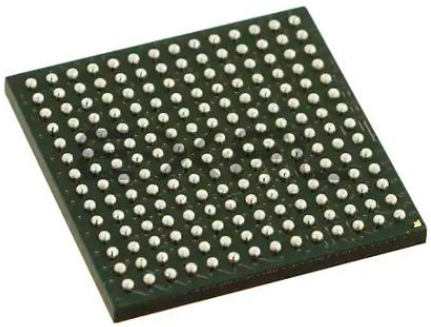

Welcome to E-XFL.COM

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	Obsolete
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	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	196-LBGA
Supplier Device Package	196-LBGA (15x15)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcf5272cvm66r2

Table of Contents (Continued)

Chapter 7
Interrupt Controller

Chapter 8
Chip Select Module

Table of Contents (Continued)

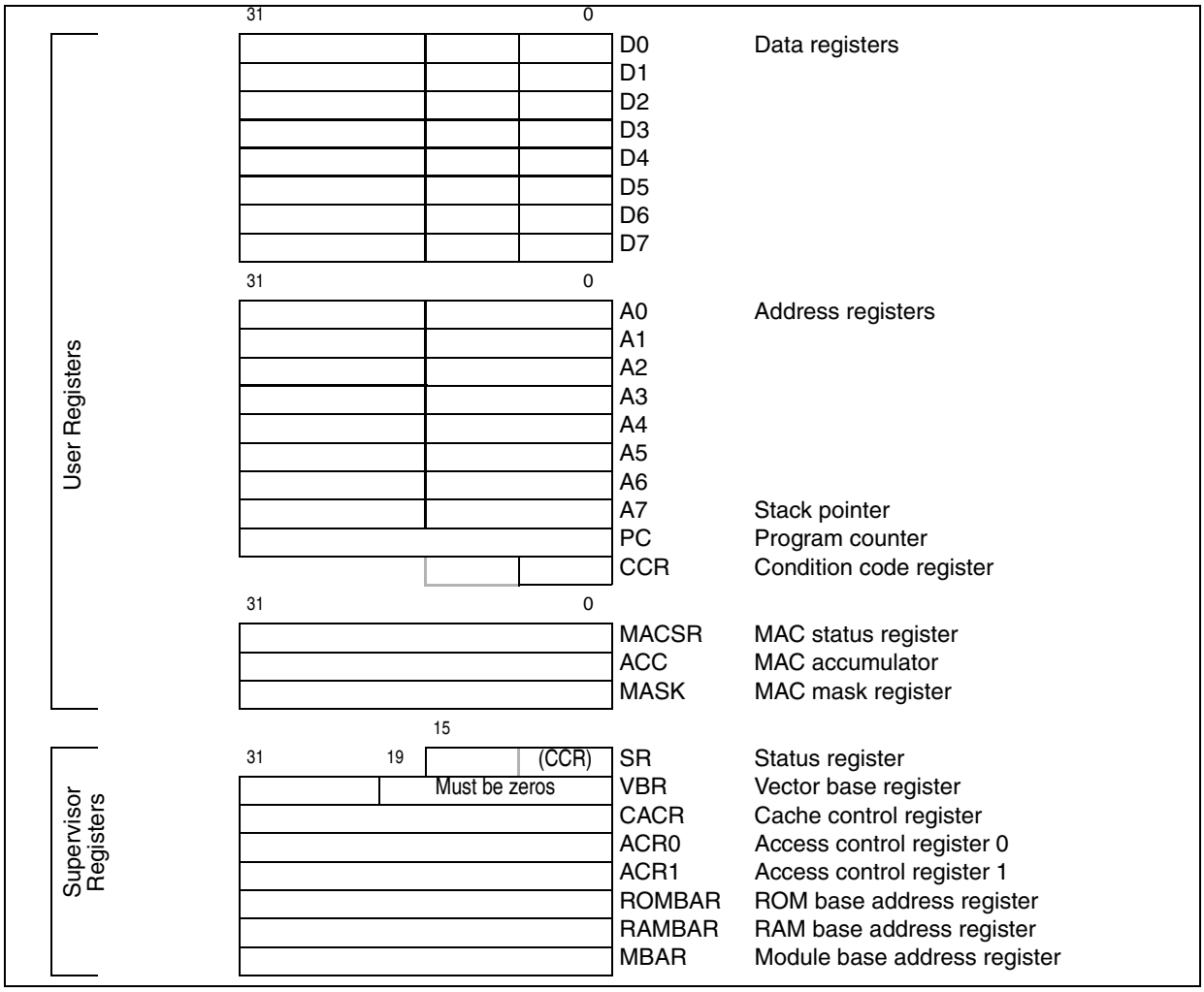

Figure 2-3. ColdFire Programming Model

2.2.1.1 Data Registers (D0–D7)

Registers D0–D7 are used as data registers for bit, byte (8-bit), word (16-bit), and longword (32-bit) operations. They may also be used as index registers.

2.2.1.2 Address Registers (A0–A6)

The address registers (A0–A6) can be used as software stack pointers, index registers, or base address registers and may be used for word and longword operations.

