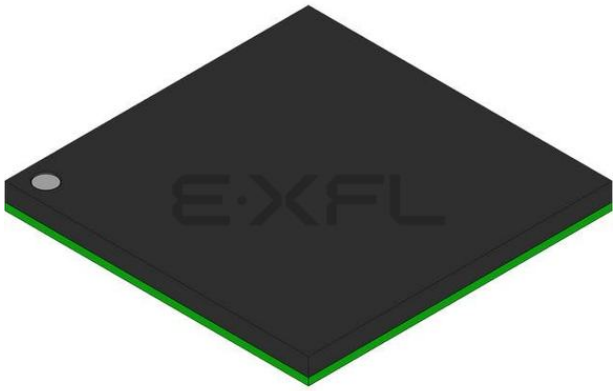

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	Coldfire V2
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	EBI/EMI, Ethernet, I²C, SPI, UART/USART, USB
Peripherals	DMA, WDT
Number of I/O	32
Program Memory Size	16KB (4K x 32)
Program Memory Type	ROM
EEPROM Size	-
RAM Size	1K x 32
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	196-LBGA
Supplier Device Package	196-MAPBGA (15x15)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mcf5272vm66j

Table of Contents (Continued)

2.1.1.2.3 Hardware Divide Unit

The hardware divide unit performs the following integer division operations:

- 32-bit operand/16-bit operand producing a 16-bit quotient and a 16-bit remainder
- 32-bit operand/32-bit operand producing a 32-bit quotient
- 32-bit operand/32-bit operand producing a 32-bit remainder

2.1.2 Debug Module Enhancements

The ColdFire processor core debug interface supports system integration in conjunction with low-cost development tools. Real-time trace and debug information can be accessed through a standard interface, which allows the processor and system to be debugged at full speed without costly in-circuit emulators. On-chip breakpoint resources include the following:

- Configuration/status register (CSR)
- Bus attributes and mask register (AATR)
- Breakpoint registers. These can be used to define triggers combining address, data, and PC conditions in single- or dual-level definitions. They include the following:
  - PC breakpoint register (PBR)
  - PC breakpoint mask register (PBMR)
  - Data operand address breakpoint registers (ABHR/ABLR)
  - Data breakpoint register (DBR)
- Data breakpoint mask register (DBMR)
- Trigger definition register (TDR) can be programmed to generate a processor halt or initiate a debug interrupt exception

These registers can be accessed through the dedicated debug serial communication channel, or from the processor's supervisor programming model, using the WDEBUG instruction.

2.2 Programming Model

The MCF5272 programming model consists of three instruction and register groups—user, MAC (also user-mode), and supervisor, shown in Figure 2-2. User mode programs are restricted to user and MAC instructions and programming models. Supervisor-mode system software can reference all user-mode and MAC instructions and registers and additional supervisor instructions and control registers. The user or supervisor programming model is selected based on SR[S]. The following sections describe the registers in the user, MAC, and supervisor programming models.

2.2.1 User Programming Model

As Figure 2-3 shows, the user programming model consists of the following registers:

- 16 general-purpose 32-bit registers, D0–D7 and A0–A7
- 32-bit program counter
- 8-bit condition code register

2.2.1.6 MAC Programming Model

Figure 2-3 shows the registers in the MAC portion of the user programming model. These registers are described as follows:

- Accumulator (ACC)—This 32-bit, read/write, general-purpose register is used to accumulate the results of MAC operations.
- Mask register (MASK)—This 16-bit general-purpose register provides an optional address mask for MAC instructions that fetch operands from memory. It is useful in the implementation of circular queues in operand memory.
- MAC status register (MACSR)—This 8-bit register defines configuration of the MAC unit and contains indicator flags affected by MAC instructions. Unless noted otherwise, MACSR indicator flag settings are based on the final result, that is, the result of the final operation involving the product and accumulator.

2.2.2 Supervisor Programming Model

The MCF5272 supervisor programming model is shown in Figure 2-3. Typically, system programmers use the supervisor programming model to implement operating system functions and provide memory and I/O control. The supervisor programming model provides access to the user registers and additional supervisor registers, which include the upper byte of the status register (SR), the vector base register (VBR), and registers for configuring attributes of the address space connected to the Version 2 processor core. Most supervisor-mode registers are accessed by using the MOVEC instruction with the control register definitions in Table 2-2.

Table 2-2. MOVEC Register Map

Rc[11–0]	Register Definition
0x002	Cache control register (CACR)
0x004	Access control register 0 (ACR0)
0x005	Access control register 1 (ACR1)
0x801	Vector base register (VBR)
0xC00	ROM base address register
0xC04	RAM base address register (RAMBAR)
0xC0F	Module base address register (MBAR)

Table 4-8. CACR Field Descriptions (continued)

Bits	Name	Description
1–0	CLNF	Control longword fetch. Controls the size of the memory request the cache issues to the bus controller for different initial line access offsets.

CLNF	Longword Address Bits			
	00	01	10	11
00	Line	Line	Line	Longword
01	Line	Line	Longword	Longword
1x	Line	Line	Line	Line

4.5.3.2 Access Control Registers (ACR0 and ACR1)

The ACRs define memory reference attributes for two memory regions (one per ACR). These attributes affect every memory reference using the ACRs or the set of default attributes contained in the CACR. ACRs are examined for each memory reference not mapped to the SRAM or ROM module. The supervisor-level ACRs are accessed in the CPU address space using the MOVEC instruction with an Rc encoding of 0x004 and 0x005. ACRs can be read and written in BDM mode.

	31–24	23–16	15	14–13	12–7	6	5	4	3	2	1	0
Field	BA	BAM	EN	SM	—	CM	BWE	—		WP	—	
Reset	0000_0000_0000_0000											
R/W	Write (R/W by debug module)											
Rc	ACR0: 0x004; ACR1: 0x005											

Figure 4-5. Access Control Register Format (ACR*n*)

Table 4-9 describes ACR*n* fields.

Table 4-9. ACR*n* Field Descriptions

Bits	Name	Description
31–24	BA	Base address. Compared with A[31:24]. Eligible addresses that match are assigned the access control attributes of this register.
23–16	BAM	Base address mask. Setting a BAM bit masks the corresponding BA bit. Setting low-order BAM bits can define contiguous regions exceeding 16 Mbytes. BAM can define multiple noncontiguous regions.
15	EN	Enable. Enables or disables the other ACR*n* bits. 0 Access control attributes disabled 1 Access control attributes enabled
14–13	SM	Supervisor mode. Specifies whether only user or supervisor accesses are allowed in this address range or if the type of access is a don't care. 00 Match addresses only in user mode 01 Match addresses only in supervisor mode 1x Execute cache matching on all accesses
12–7	—	Reserved; should be cleared.

