pin assignment (PIPAR, PJPAR, PKPAR, PLPAR)
S/U	0x8007 FCA0	Port I, J, K, L data (PORTI, PORTJ, PORTK, PORTL)
—	0x8007 FCA4 – 0x8007 FCFF	Reserved

### 5.9.1 Port Timing

Ports A through L can be used with a port replacement unit (PRU). These ports provide three-clock-cycle access. If PRU mode is enabled at reset, access to these registers is disabled, and an external bus cycle is initiated. Other (non-PRU) ports provide two-clock-cycle access. Input port pins are sampled synchronously.

After a pin assignment or data direction register is modified, the change may require an additional clock cycle to take effect at the pin.

Note that the timing of output port pins does not match the timing of the corresponding bus control pins.

## 5.9.2 Port M



### PORTM — Port M Data Register

0x8007 FC68

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	PM3	PM4	PM5	PM6	PM7	RESERVED							

RESET:

0 0 0 U U U U 0 0 0 0 0 0 0 0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED															

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

U = Unaffected by reset

Writes to PORTM are stored in internal data latches. If any bit of the port is configured as an output, the value latched for that bit is driven onto the pin. A read of PORTM returns the value at the pin only if the pin is configured as a discrete input. Otherwise, the value read will be the value stored in the internal data latch. PORTM can be read or written at any time. This register is unaffected by reset.

### DDRM — Port M Data Direction Register

0x8007 FC60

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	DDM3	DDM4	DDM5	DDM6	DDM7	RESERVED							

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED															

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

The bits in this register control the direction of the port M pin drivers when the pins are configured as I/O pins. Setting a bit in this register to one configures the corresponding pin as an output; clearing the bit configures the pin as an input.

## PMPAR — Port M Pin Assignment Register

0x8007 FC64



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	PMPA 2	PMPA 3	PMPA 4	PMPA 5	PMPA 6	PMPA 7	RESERVED							

RESET:

0 0 \* \* \* \* \* 0 0 0 0 0 0 0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED															

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

\* Reset setting depends on the value of the configuration word at reset.

The bits in this register control the function of the associated pins. Setting a bit in this register to one configures the corresponding pin as a bus control signal; clearing a bit configures the pin as an I/O pin (or as the  $\overline{DS}$  signal, in the case of PMPA2).

**Table 5-51 Port M Pin Assignments**

PMPAR Bit	Port M Signal	Bus Control Signal
PMPA2	$\overline{DS}$	$\overline{CR}$
PMPA3	PM3	$\overline{BI}$
PMPA4	PM4	$\overline{BR}$
PMPA5	PM5	$\overline{BB}$
PMPA6	PM6	$\overline{BG}$
PMPA7	PM7	$\overline{ARETRY}$

### 5.9.3 Ports A and B

Ports A and B are 8-bit output ports. Associated with each port is a data register and a pin assignment register; data direction registers are not needed.

## PORTA, PORTB — Port A, B Data Registers

0x8007 FC88



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7	PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7

RESET:

U    U    U    U    U    U    U    U    U    U    U    U    U    U    U    U

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED															

RESET:

0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

U = Unaffected by reset

When a port A or port B pin is configured as a general-purpose output, the value in the port A or port B data register is driven onto the pin. PORTA and PORTB are unaffected by reset.

## PAPAR, PBPAPAR — Port A, B Pin Assignment Register

0x8007 FC84

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PAPA 0	PAPA 1	PAPA 2	PAPA 3	PAPA 4	PAPA 5	PAPA 6	PAPA 7	PBPA 0	PBPA 1	PBPA 2	PBPA 3	PBPA 4	PBPA 5	PBPA 6	PBPA 7

RESET:

1    1    1    1    1    1    1    1    1    1    1    1    \*    \*    \*    \*

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED															

RESET:

0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

\* Reset setting depends on the value of the configuration word at reset.

Each bit in this register controls the function of the associated pin, provided the pin is configured for non-chip-select function in the corresponding chip-select options register. Setting a bit in the PAPAR or PBPAPAR configures the corresponding pin as an address bus pin; clearing the bit configures the pin as an I/O pin.



**Table 5-52 Port A Pin Assignments**

PMPAR Bit	Port A Signal	Bus Control Signal
PAPA0	PA0	ADDR0
PAPA1	PA1	ADDR1
PAPA2	PA2	ADDR2
PAPA3	PA3	ADDR3
PAPA4	PA4	ADDR4
PAPA5	PA5	ADDR5
PAPA6	PA6	ADDR6
PAPA7	PA7	ADDR7

**Table 5-53 Port B Pin Assignments**

PMPAR Bit	Port B Signal	Bus Control Signal
PBPA0	PB0	ADDR8
PBPA1	PB1	ADDR9
PBPA2	PB2	ADDR10
PBPA3	PB3	ADDR11
PBPA4	PB4	ADDR12
PBPA5	PB5	ADDR13
PBPA6	PB6	ADDR14
PBPA7	PB7	ADDR15