2.2.1.3 Stack Pointer (A7, SP)

The processor core supports a single hardware stack pointer (A7) used during stacking for subroutine calls, returns, and exception handling. The stack pointer is implicitly referenced by certain operations and can be explicitly referenced by any instruction specifying an address register. The initial value of A7 is loaded

2.2.2.1 Status Register (SR)

The SR stores the processor status, the interrupt priority mask, and other control bits. Supervisor software can read or write the entire SR; user software can read or write only SR[7–0], described in Section 2.2.1.5, “Condition Code Register (CCR).” The control bits indicate processor states—trace mode (T), supervisor or user mode (S), and master or interrupt state (M). SR is set to 0x27*xx* after reset.

	15					8	7					0
	System Byte						Condition Code Register (CCR)					
Field	T	—	S	M	—	I	—	X	N	Z	V	C
Reset	0	0	1	0	0	111	000	—	—	—	—	—
R/W	R/W	R	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W

Figure 2-5. Status Register (SR)

Table 2-3 describes SR fields.

Table 2-3. Status Field Descriptions

Bits	Name	Description
15	T	Trace enable. When T is set, the processor performs a trace exception after every instruction.
13	S	Supervisor/user state. Indicates whether the processor is in supervisor or user mode 0 User mode 1 Supervisor mode
12	M	Master/interrupt state. Cleared by an interrupt exception. It can be set by software during execution of the RTE or move to SR instructions so the OS can emulate an interrupt stack pointer.
10–8	I	Interrupt priority mask. Defines the current interrupt priority. Interrupt requests are inhibited for all priority levels less than or equal to the current priority, except the edge-sensitive level-7 request, which cannot be masked.
7–0	CCR	Condition code register. See Figure 2-4.

2.2.2.2 Vector Base Register (VBR)

The VBR holds the base address of the exception vector table in memory. The displacement of an exception vector is added to the value in this register to access the vector table. VBR[19–0] are not implemented and are assumed to be zero, forcing the vector table to be aligned on a 0-modulo-1-Mbyte boundary.

	31	20	19	0
Field	Exception vector table base address		—	
Reset	0000_0000_0000_0000_0000_0000_0000_0000			
R/W	Written from a BDM serial command or from the CPU using the MOVEC instruction. VBR can be read from the debug module only. The upper 12 bits are returned, the low-order 20 bits are undefined.			
Rc[11–0]	0x801			

Figure 2-6. Vector Base Register (VBR)

2.6 Instruction Set Summary

The ColdFire instruction set is a simplified version of the M68000 instruction set. The removed instructions include BCD, bit field, logical rotate, decrement and branch, and integer multiply with a 64-bit result. Nine new MAC instructions have been added.

Table 2-6 lists notational conventions used throughout this manual.

Table 2-6. Notational Conventions

Instruction	Operand Syntax
Opcode Wildcard	
cc	Logical condition (example: NE for not equal)
Register Specifications	
An	Any address register n (example: A3 is address register 3)
Ay,Ax	Source and destination address registers, respectively
Dn	Any data register n (example: D5 is data register 5)
Dy,Dx	Source and destination data registers, respectively
Rc	Any control register (example VBR is the vector base register)
Rm	MAC registers (ACC, MAC, MASK)
Rn	Any address or data register
Rw	Destination register w (used for MAC instructions only)
Ry,Rx	Any source and destination registers, respectively
Xi	index register i (can be an address or data register: Ai, Di)
Register Names	
ACC	MAC accumulator register
CCR	Condition code register (lower byte of SR)
MACSR	MAC status register
MASK	MAC mask register
PC	Program counter
SR	Status register
Port Names	
DDATA	Debug data port
PST	Processor status port
Miscellaneous Operands	
#<data>	Immediate data following the 16-bit operation word of the instruction
<ea>	Effective address

2.7 Instruction Timing

The timing data presented in this section assumes the following:

- The OEP is loaded with the opword and all required extension words at the beginning of each instruction execution. This implies that the OEP spends no time waiting for the IFP to supply opwords and/or extension words.
- The OEP experiences no sequence-related pipeline stalls. For the MCF5272,the most common example of this type of stall involves consecutive store operations, excluding the MOVEM instruction. For all store operations (except MOVEM), certain hardware resources within the processor are marked as busy for two clock cycles after the final DSOC cycle of the store instruction. If a subsequent store instruction is encountered within this two-cycle window, it is stalled until the resource again becomes available. Thus, the maximum pipeline stall involving consecutive store operations is two cycles.
- The OEP can complete all memory accesses without memory causing any stall conditions. Thus, timing details in this section assume an infinite zero-wait state memory attached to the core.
- All operand data accesses are assumed to be aligned on the same byte boundary as the operand size:
  — 16-bit operands aligned on 0-modulo-2 addresses
  — 32-bit operands aligned on 0-modulo-4 addresses
  Operands that do not meet these guidelines are misaligned. Table 2-9 shows how the core decomposes a misaligned operand reference into a series of aligned accesses.

Table 2-9. Misaligned Operand References

A[1:0]	Size	Bus Operations	Additional C(R/W) [1]
x1	Word	Byte, Byte	2(1/0) if read 1(0/1) if write
x1	Long	Byte, Word, Byte	3(2/0) if read 2(0/2) if write
10	Long	Word, Word	2(1/0) if read 1(0/1) if write

[1] Each timing entry is presented as C(r/w), described as follows:
C is the number of processor clock cycles, including all applicable operand fetches and writes, as well as all internal core cycles required to complete the instruction execution.
r/w is the number of operand reads (r) and writes (w) required by the instruction. An operation performing a read-modify write function is denoted as (1/1).