Chapter 2, “ColdFire Core.” Pending interrupts from external sources ($\overline{\text{INT}}$[6:1]) can be cleared using the ICRs.

For an interrupt to be successfully processed, stack RAM must be available. A programmable chip select is often used for the RAM, in which case, the RAM is not immediately available at startup. Thus, no interrupts are recognized until PIVR is initialized. The RAM chip select and system stack should be set up before this initialization.

If more than one interrupt source has the same interrupt priority level (IPL), the interrupt controller daisy chains the interrupts with the priority order following the bit placement in the PIWR, with $\overline{\text{INT1}}$ having the highest priority and SWTO having the lowest priority, as shown in Figure 7-8.

7.2.1 Interrupt Controller Registers

This section describes the registers associated with the interrupt controller. Table 7-2 gives the nomenclature used for the interrupt and power management registers.

Table 7-2. Interrupt and Power Management Register Mnemonics

Mnemonic or Portion Thereof	Description
INT1, INT2, INT3, INT4, INT5, INT6	External interrupt signals 1–6.
TMR0, TMR1, TMR2, TMR3	Timers 3–0 from timer module
USB0, USB1, USB2, USB3, USB4, USB5, USB6, USB7	USB endpoint 0–7
UART1, UART2	UART1, UART2 modules
PLIP	PLIC 2-KHz periodic interrupt, 2B+D data
PLIA	PLIC asynchronous and maintenance channels interrupt
DMA	DMA controller interrupt
ETx	Ethernet module transmit data interrupt
ERx	Ethernet module receive data interrupt
ENTC	Ethernet module non-time-critical interrupt
QSPI	Queued serial peripheral interface
IPL2, IPL1, IPL0	Interrupt priority level bits 2–0
PI	Pending interrupt
PDN	Power down enable
WK	Wakeup enable
SWTO	Software watchdog timer time out

11.5.21 Receive Buffer Size Register (EMRBR)

The EMRBR register dictates the maximum size of all receive buffers. Note that because receive frames are truncated at 2k-1 bytes, only bits 10–4 are used. This value should take into consideration that the receive CRC is always written into the last receive buffer. To allow one maximum size frame per buffer, R_BUFF_SIZE must be set to MAX_FL or larger. The R_BUFF_SIZE must be evenly divisible by 16. To insure this, bits 3–0 are forced low. To minimize bus utilization (descriptor fetches), it is recommended that R_BUFF_SIZE be at least 256 bytes.

This register, Figure 11-26, is not reset and must be initialized by the user prior to operation.

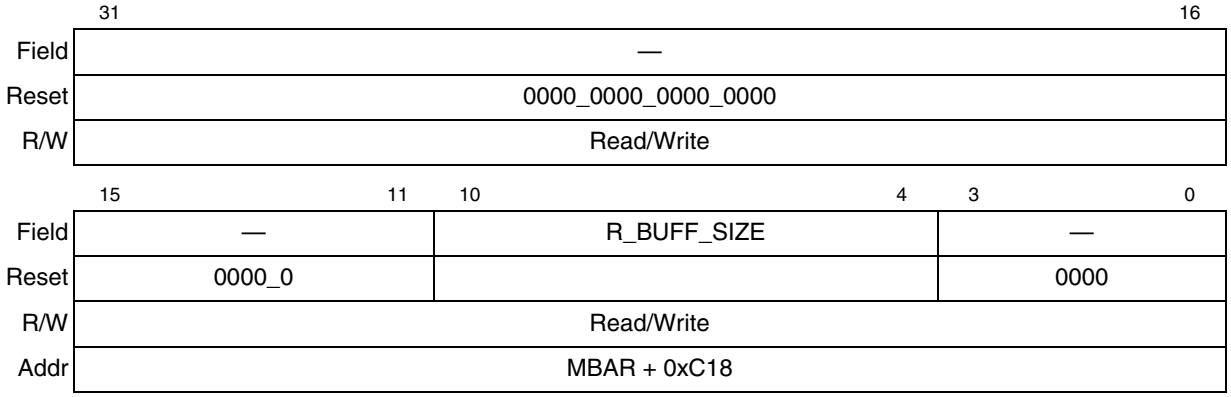

Figure 11-26. Receive Buffer Size (EMRBR)

Table 11-29. EMRBR Field Descriptions

Bits	Name	Description
31–11	—	Reserved, should be cleared.
10–4	R_BUFF_SIZE	Receive buffer size.
3–0	—	Reserved, should be cleared.

12.3.2.15 USB Endpoint 0 Interrupt Mask (EP0IMR) and General/Endpoint 0 Interrupt Registers (EP0ISR)

Figure 12-18 shows the USB endpoint 0 interrupt mask and general/endpoint 0 interrupt registers.