## 5.9.4 Ports I, J, K, and L

**PORTI, PORTJ, PORTK, PORTL — Port I, J, K, L Data Registers**

**0x8007 FCA0**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PI0	PI1	PI2	PI3	PI4	PI5	PI6	PI7	0	PJ1	PJ2	PJ3	PJ4	PJ5	PJ6	PJ7
RESET:															
U	U	U	U	U	U	U	U	0	U	U	U	U	U	U	U

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
PK0	PK1	PK2	PK3	PK4	PK5	PK6	PK7	0	0	PL2	PL3	PL4	PL5	PL6	PL7
RESET:															
U	U	U	U	U	U	U	U	0	0	U	U	U	U	U	U

U = Unaffected by reset

Writes to port I, J, K, and L data registers are stored in internal data latches. If any pin in one of these ports is configured as an output, the value latched for the corresponding data register bit is driven onto the pin. A read of one of these data registers returns the value at the pin only if the pin is configured as a discrete input. Otherwise, the value read is the value stored in the internal data latch. Port I, J, K, and L data registers can be read at any time. These registers are unaffected by reset.



### DDRI, DDRJ, DDRK, DDRL — Port I, J, K, L Data Direction Registers 0x8007 FC98

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DDI0	DDI1	DDI2	DDI3	DDI4	DDI5	DDI6	DDI7	0	DDJ1	DDJ2	DDJ3	DDJ4	DDJ5	DDJ6	DDJ7

RESET:

0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
DDK0	DDK1	DDK2	DDK3	DDK4	DDK5	DDK6	DDK7	0	0	DDL2	DDL3	DDL4	DDL5	DDL6	DDL7

RESET:

0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

The bits in these registers control the direction of the associated pin drivers when the pins are configured as I/O pins. Setting a bit in these registers configures the corresponding pin as an output; clearing the bit configures the pin as an input.

### PIPAR, PJPAR, PKPAR, PLPAR — Port I, J, K, L Pin Assignment Registers 0x8007 FC9C

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PIPA0	PIPA1	PIPA2	PIPA3	PIPA4	PIPA5	PIPA6	PIPA7	0	PJPA1	PJPA2	PJPA3	PJPA4	PJPA5	PJPA6	PJPA7

RESET:

\*   \*   \*   \*   \*   \*   \*   \*   0   \*   \*   \*   \*   \*   \*   \*

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
PKPA 0	PKPA 1	PKPA 2	PKPA 3	PKPA 4	PKPA 5	PKPA 6	PKPA 7	0	0	PLPA2	PLPA3	PLPA4	PLPA5	PLPA6	PLPA7

RESET:

\*   \*   \*   \*   \*   \*   \*   \*   0   0   \*   \*   \*   \*   \*   \*

\* Reset setting depends on the value of the configuration word at reset.

The bits in these registers control the function of the associated pins. Setting a bit configures the corresponding pin as a bus control signal; clearing the bit configures the pin as an I/O pin.

**Table 5-54 Port I Pin Assignments**

PIPAR Bit	Port I Signal	Bus Control Signal
PIPA0	PI0	$\overline{\text{BURST}}$
PIPA1	PI1	$\overline{\text{TEA}}$
PIPA2	PI2	$\overline{\text{AACK}}$
PIPA3	PI3	$\overline{\text{TA}}$
PIPA4	PI4	$\overline{\text{BE0}}$
PIPA5	PI5	$\overline{\text{BE1}}$
PIPA6	PI6	$\overline{\text{BE2}}$
PIPA7	PI7	$\overline{\text{BE3}}$

**Table 5-55 Port J Pin Assignments**

PJPAR Bit	Port J Signal	Bus Control Signal
PJPA1	PJ1	AT0
PJPA2	PJ2	AT1
PJPA3	PJ3	$\overline{\text{TS}}$
PJPA4	PJ4	CT0
PJPA5	PJ5	CT1
PJPA6	PJ6	CT2
PJPA7	PJ7	CT3

**Table 5-56 Port K Pin Assignments**

PKPAR Bit	Port K Signal	Bus Control Signal
PKPA0	PK0	$\overline{\text{BDIP}}$
PKPA1	PK1	$\overline{\text{WR}}$
PKPA2	PK2	PLLL/DSDO
PKPA3	PK3	VF0
PKPA4	PK4	VF1
PKPA5	PK5	VF2
PKPA6	PK6	VFLS0
PKPA7	PK7	VFLS1





**Table 5-57 Port L Pin Assignments**

PLPAR Bit	Port L Signal	Bus Control Signal
PLPA2	PL2	WP0
PLPA3	PL3	WP1
PLPA4	PL4	WP2
PLPA5	PL5	WP3
PLPA6	PL6	WP4
PLPA7	PL7	WP5

### **5.9.5 Port Replacement Unit (PRU) Mode**

The entire external bus interface must be supported in order to build an emulator for an MCU. The SIU contains support for external port replacement logic which can be used to faithfully replicate on-chip ports externally. This PRU mode allows system development of a single-chip application in expanded mode. Access (including access time) to the port replacement logic can be made transparent to the application software.

In PRU mode, all data, data direction, and pin assignment registers for ports A, B, I, J, K, and L are mapped externally. The SIU does not respond to these accesses, allowing external logic, such as a PRU, to respond.

PRU mode is invoked by pulling DATA25 high during reset. Other pins should be configured as bus control pins.