5.4.1 Revision A Shared Debug Resources

In the Revision A implementation of the debug module, certain hardware structures are shared between BDM and breakpoint functionality as shown in Table 5-4.

Table 5-4. Rev. A Shared BDM/Breakpoint Hardware

Register	BDM Function	Breakpoint Function
AATR	Bus attributes for all memory commands	Attributes for address breakpoint
ABHR	Address for all memory commands	Address for address breakpoint
DBR	Data for all BDM write commands	Data for data breakpoint

Thus, loading a register to perform a specific function that shares hardware resources is destructive to the shared function. For example, a BDM command to access memory overwrites an address breakpoint in ABHR. A BDM write command overwrites the data breakpoint in DBR.

5.4.2 Address Attribute Trigger Register (AATR)

The address attribute trigger register (AATR), Figure 5-5, defines address attributes and a mask to be matched in the trigger. The register value is compared with address attribute signals from the processor's local high-speed bus, as defined by the setting of the trigger definition register (TDR).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RM	SZM		TTM		TMM			R	SZ		TT		TM		
Reset	0000_0000_0000_0101															
R/W	Write only. AATR is accessible in supervisor mode as debug control register 0x06 using the WDEBUG instruction and through the BDM port using the WDMREG command.															
DRc[4–0]	0x06															

Figure 5-5. Address Attribute Trigger Register (AATR)

Table 5-5 describes AATR fields.

Table 5-5. AATR Field Descriptions

Bits	Name	Description
15	RM	Read/write mask. Setting RM masks R in address comparisons.
14–13	SZM	Size mask. Setting an SZM bit masks the corresponding SZ bit in address comparisons.
12–11	TTM	Transfer type mask. Setting a TTM bit masks the corresponding TT bit in address comparisons.
10–8	TMM	Transfer modifier mask. Setting a TMM bit masks the corresponding TM bit in address comparisons.
7	R	Read/write. R is compared with the R/$\overline{W}$ signal of the processor's local bus.
6–5	SZ	Size. Compared to the processor's local bus size signals. 00 Longword 01 Byte 10 Word 11 Reserved

Command Sequence:

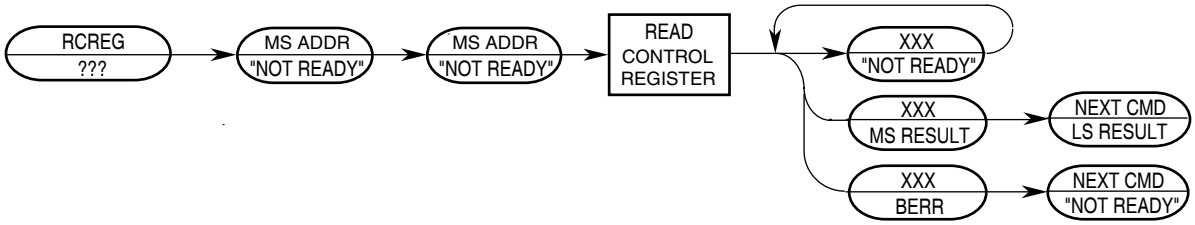

Figure 5-34. RCREG Command Sequence

Operand Data: The only operand is the 32-bit Rc control register select field.

Result Data: Control register contents are returned as a longword, most-significant word first. The implemented portion of registers smaller than 32 bits is guaranteed correct; other bits are undefined.

5.5.3.3.10 Write Control Register (WCREG)

The operand (longword) data is written to the specified control register. The write alters all 32 register bits.

Command/Result Formats:

	15–12	11–8	7–4	3–0
Command	0x2	0x8	0x8	0x0
	0x0	0x0	0x0	0x0
	0x0	Rc		
Result	D[31:16]			
	D[15:0]			

Figure 5-35. WCREG Command/Result Formats

Command Sequence:

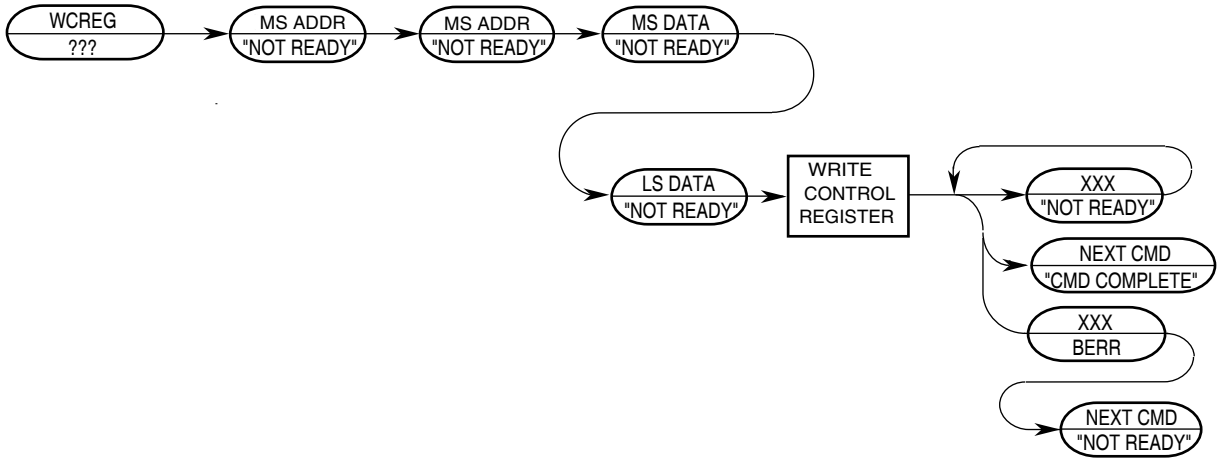

Figure 5-36. WCREG Command Sequence

Table 5-22. PST/DDATA Specification for User-Mode Instructions (continued)