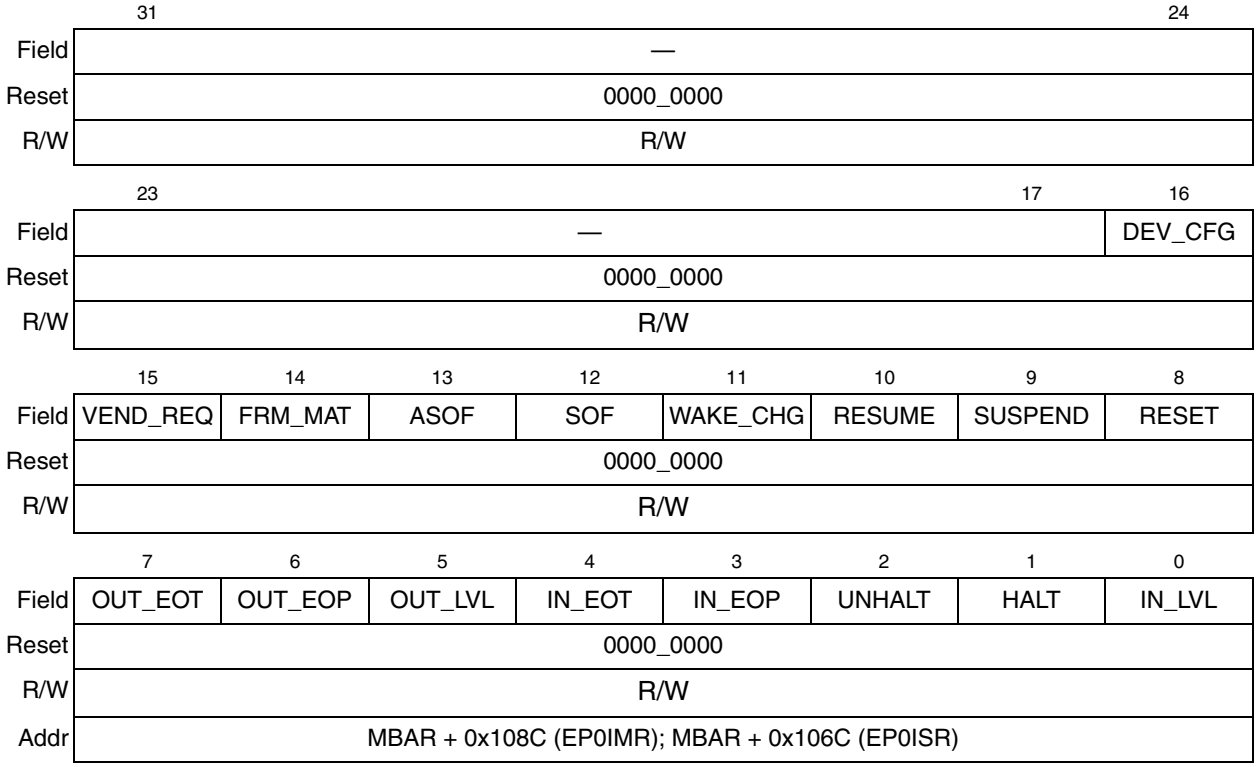

Figure 12-18. USB Endpoint 0 Interrupt Mask (EP0IMR) and General/Endpoint 0 Interrupt Registers (EP0ISR)

NOTE

Interrupt bits are reset by writing a 1 to the specified bits. Writing 0 has no effect.

Table 12-14 lists field descriptions for the USB endpoint 0 interrupt mask and general/endpoint 0 interrupt registers.

Table 12-14. EP0IMR and EP0ISR Field Descriptions

Bits	Name	Description
31–17	—	Reserved, should be cleared.
16	DEV_CFG	Device configuration change interrupt. Set when a device configuration change has been received. The USB standard device requests SET_CONFIGURATION and SET_INTERFACE generate a DEV_CFG interrupt. Any IN or OUT packets to the active endpoints cause a NAK response to the host while this bit is set in order to allow the user to initialize the endpoints' FIFO's. Note that if one of these requests is done repeatedly and therefore the registers don't change, a DEV_CFG interrupt is still generated. If debug mode is enabled, a change in FAR also generates an interrupt. 0 No interrupt pending 1 Device configuration change received

13.5.2 B2 Data Receive Registers (P0B2RR–P3B2RR)

All bits in these registers are read only and are set on hardware or software reset.

The P*n*B2RR registers contain the last four frames of data received on channel B2. (P0B2RR is the B2 channel data for port 0, P1B2RR is B2 for port 1, and so on.) The data are packed from LSB to MSB.

These registers are aligned on long-word boundaries from MBAR + 0x310 for P0B2RR to MBAR + 0x31C for P3B2RR. See Section 13.2.3, “GCI/IDL B- and D-Channel Bit Alignment,” for the frame and bit alignment within the 32-bit word.

Figure 13-14 shows the B2 receive data registers.

	31 – 24	23 – 16
Field	Frame 0	Frame 1
Reset	1111_1111	1111_1111
R/W	Read Only	

	15 – 8	7 – 0
Field	Frame 2	Frame 3
Reset	1111_1111	1111_1111
R/W	Read Only	
Addr	MBAR + 0x310 (P0B2RR); 0x314 (P1B2RR); 0x318 (P2B2RR); 0x31C (P3B2RR)	

Figure 13-14. B2 Receive Data Registers P0B2RR – P3B2RR

13.5.3 D Data Receive Registers (P0DRR–P3DRR)

All bits in these registers are read-only and are set on hardware or software reset.

The P*n*DRR registers contain the last four frames of D-channel receive data packed from the least significant bit, (lsb), to the most significant bit, (msb), for each of the four physical ports on the MCF5272. P0DRR is the D-channel byte for port 0, P1DRR the D channel for port 1, and so on.

Each of the four byte-addressable registers, P0DRR-P3DRR, are packed to form one 32-bit register, P*n*DRR, located at MBAR + 0x320. P0DRR is located in the MSB of the P*n*DRR register, P3DRR is located in the LSB of the P*n*DRR register.

	31 – 24	23 – 16
Field	P0DRR	P1DRR
Reset	1111_1111	1111_1111
R/W	Read Only	

	15 – 8	7 – 0
Field	P2DRR	P3DRR
Reset	1111_1111	1111_1111
R/W	Read Only	
Addr	MBAR + 0x320 (P0DRR); 0x321 (P1DRR); 0x322 (P2DRR); 0x323 (P3DRR)	

Figure 13-15. D Receive Data Registers P0DRR–P3DRR

13.5.10 Periodic Status Registers (P0PSR–P3PSR)

All bits in these registers are read only and are set on hardware or software reset.

	15–12	11	10	9	8	7	6	5	4	3	2	1	0
P0PSR–3	—	DTUE	B2TUE	B1TUE	DROE	B2ROE	B1ROE	DTDE	B2TDE	B1TDE	DRDF	B2RDE	B1RDF
Reset	0000_0000_0000_0000												
R/W	Read Only												
Addr	MBAR + 0x384 (P0PSR); 0x386 (P1PSR); 0x388 (P2PSR); 0x38A (P3PSR)												

Figure 13-22. Periodic Status Registers (P0PSR–P3PSR)

P*n*PSR are 16-bit registers containing the interrupt status information for the B- and D-channel transmit and receive registers for each of the four ports on the MCF5272.