Instruction	Operand Syntax	PST/DDATA
rems.l	<ea>y,Dx:Dw	PST = 0x1, {PST = 0xB, DD = source operand}
remu.l	<ea>y,Dx:Dw	PST = 0x1, {PST = 0xB, DD = source operand}
rts		PST = 0x1, {PST = 0xB, DD = source operand}, PST = 0x5, {PST = [0x9AB], DD = target address}
scc	Dx	PST = 0x1
sub.l	<ea>y,Rx	PST = 0x1, {PST = 0xB, DD = source operand}
sub.l	Dy,<ea>x	PST = 0x1, {PST = 0xB, DD = source}, {PST = 0xB, DD = destination}
subi.l	#imm,Dx	PST = 0x1
subq.l	#imm,<ea>x	PST = 0x1, {PST = 0xB, DD = source}, {PST = 0xB, DD = destination}
subx.l	Dy,Dx	PST = 0x1
swap	Dx	PST = 0x1
trap	#imm	PST = 0x1[3]
trapf		PST = 0x1
tst.b	<ea>x	PST = 0x1, {PST = 0x8, DD = source operand}
tst.l	<ea>x	PST = 0x1, {PST = 0xB, DD = source operand}
tst.w	<ea>x	PST = 0x1, {PST = 0x9, DD = source operand}
unlk	Ax	PST = 0x1, {PST = 0xB, DD = destination operand}
wddata.b	<ea>y	PST = 0x4, {PST = 0x8, DD = source operand
wddata.l	<ea>y	PST = 0x4, {PST = 0xB, DD = source operand
wddata.w	<ea>y	PST = 0x4, {PST = 0x9, DD = source operand

1 For JMP and JSR instructions, the optional target instruction address is displayed only for those effective address fields defining variant addressing modes. This includes the following <ea>x values: (An), (d16,An), (d8,An,Xi), (d8,PC,Xi).

2 For Move Multiple instructions (MOVEM), the processor automatically generates line-sized transfers if the operand address reaches a 0-modulo-16 boundary and there are four or more registers to be transferred. For these line-sized transfers, the operand data is never captured nor displayed, regardless of the CSR value.
The automatic line-sized burst transfers are provided to maximize performance during these sequential memory access operations.

3 During normal exception processing, the PST output is driven to a 0xC indicating the exception processing state. The exception stack write operands, as well as the vector read and target address of the exception handler may also be displayed.

```
Exception ProcessingPST = 0xC,{PST = 0xB,DD = destination},// stack frame
                  {PST = 0xB,DD = destination},// stack frame
                  {PST = 0xB,DD = source},// vector read
         PST = 0x5,{PST = [0x9AB],DD = target}// handler PC
```

The PST/DDATA specification for the reset exception is shown below:

```
Exception ProcessingPST = 0xC,
         PST = 0x5,{PST = [0x9AB],DD = target}// handler PC
```

The initial references at address 0 and 4 are never captured nor displayed since these accesses are treated as instruction fetches.

For all types of exception processing, the PST = 0xC value is driven at all times, unless the PST output is needed for one of the optional marker values or for the taken branch indicator (0x5).

5.7.2 Supervisor Instruction Set

The supervisor instruction set has complete access to the user mode instructions plus the opcodes shown below. The PST/DDATA specification for these opcodes is shown in Table 5-23.

Table 5-23. PST/DDATA Specification for Supervisor-Mode Instructions

Instruction	Operand Syntax	PST/DDATA
cpushl		PST = 0x1
halt		PST = 0x1, PST = 0xF
move.w	SR,Dx	PST = 0x1
move.w	{Dy,#imm},SR	PST = 0x1, {PST = 0x3}
movec	Ry,Rc	PST = 0x1
rte		PST = 0x7, {PST = 0xB, DD = source operand}, {PST = 3},{ PST =0xB, DD =source operand}, PST = 0x5, {[PST = 0x9AB], DD = target address}
stop	#imm	PST = 0x1, PST = 0xE
wdebug	<ea>y	PST = 0x1, {PST = 0xB, DD = source, PST = 0xB, DD = source}

The move-to-SR and RTE instructions include an optional PST = 0x3 value, indicating an entry into user mode. Additionally, if the execution of a RTE instruction returns the processor to emulator mode, a multiple-cycle status of 0xD is signaled.

Similar to the exception processing mode, the stopped state (PST = 0xE) and the halted state (PST = 0xF) display this status throughout the entire time the ColdFire processor is in the given mode.

Chapter 8
Chip Select Module

This chapter describes the chip select module, including the chip select registers, the configuration and behavior of the chip select signals, and the global chip select functions.

8.1 Overview

The chip select module provides user-programmable control of the eight chip select and four byte strobe outputs. This subsection describes the operation and programming model of the chip select registers, including the chip select base and option registers.

8.1.1 Features

The following list summarizes the key chip select features:

- Eight dedicated programmable chip selects
- Address masking for memory block sizes from 4 Kbytes to 2 Gbytes
- Programmable wait states and port sizes
- Programmable address setup
- Programmable address hold for read and write
- SDRAM controller interface supported with $\overline{\text{CS7}}/\overline{\text{SDCS}}$
- Global chip select functionality

8.1.2 Chip Select Usage

Each of the eight chip selects, $\overline{\text{CS0}}$–$\overline{\text{CS7}}$, is configurable for external SRAM, ROM, and peripherals. $\overline{\text{CS0}}$ is used to access external boot ROM and is enabled after a reset. The data bus width of the external ROM must be configured at reset by having appropriate pull-down resistors on QSPI_CLK/BUSW1 and QSPI_CS0/BUSW0. At reset these two signals replace the bus width field in the chip select 0 base register (CSBR0[BW]).

$\overline{\text{CS7}}$ must be used for enabling an external SDRAM array. In this mode, it is referred to as $\overline{\text{SDCS}}$.

NOTE

A detailed description of each bus access type supported by the MCF5272 device is given in Chapter 20, “Bus Operation.”

Table 9-2 shows how $\overline{BS}$[3:0] should be connected to DQM*x* for 16- and 32-bit SDRAM configurations.

Table 9-2. Connecting $\overline{BS}$[3:0] to DQM*x*

5272		SDRAM			Data Signals
16 Bit	32 Bit	16 Bit	32 Bit (2 x 16)	32 Bit (1 x 32)	
BS3	BS3	DQMH	DQMH	DQM3	D[31:24]
BS2	BS2	DQML	DQML	DQM2	D[23:16]
NC	BS1	NC	DQMH	DQM1	D[15:8]
NC	BS0	NC	DQML	DQM0	D[7:0]

9.3 Interface to SDRAM Devices

The following tables describes possible memory configurations using most common SDRAM devices for 16- and 32-bit wide data buses.

Table 9-3. Configurations for 16-Bit Data Bus

Parameter	8-Bit		16-Bit			
	16 Mbits	64 Mbits	16 Mbits	64 Mbits	128 Mbits	256 Mbits
Number of devices	2		1			
Total size	4 Mbytes	16 Mbytes	2 Mbytes	8 Mbytes	16 Mbytes	32 Mbytes
Total page size	1 Kbyte	1 Kbyte	512 Kbytes	512 Kbytes	1 Kbyte	1 Kbyte
Number of banks	2	4	2	4	4	4
Refresh count in 64 mS	4K	4K	4K	4K	4K	8K

Table 9-4. Configurations for 32-Bit Data Bus

Parameter	8-Bit		16-Bit				32-Bit	
	16 Mbits	64Mbits	16 Mbits	64 Mbits	128 Mbits	256 Mbits	64 Mbits	128 Mbits
Number of devices	4		2				1	
Total size	8 Mbytes	32 Mbytes	4 Mbytes	16 Mbytes	32 Mbytes	64 Mbytes	8 Mbytes	16 Mbytes
Total page size	2 Kbytes	2 Kbytes	1 Kbyte	1 Kbyte	2 Kbytes	2 Kbytes	1 Kbyte	1 Kbyte
Number of banks	2	4	2	4	4	4	4	4
Refresh count in 64 mS	4K	4K	4K	4K	4K	8K	4K	8K

A device's data bus width affects its connection to SDRAM address pins. A device can be configured for external data bus width of 16 or 32 bits by appropriate configuration of the WSEL signal during reset. See Section 19.18, "Operating Mode Configuration Pins." The following tables describe address pin connections and internal address multiplexing.

11.5.16.1 RAM Perfect Match Address High (MAUR)

The MAUR register contains bytes 4 and 5 of the 48-bit MAC address used in the address recognition process to compare with the destination address field of the receive frames. Byte 0 is the first byte transmitted on the network at the start of the frame.

This register is not reset and must be initialized by the user prior to operation. See Figure 11-21.

	31	16
Field	ADDR_HIGH	
Reset	Undefined	
R/W	Read/Write	

	15	0
Field	ADDR_HIGH	
Reset	0000_0000_0000_0000	
R/W	Read/Write	
Addr	MBAR + 0xC04	

Figure 11-21. RAM Perfect Match Address High (MAUR)

Table 11-24. MAUR Field Descriptions

Bits	Name	Description
31–0	ADDR_HIGH	Bytes 4 (bits 31–24) and 5 (bits 23–16) of the 6-byte address.

11.6.1.2 Ethernet Transmit Buffer Descriptor

Data is presented to the FEC for transmission by arranging it in buffers referenced by the channel's TxBDs. The FEC confirms transmission or indicates error conditions using the BDs to inform the host that the buffers have been serviced. In the TxBD, the user initializes the R, W, L, and TC bits and the length (in bytes) in the first word and the buffer pointer in the second words.

If L = 0, then the FEC sets the R bit to 0 in the first word of the BD when the buffer is sent as a DMA. Status bits are not modified.

If L = 1, then the FEC sets the R bit to 0 and will modify the DEF, HB, LC, RL, RC, UN, and CSL status bits in the first word of the BD after the buffer is sent as a DMA, and frame transmission is complete.