Table 13-5. P0PSR–P3PSR Field Descriptions

Bits	Name	Description
15–12	—	Reserved, should be cleared.
11	DTUE	D data transmit underrun error. This bit is set when the data in the PLTD transmit data register for the respective port was transferred to the transmit shadow register, which was already empty indicated by DTDE. DTUE is automatically cleared, when the P*n*PSR register has been read by the CPU.
10	B2TUE	B2 data transmit underrun error. This bit is set when the data in the P*n*B2TR transmit data register for the respective port was transferred to the transmit shadow register, which was already empty indicated by B2TDE. B2TUE is automatically cleared when the P*n*PSR register has been read by the CPU.
9	B1TUE	B1 data transmit underrun error. This bit is set when the data in the P*n*B1TR transmit data register for the respective port was transferred to the transmit shadow register, which was already empty indicated by B1TDE. B1TUE is automatically cleared when the P*n*PSR register has been read by the CPU.
8	DROE	D-Channel data receive overrun error. This bit is set when the data in the D receive shadow register for the respective port has been transferred to the receive data register P*n*DRR, which was already full indicated by DRDF. DROE is automatically cleared when the P*n*PSR register has been read by the CPU.
7	B2ROE	B2 data receive overrun error. This bit is set when the data in the B2 receive shadow register for the respective port has been transferred to the receive data register P*n*B2RR, which was already full indicated by B2RDF. B2ROE is automatically cleared when the P*n*PSR register has been read by the CPU.
6	B1ROE	B1 data receive overrun error. This bit is set when the data in the B1 receive shadow register for the respective port has been transferred to the receive data register P*n*B1RR, which was already full indicated by B1RDF. B1ROE is automatically cleared when the P*n*PSR register has been read by the CPU. Note: Overrun and Underrun conditions are caused by the B and/or D-channel receive or transmit data registers not being read or written prior to a 2-KHz super frame arriving.
5	DTDE	D data transmit data empty. This bit is set when the data in the PLTD transmit data register for the respective port has been transferred to the transmit shadow register. This bit is cleared when the CPU writes data to PLTD.
4	B2TDE	B2 data transmit data empty. This bit is set when the data in the P*n*B2TR transmit data register for the respective port has been transferred to the transmit shadow register. This bit is cleared when the CPU writes data to P*n*B2TR.
3	B1TDE	B1 data transmit data empty. This bit is set when the data in the P*n*B1TR transmit data register for the respective port has been transferred to the transmit shadow register. This bit is cleared when the CPU writes data to P*n*B1TR.

A write to QDR causes data to be written to the RAM entry specified by QAR[ADDR] and causes the value in QAR to increment. Correspondingly, a read at QDR returns the data in the RAM at the address specified by QAR[ADDR]. This also causes QAR to increment. A read access requires a single wait state.

Relative Address	Register	Function
0x00	QTR0	Transmit RAM
0x01	QTR1	
.	.	16 bits wide
.	.	
.	.	
0x0F	QTR15	
0x10	QRR0	Receive RAM
0x11	QRR1	
.	.	16 bits wide
.	.	
.	.	
0x1F	QRR15	
0x20	QCR0	Command RAM
0x21	QCR1	
.	.	8 bits wide
.	.	
.	.	
0x2F	QCR15	

Figure 14-2. QSPI RAM Model

14.4.1.1 Receive RAM

Data received by the QSPI is stored in the receive RAM segment located at 0x10 to 0x1F in the QSPI RAM space. The user reads this segment to retrieve data from the QSPI. Data words with less than 16 bits are stored in the least significant bits of the RAM. Unused bits in a receive queue entry are set to zero upon completion of the individual queue entry.

NOTE

Throughout ColdFire documentation, ‘word’ is used consistently and exclusively to designate a 16-bit data unit. The only exceptions to this appear in discussions of serial communication modules such as QSPI that support variable-length data units. To simplify these discussions the functional unit is referred to as a ‘word’ regardless of length.

QWR[CPTQP] shows which queue entries have been executed. The user can query this field to determine which locations in receive RAM contain valid data.

14.5.8 Programming Example

The following steps are necessary to set up the QSPI 12-bit data transfers and a QSPI_CLK of 4.125 MHz. The QSPI RAM is set up for a queue of 16 transfers. All four QSPI_CS signals are used in this example.

1. Enable all QSPI_CS pins on the MCF5272. Write PACNT with 0x0080_4000 to enable QSPI_CS1 and QSPI_CS3.Write PDCNT with 0x0000_0030 to enable QSPI_CS2.
2. Write the QMR with 0xB308 to set up 12-bit data words with the data shifted on the falling clock edge, and a clock frequency of 4.125 MHz (assuming a 66-MHz CLKIN).
3. Write QDLYR with the desired delays.
4. Write QIR with 0xD00D to enable write collision, abort bus errors, and clear any interrupts.
5. Write QAR with 0x0020 to select the first command RAM entry.
6. Write QDR with 0x7E00, 0x7E00, 0x7E00, 0x7E00, 0x7D00, 0x7D00, 0x7D00, 0x7D00, 0x7B00, 0x7B00, 0x7B00, 0x7B00, 0x7700, 0x7700, 0x7700, and 0x7700 to set up four transfers for each chip select. The chip selects are active low in this example.
7. Write QAR with 0x0000 to select the first transmit RAM entry.
8. Write QDR with sixteen 12-bit words of data.
9. Write QWR with 0x0F00 to set up a queue beginning at entry 0 and ending at entry 15.
10. Set QDLYR[SPE] to enable the transfers.
11. Wait until the transfers are complete. QIR[SPIF] is set when the transfers are complete.
12. Write QAR with 0x0010 to select the first receive RAM entry.
13. Read QDR to get the received data for each transfer.
14. Repeat steps 5 through 13 to do another transfer.

Chapter 18
Pulse-Width Modulation (PWM) Module

This chapter describes the configuration and operation of the pulse-width modulation (PWM) module. It includes a block diagram, programming model, and timing diagram.

18.1 Overview

The PWM module shown in Figure 18-1, generates a synchronous series of pulses having programmable duty cycle. With a suitable low-pass filter, the PWM can be used as a digital-to-analog converter.

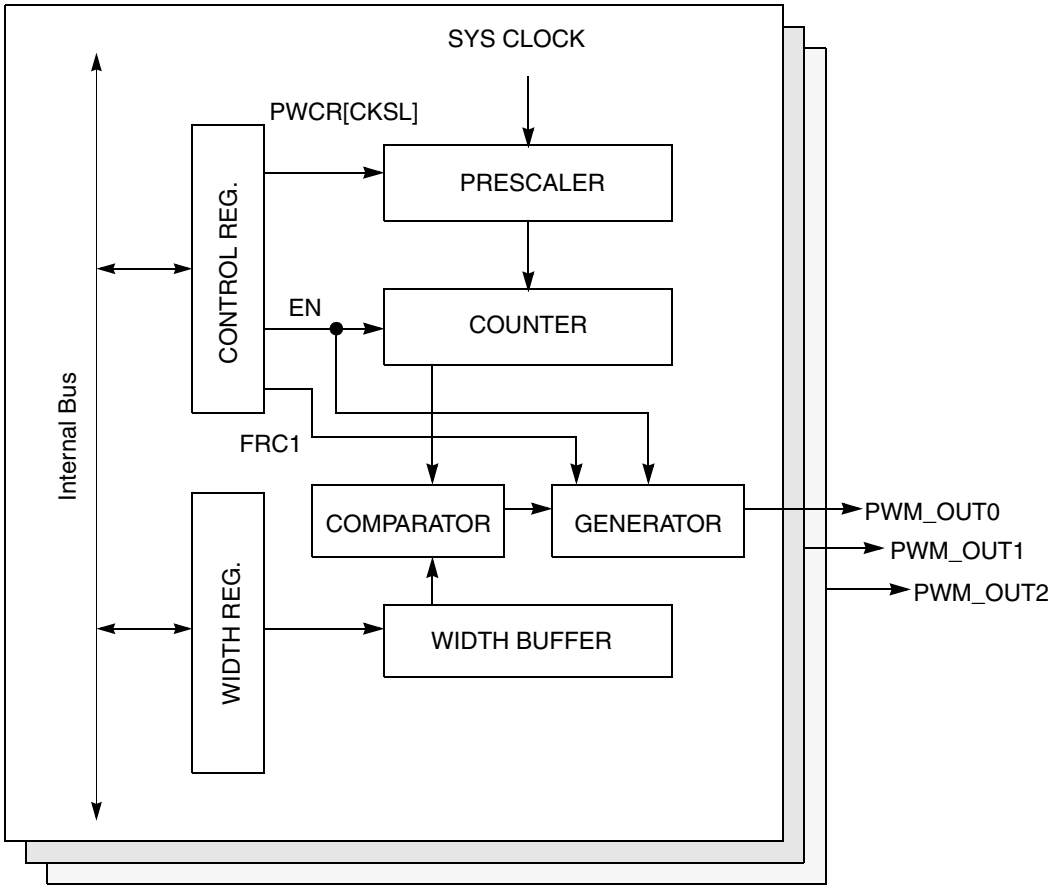

Figure 18-1. PWM Block Diagram (3 Identical Modules)

18.3.1 PWM Control Register (PWCR*n*)

This register, shown in Figure 18-2, controls the overall operation of the PWM. Unless disabled and then re-enabled, writing to PWCR while the PWM is running will not alter its operation until the current output cycle finishes. For example, if the prescale value is changed while the PWM is enabled, the new value will not take effect until after the counter has “wrapped around”. The PWM must be disabled and then re-enabled to affect its operation before the end of the current output cycle.

	7	6	5	4	3 … 0
Field	EN	FRC1	LVL	—	CKSEL
Reset	0010_0000				
R/W	Read/Write				
Address	MBAR + 0x0C0 (PWCR0); + 0x0C4 (PWCR1); + 0x0C8 (PWCR2)				

Figure 18-2. PWM Control Registers (PWCR*n*)

Table 18-2 gives PWCR field descriptions.

Table 18-2. PWCR*n* Field Descriptions

Bits	Name	Description
7	EN	Enable. 0 Disables the PWM. While disabled, the PWM is in low-power mode and the prescaler does not count. When the PWM is disabled, the output is forced to the value of PWCR*n*[LVL]. 1 Enables the PWM.
6	FRC1	Force output high. 0 Default reset value. PWM functions normally. 1 The PWM drives the output high for the entire counter period. PWCR*n*[FRC1] has a lower priority than PWCR*n*[EN], so setting PWCR*n*[FRC1] while PWCR*n*[EN] is cleared has no effect. There are two ways to drive the PWM output high. If PWCR*n*[EN] is cleared, PWM output immediately assumes the value of PWCR*n*[LVL]. If PWCR*n*[FRC1] is set while PWCR*n*[EN] is set, the PWM output does not go high until after the current output cycle completes.
5	LVL	Disable level. Determines the PWM output level whenever the PWM is disabled. 0 The PWM output is low while disabled. 1 The PWM output is high while disabled.
4	—	Reserved, should be cleared.
3–0	CKSL	Prescale clock. These bits select the clock frequency divider, that is, the output of the divider chain, as shown below. CKSL[3:0] Divisor 0000 1 0001 2 0010 4 111132768

Table 20-6 lists the bytes that should be driven on the data bus during read cycles by the external peripheral device being accessed, and the pattern of the data transfer for write cycles to the external data bus.

For read cycles, the entries shown as Byte X are portions of the requested operand that are read. The operand being read is defined by the size of the transfer and A[1:0] for the bus cycle. Bytes labeled “X” are don’t cares and are not required during that read cycle. Bytes labeled “–” are not valid transfers.

For write cycles from the internal multiplexer of the MCF5272 to the external data bus A[1:0] is incremented according to the transfer size. For example, if a longword transfer is generated to a 16-bit port, the MCF5272 starts the cycle with A[1:0] set to 0x0 and reads the first word. The address in then incremented to 0x2 and the second word is read. The data for both word reads is taken from D[31:16]. Bytes labeled “X” are don’t cares.

Table 20-6. Data Bus Requirement for Read/Write Cycles

Transfer Size	A[1:0]	External Data Bytes Required						
		32-Bit Port				16-Bit Port		8-Bit Port
		D[31:24]	D[23:16]	D[15:8]	D[7:0]	D[31:24]	D[23:16]	D[31:24]
Byte	00	Byte 0	X	X	X	Byte 0	X	Byte 0
	01	X	Byte 1	X	X	X	Byte 1	Byte 1
	10	X	X	Byte 2	X	Byte 2	X	Byte 2
	11	X	X	X	Byte 3	X	Byte 3	Byte 3
Word	00	Byte 0	Byte 1	X	X	Byte 0	Byte 1	Byte 0
	01	—	—	—	—	—	—	Byte 1
	10	X	X	Byte2	Byte 3	Byte 2	Byte 3	Byte 2
	11	—	—	—	—	—	—	Byte 3
Longword	00	Byte 0	Byte 1	Byte 2	Byte 3	Byte 0	Byte 1	Byte 0
	01	—	—	—	—	—	—	Byte 1
	10	—	—	—	—	Byte 2	Byte 3	Byte 2
	11	—	—	—	—	—	—	Byte 3
Line	00	Byte 0	Byte 1	Byte 2	Byte 3	Byte 0	Byte 1	Byte 0
	01	—	—	—	—	—	—	Byte 1
	10	—	—	—	—	Byte 2	Byte 3	Byte 2
	11	—	—	—	—	—	—	Byte 3

20.6 External Bus Interface Types

The MCF5272 supports three types of external bus interfaces. The interface type is programmed using CSBR*n*[EBI]. The EBI codes are summarized in Table 20-7.