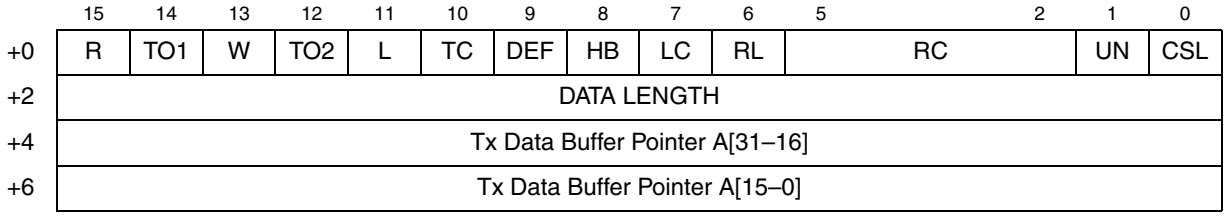

Figure 11-28. Transmit Buffer Descriptor (TxBD)

The TxBD fields are detailed in Table 11-35.

Table 11-35. TxBD Field Descriptions

Bits	Name	Description
15	R	Ready. Written by the FEC (= 0) and user (= 1). 0 The data buffer associated with this BD is not ready for transmission. The user is free to manipulate this BD or its associated data buffer. The FEC clears this bit after the buffer has been transmitted or after an error condition is encountered. 1 The data buffer, which has been prepared for transmission by the user, has not been transmitted or is currently being transmitted. No fields of this BD may be written by the user once this bit is set.
14	TO1	Transmit software ownership bit.Reserved for use by software. This read/write bit is not modified by hardware, nor does its value affect hardware.
13	W	Wrap. Written by the user. 0 The next buffer descriptor is found in the consecutive location. 1 The next buffer descriptor is found at the location defined in ETDSR.
12	TO2	Transmit software ownership bit. Reserved for use by software. This read/write bit is not modified by hardware, nor does its value affect hardware.
11	L	Last in frame. Written by the user. 0 The buffer is not the last in the transmit frame. 1 The buffer is the last in the transmit frame.
10	TC	Tx CRC. Written by the user and is only valid if L = 1. 0 End transmission immediately after the last data byte. 1 Transmit the CRC sequence after the last data byte.
9	DEF	Defer indication. Written by the FEC and is only valid if L = 1. The FEC had to defer while trying to transmit a frame. This bit is not set if a collision occurs during transmission.

9. Interface #0 Descriptor
10. Endpoint #1 Descriptor
11. Configuration #3 Descriptor
12. Interface #0 Descriptor
13. Endpoint #1 Descriptor
14. Endpoint #2 Descriptor

12.3.4 USB Module Access Times

The access times for the USB module depend on whether the access is to a register, to an endpoint FIFO (EP*n*DR register), or to the configuration RAM.

12.3.4.1 Registers

The USB module registers are accessed through the internal S-bus. Each register access takes 3 clock cycles for reads and writes.

12.3.4.2 Endpoint FIFOs

The FIFO access time depends on the size, the time between accesses, and whether the previous FIFO access was for the same endpoint. After a longword access to an endpoint's FIFO, the next longword in the FIFO is cached for a quicker access time on the next longword read. This mechanism is reset every time another endpoint is accessed. Table 12-19 shows the access times for the FIFOs.

Table 12-19. USB FIFO Access Timing

Access Type	Read	Write
Byte	5	4
Word	6	4
Long (back to back)	8-4-6-6-6-6...	4-6-6-6-6-6...
Long (1 clock gap)	8-3-5-5-5-5...	4-5-5-5-5-5...
Long (2 clock gap)	8-3-4-4-4-4...	4-4-4-4-4-4...
Long (3+ clock gap)	8-3-3-3-3-3...	4-4-4-4-4-4...

12.3.4.3 Configuration RAM

The configuration RAM is longword accessible only. Access times for reads from the configuration RAM are eight clock cycles per access. Clock cycle access times for back-to-back writes to the configuration RAM are 3-5-5-5-5-5... Access times for writes separated by at least 1 clock cycle are 3-3-3-3-3-3…

16.3.7 UART Transmitter Buffers (UTB*n*)

The transmitter buffers consist of a 24-byte FIFO and the transmitter shift register. The FIFO accepts characters from the bus master if the channel's USR*n*[TxRDY] is set. A write to the transmitter buffer clears TxRDY, inhibiting any more characters until the FIFO can accept more data. When the shift register is empty, it checks if the holding register has a valid character to be sent (TxRDY = 0). If there is a valid character, the shift register loads it and sets USR*n*[TxRDY] again. Writes to the transmitter buffer have no effect when the channel's TxRDY = 0 and when the transmitter is disabled.

Figure 16-8 shows UTB*n*. TB contains the character in the transmitter buffer.

	7 … 0
Field	TB
Reset	0000_0000
R/W	Write only
Address	MBAR + 0x10C,0x14C

Figure 16-8. UART Transmitter Buffers (UTB*n*)

16.3.8 UART Input Port Change Registers (UIPCR*n*)

The input port change registers (UIPCR*n*), Figure 16-9, hold the current state and the change-of-state for $\overline{\text{CTS}}$.

	7 … 5	4	3 … 1	0
Field	—	COS	111	CTS
Reset	0000_011			CTS
R/W	Read only			
Address	MBAR + 0x110 (UIPCR0), 0x150 (UIPCR1)			

Figure 16-9. UART Input Port Change Registers (UIPCR*n*)

Table 16-7 describes UIPCR*n* fields.

Table 16-7. UIPCR*n* Field Descriptions

Bits	Name	Description
7–5	—	Reserved, should be cleared.
4	COS	Change of state (high-to-low or low-to-high transition). 0 No change-of-state since the CPU last read UIPCR*n*. Reading UIPCR*n* clears UISR*n*[COS]. 1 A change-of-state longer than 25–50 μs occurred on the $\overline{\text{CTS}}$ input. UACR*n* can be programmed to generate an interrupt to the CPU when a change of state is detected.
3–1	—	Reserved, should be cleared.
0	CTS	Current state. Starting two serial clock periods after reset, CTS reflects the state of $\overline{\text{CTS}}$. If $\overline{\text{CTS}}$ is detected asserted at that time, COS is set, which initiates an interrupt if UACR*n*[IEC] is enabled. 0 The current state of the $\overline{\text{CTS}}$ input is asserted. 1 The current state of the $\overline{\text{CTS}}$ input is negated.

Table 17-4. Port A Control Register Function Bits

Pin Number	PACNT[*xx*] = 00 (Function 0b00)	PACNT[*xx*] = 01 (Function 0b01)	PACNT[*xx*] = 10 (Function 0b10)	PACNT[*xx*] = 11 (Function 0b11)
D2	PA0	USB_TP	—	—
D1	PA1	USB_RP	—	—
E5	PA2	USB_RN	—	—
E4	PA3	USB_TN	—	—
E3	PA4	USB_Susp	—	—
E2	PA5	USB_TxEN	—	—
E1	PA6	USB_RxD	—	—
P1	PA7	QSPI_CS3	DOUT3	—
J2	PA8	FSC0/FSR0	—	—
J3	PA9	DGNT0	—	—
K5	PA10	DREQ0	—	—
L1	PA11	Reserved	QSPI_CS1	—
L2	PA12	DFSC2	—	—
L3	PA13	DFSC3	—	—
M2	PA14	DREQ1	—	—
M3	PA15	DGNT1[1]	—	—

[1] If this pin is programmed to function as $\overline{\text{INT6}}$, it is not available as a GPIO.

17.2.2 Port B Control Register (PBCNT)

PBCNT, shown in Figure 17-2, is used to configure the pins assigned to signals that are multiplexed with GPIO port B.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	PBCNT15		PBCNT14		PBCNT13		PBCNT12		PBCNT11		PBCNT10		PBCNT9		PBCNT8	
Reset	0000_0000_0000_0000															
R/W	Read/Write															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	PBCNT7		PBCNT6		PBCNT5		PBCNT4		PBCNT3		PBCNT2		PBCNT1		PBCNT0	
Reset	0000_0000_0000_0000															
R/W	Read/Write															
Addr	MBAR + 0x0088															

Figure 17-2. Port B Control Register (PBCNT)

Table 17-5 describes PBCNT fields. Table 17-6 provides the same information organized by function.

19.13.10 Receive Error (E_RxER/PB14)

Ethernet mode: E_RxER is an input signal which when asserted along with E_RxDV signals that the PHY has detected an error in the current frame. When E_RxDV is not asserted E_RxER has no effect. Applies to MII mode operation.

Port B mode: This pin can also be configured as PB14 I/O.

19.13.11 Management Data Clock (E_MDC/PB15)

Ethernet mode: E_MDC is an output clock which provides a timing reference to the PHY for data transfers on the E_MDIO signal. Applies to MII mode operation.

Port B mode: This pin can also be configured as I/O pin PB15.

19.13.12 Management Data (E_MDIO)

The bidirectional E_MDIO signal transfers control information between the external PHY and the media-access controller. Data is synchronous to E_MDC. Applies to MII mode operation. This signal is an input after reset. When the FEC is operated in 10Mbps 7-wire interface mode, this signal should be connected to Vss.

19.13.13 Transmit Error (E_TxER)

Ethernet mode: When the E_TxER output is asserted for one or more clock cycles while E_TxEN is also asserted, the PHY sends one or more illegal symbols. E_TxER has no effect at 10 Mbps or when E_TxEN is negated. Applies to MII mode operation.

19.13.14 Carrier Receive Sense (E_CRS)

E_CRS is an input signal which when asserted signals that transmit or receive medium is not idle. Applies to MII mode operation.

19.14 PWM Module Signals (PWM_OUT0–PWM_OUT2])

PWM_OUT0–PWM_OUT2 are the outputs of the compare logic within the pulse-width modulator (PWM) modules.

- PWM_OUT0 is always available.
- PWM_OUT1 is multiplexed with TOUT1.
- PWM_OUT2 is multiplexed with TIN1.

19.15 Queued Serial Peripheral Interface (QSPI) Signals

This section describes signals used by the queued serial peripheral interface (QSPI) module. Four QSPI chip selects, QSPI_CS[3:0], are multiplexed with the physical layer interface pins and GPIO port A. QSPI_CS0 is always available. QSPI_CS3 is multiplexed with DOUT3 and PA7.

GCI mode: GDCL1_OUT is used to clock data in and out of DIN1 and DOUT1 for GCI port 1. DCL1 is twice the bit rate; that is, two clocks per data bit.

When this pin is configured as an output, the GDCL1_OUT clock signal from the on-chip synthesizer clock generator is output on this pin. Also GDCL1_OUT is used to internally drive all ports and delayed sync generators associated with ports 1, 2, and 3.