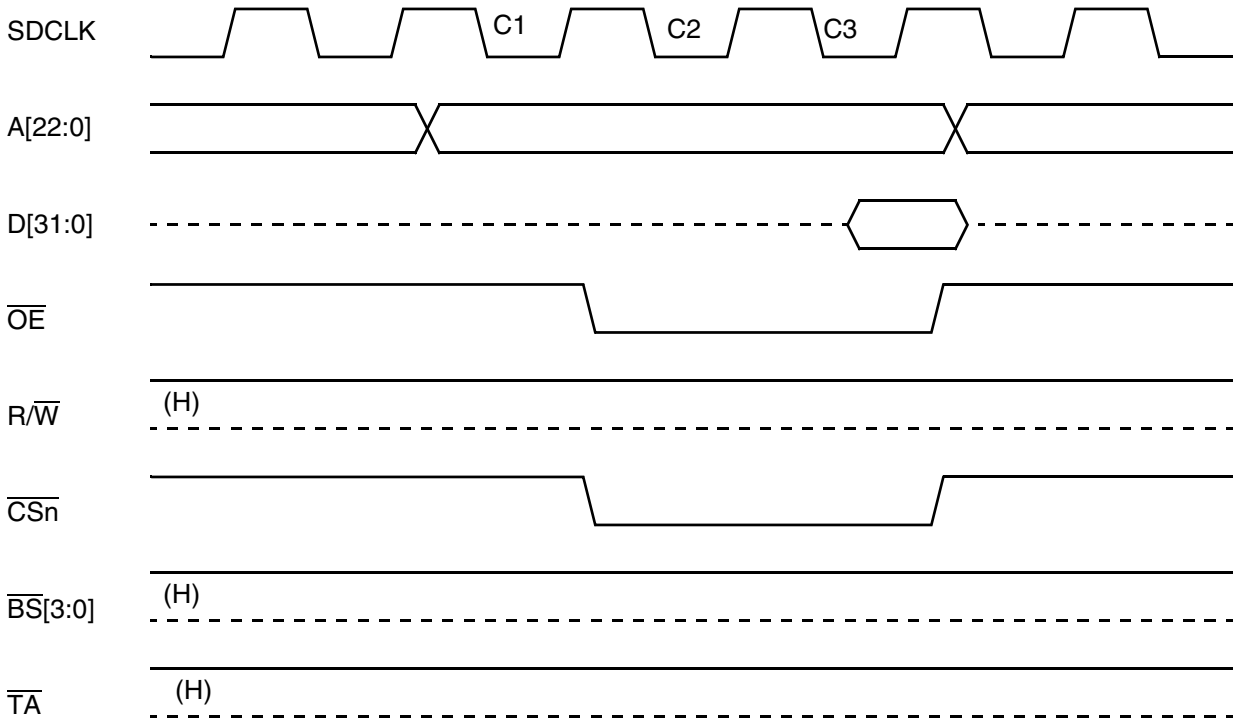

Figure 20-12. Read with Address Setup; EBI=11; 32-Bit Port; Internal Termination

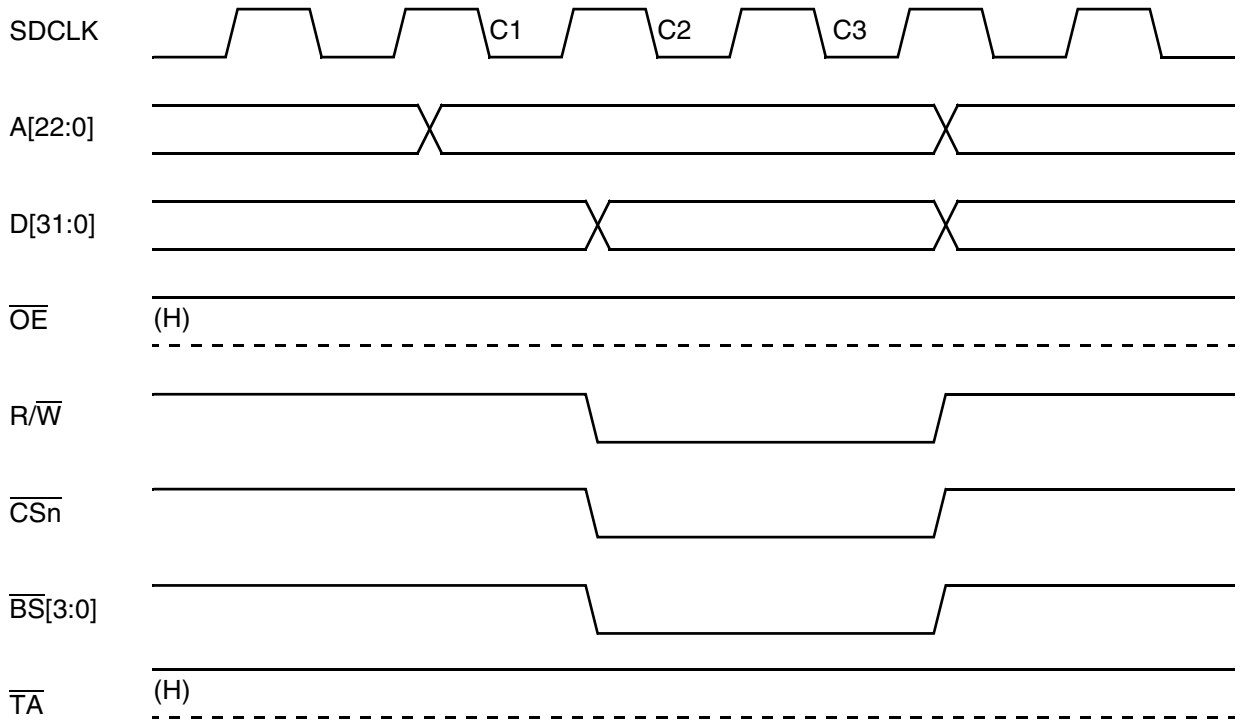

Figure 20-13. Longword Write with Address Setup; EBI=11; 32-Bit Port; Internal Termination

Chapter 21
IEEE 1149.1 Test Access Port (JTAG)

This chapter describes the dedicated user-accessible test logic implemented on the MCF5272. This test logic complies fully with the IEEE 1149.1 Standard Test Access Port and Boundary Scan Architecture. This chapter describes those items required by the standard and provides additional information specific to the MCF5272 implementation. For internal details and sample applications, see the IEEE 1149.1 document.

21.1 Overview

Problems with testing high-density circuit boards led to development of this standard under the sponsorship of the Test Technology Committee of IEEE and the Joint Test Action Group (JTAG). The MCF5272 supports circuit board test strategies based on this standard.