19.16.2.2 GCI/IDL Data Out (DOUT1)

IDL mode: The DOUT1 output is for clocking data out of IDL port 1. Data is clocked out of DOUT1 on the rising edge of DCL1. DOUT1 is also used for clocking data from ports 2 and 3.

GCI mode: The DOUT1 output is for clocking data out of GCI port 1. DCL1 is twice the bit rate, that is, two clocks per data bit.

19.16.2.3 GCI/IDL Data In (DIN1)

IDL mode: The DIN1 input is for clocking data into IDL port 1. Data is clocked into DIN1 on the falling edge of DCL1. DIN1 is also used for clocking data into ports 2 and 3.

GCI mode: The DIN1 input is for clocking data into GCI port 1. DCL1 is twice the bit rate, that is, two clocks per data bit. DIN1 is also used for clocking data into ports 2 and 3.

19.16.2.4 GCI/IDL Frame Sync (FSC1/FSR1/DFSC1)

IDL mode: FSR1 is an input for the 8-KHz frame sync for port 1.

GCI mode: FSC1 is an input for the 8-KHz frame sync for port 1. Normally the GCI FSC signal is two clocks wide and is aligned with the first B channel bit of the GCI frame. Many U-interface devices including the MC145572 and MC145576 change the width of FSC to one clock every 12 mS, indicating a U-interface super frame boundary. The MCF5272 can accept either frame sync width.

When this pin is configured as an output, the DFSC1 sync signal from the on-chip clock synthesizer is output on this pin. Also DFSC1 is used to internally drive the port 1 frame sync and the delayed sync generators associated with ports 2 and 3. The width of DFSC1 can be configured for 1, 2, 8, or 16 DCL clocks duration. The location of DFSC1 is programmable in single clock increments up to a maximum count of 0x3FF.

19.16.2.5 D-Channel Request (DREQ1/PA14)

IDL mode: This pin can be configured as the DREQ1 output in IDL mode for signalling to a layer 1 S/T transceiver that a frame of data is ready to be sent on the port 1 D channel.

Port A mode: I/O pin PA14.

19.16.2.6 D-Channel Grant ($\overline{\text{DGNT1\_INT6}}$/$\overline{\text{PA15\_INT6}}$)

This pin can be independently configured as the input, DGNT1, used by a Layer one ISDN S/T transceiver to indicate that D-channel access has been granted.

NOTE

The levels of the mode pins are not sampled during a software watchdog reset. If the port size and acknowledge features of $\overline{CS0}$ are different from the values programmed in CSBR0 and CSOR0 at the time of the software watchdog reset, you must assert $\overline{RSTI}$ during software watchdog reset to cause the mode pins to be resampled.

20.12.4 Soft Reset Operation

If the soft reset bit, SCR[SOFTRST], is programmed to generate a reset, $\overline{RSTO}$ is asserted for 128 clocks, resetting all external devices as with a normal or master reset. All internal peripherals with the exception of the SIM, chip select, interrupt controller, GPIO module, and SDRAM controller are reset also. The SDRAM controller is reset only when $\overline{DRESETEN}$ is tied low.

SCR[SOFTRST] is automatically cleared at the end of the 128 clock period. Software can monitor this bit to determine the end of the soft reset. Figure 20-24 shows the timing of $\overline{RSTO}$ when asserted by SCR[SOFTRST].

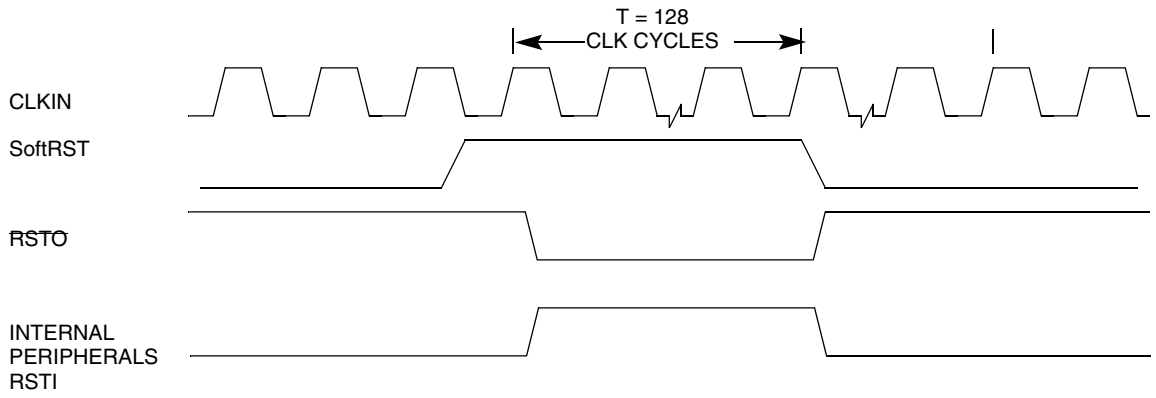

Figure 20-24. Soft Reset Timing

NOTE

Like the normal reset, the soft reset does not reset the SDRAM controller unless $\overline{DRESETEN}$ is asserted during the reset. When $\overline{DRESETEN}$ is negated, SDRAM refreshes continue to be generated during and after reset at the programmed rate and with the programmed waveform timing.

During the soft reset period, all bus signals continue to operate normally.