The test logic includes a test access port (TAP) consisting a 16-state controller, an instruction register, and three test registers (a 1-bit bypass register, a 265-bit boundary-scan register, and a 32-bit ID register). The boundary scan register links the device's pins into one shift register. The contents of this register can be found at the ColdFire website at http://www.freescale.com. Test logic, implemented using static logic design, is independent of the device system logic. The TAP includes the following dedicated signals:

- TCK—Test clock input to synchronize the test logic.
- TMS—Test mode select input (with an internal pullup resistor) that is sampled on the rising edge of TCK to sequence the TAP controller's state machine.
- TDI—Test data input (with an internal pull-up resistor) that is sampled on the rising edge of TCK.
- TDO—three-state test data output that is actively driven in the shift-IR and shift-DR controller states. TDO changes on the falling edge of TCK.

These signals, described in detail in Table 21-1, are enabled by negating the Freescale test mode signal (MTMOD).

The MCF5272 implementation can do the following:

- Perform boundary scan operations to test circuit board electrical continuity
- Sample MCF5272 system pins during operation and transparently shift out the result in the boundary scan register
- Bypass the MCF5272 for a given circuit board test by effectively reducing the boundary-scan register to a single bit
- Disable the output drive to pins during circuit-board testing
- Drive output pins to stable levels

NOTE

Precautions to ensure that the IEEE 1149.1 test logic does not interfere with non-test operation are described in Section 21.7, “Non-IEEE 1149.1 Operation.”

Figure 21-1 shows the MCF5272 implementation of IEEE 1149.1.

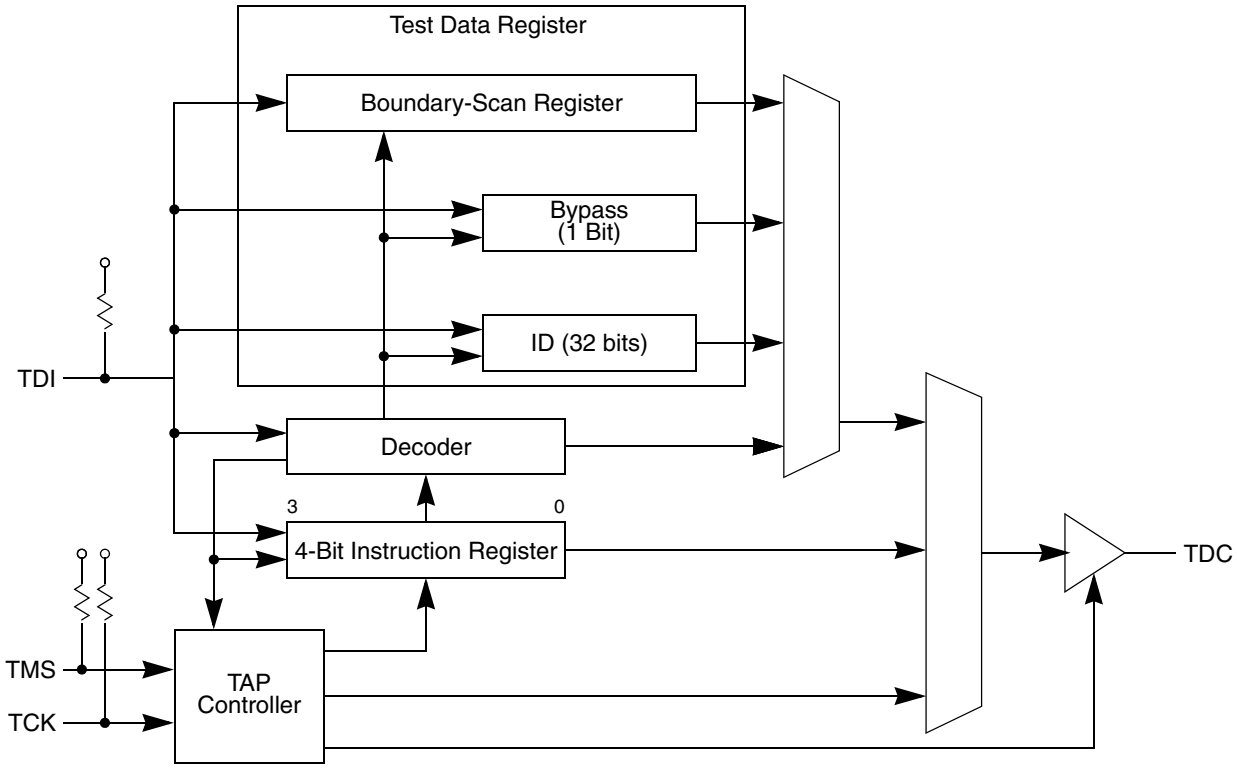

Figure 21-1. Test Access Port Block Diagram

21.2 JTAG Test Access Port and BDM Debug Port

The JTAG test interface shares pins with the debug modules (see Table 21-1).

Table 21-1. JTAG Signals

Signal	Description
TCK/ PSTCLK	Test clock. TCK is the dedicated JTAG test logic clock input, independent of the CPU system clock. It provides a clock for on-board test logic defined by the IEEE 1149.1 standard. TCK should be grounded if the JTAG port is not used and MTMOD is tied low.
$\overline{\text{TMS}}$/ $\overline{\text{BKPT}}$	Test mode select. This input controls test mode operations for on-board test logic defined by the IEEE 1149.1 standard. Connecting TMS to VDD disables the test controller, making all JTAG circuits transparent to the system.
TDO/ DSO	Test and debug data out. Output for shifting data out of serial data port logic. Shifting out data depends on the state of the JTAG controller state machine and the instructions in the instruction register. The shift occurs on the falling edge of TCK. When not outputting data, TDO is placed in high-impedance state. TDO can also be three-stated to allow bused or parallel connections to other devices having JTAG test access ports.

21.4 Boundary Scan Register

The boundary scan register contains bits for all device signal and clock pins and associated control signals. Bidirectional pins include a single scan bit for data (IO.Cell) as shown in Figure 21-6. These bits are controlled by an enable cell, shown in Figure 21-5. The control bit value determines whether the bidirectional pin is an input or an output. One or more bidirectional data bits can be serially connected to a control bit as shown in Figure 21-7. Note that when bidirectional data bits are sampled, bit data can be interpreted only after examining the I/O control bit to determine pin direction.

Open-drain bidirectional bits require separate input and output cells as no direction control is available from which to determine signal direction. Programmable open-drain signals also have an enable cell (XXX.de) to select whether the pin is open drain or push-pull. Signals with pull-up or pull-down resistors have an associated enable cell (XXX.pu); one enable cell can control multiple resistors.

Figure 21-3 to Figure 21-8 show the four MCF5272 cell types.

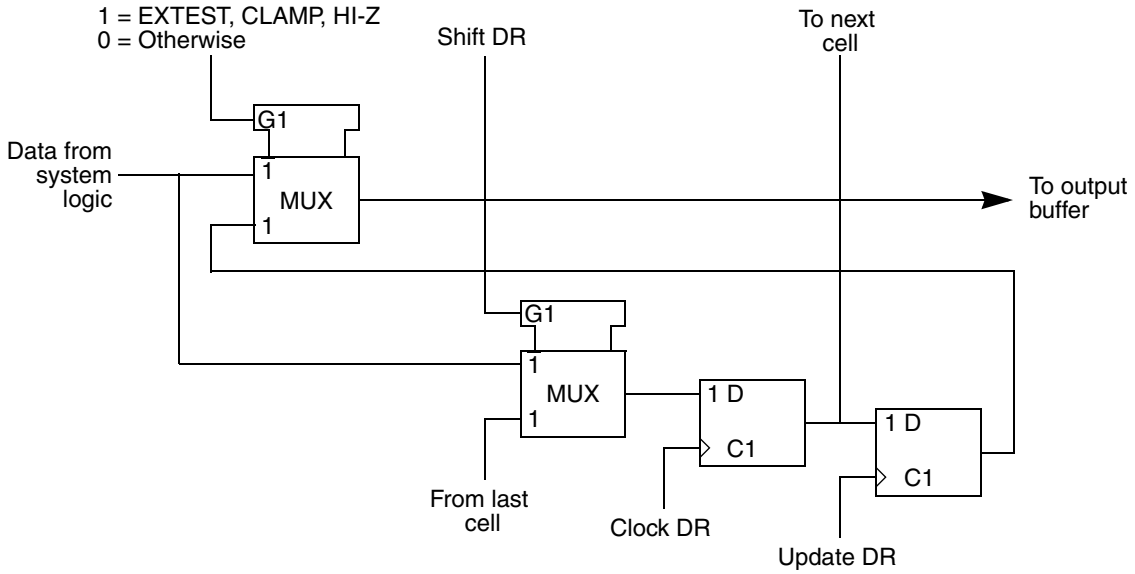

Figure 21-3. Output Cell (O.Cell) (BC–1)

23.3 AC Electrical Specifications

NOTE

AC timing specifications may be subject to change during ongoing qualification.

AC timing specifications assume maximum output load capacitance on all output pins including SDCLK. If this value is different, the input and output timing specifications would need to be adjusted to match the clock load.

AC timing specifications referenced to SDCLK assume SDRAM control register bit 3 is 0. After reset this bit is set.

23.3.1 Clock Input and Output Timing Specifications

Table 23-6 lists clock input and output timings.

Table 23-6. Clock Input and Output Timing Specifications

Name	Characteristic	0–66 MHz		Unit
		Min	Max	
	Frequency of operation	0	66.00	MHz
C1	CLKIN period (T) [1]	15	—	nS
C2 [2]	CLKIN fall time (from V_h = 2.4 V to V_l = 0.5 V)	—	2	nS
C3 [2]	CLKIN rise time (from V_l = 0.5 V to V_h = 2.4 V)	—	2	nS
C4	CLKIN duty cycle (measured at 1.5 V)	45	55	%
C4a [3]	CLKIN pulse-width high (measured at 1.5 V)	6.75	8.25	nS
C4b [3]	CLKIN pulse-width low (measured at 1.5 V)	6.75	8.25	nS

1 The clock period is referred to as T in the electrical specifications. The time for T is always in nS. Timing specifications can be given in terms of T. For example, 2T+5 nS

2 Specification values are not tested.

3 Specification values listed are for maximum frequency of operation.

Clock input and output timings listed in Table 23-6 are shown in Figure 23-1.

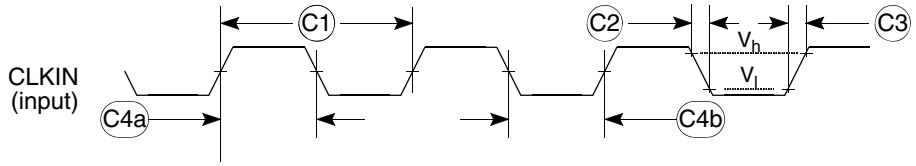

Figure 23-1. Clock Input Timing Diagram

Table A-2. CPU Space Registers Memory Map

CPU SPACE ADDRESS	NAME	Size	SYSTEM CONFIGURATION REGISTERS	Program Access	Debug Access
0x0002	(CACR)	32	Cache Control Register	MOVEC	RCREG, WCREG
0x0004	(ACR0)	32	Cache Access Control Register 0	MOVEC	RCREG, WCREG
0x0005	(ACR1)	32	Cache Access Control Register 1	MOVEC	RCREG, WCREG
0x008x	A7:A0	32	Address registers A7:A0	MOVE	RAREG, WAREG
0x008x	D7:D0	32	Data registers D7:D0	MOVE	RDREG, WDREG
0x0801	(VBR)	32	Vector Base Register	MOVEC	RCREG, WCREG
0x080E	CCR	8	Condition Code Register (Debug only)	MOVE to/from CCR	RCREG, WCREG
0x080F	PC	32	Program Counter (Debug only)		RCREG, WCREG
0x0C00	ROMBAR	32	ROM Base Address Register	MOVEC	RCREG, WCREG
0x0C04	RAMBAR	32	SRAM Base Address Register	MOVEC	RCREG, WCREG
0x0C0F	MBAR	32	Module Base Address Register	MOVEC	

NOTE

MBAR must only be written using the MOVEC instruction. Writing to address MBAR+0x0000 causes unpredictable device operation.

Table A-3. On-Chip Peripherals and Configuration Registers Memory Map

MBAR Offset	[31:24]	[23:16]	[15:8]	[7:0]
0x0000	Module Base Address Register, Read Only (MBAR)			
0x0004	System Configuration Register (SCR)		Reserved	
0x0006	Reserved		System Protection Register (SPR)	
0x0008	Power Management Register (PMR)			
0x000E	Reserved		Activate Low Power Register (ALPR)	
0x0010	Device Identification Register (DIR)			

Table A-4. Interrupt Control Register Memory Map

MBAR Offset	[31:24]	[23:16]	[15:8]	[7:0]
0x0020	Interrupt Control Register 1 (ICR1)			
0x0024	Interrupt Control Register 2 (ICR2)			
0x0028	Interrupt Control Register 3 (ICR3)			
0x002C	Interrupt Control Register 4 (ICR4)			
0x0030	Interrupt Source Register (ISR)			
0x0034	Programmable Interrupt Transition Register (PITR)			
0x0038	Programmable Interrupt Wakeup Register (PIWR)			
0x003F	Reserved			Programmable Interrupt Vector Register (PIVR)