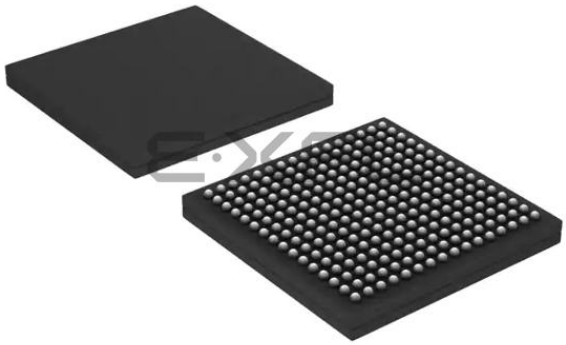

Welcome to E-XFL.COM

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	80MHz
Connectivity	CANbus, EBI/EMI, Ethernet, I²C, SPI, UART/USART
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	150
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	256-LBGA
Supplier Device Package	256-MAPBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcf5282cvf80j

Chapter 9
Clock Module

Chapter 10
Interrupt Controller Modules

Chapter 16
DMA Controller Module

Table 4-5. ACR*n* Field Descriptions

Field	Description
31–24 AB	Address base. This 8-bit field is compared to address bits [31:24] from the processor's local bus under control of the ACR address mask. If the address matches, the attributes for the memory reference are sourced from the given ACR.
23–16 AM	Address mask. Masks any AB bit. If a bit in the AM field is set, the corresponding bit of the address field comparison is ignored.
15 EN	ACR Enable. Hardware reset clears this bit, disabling the ACR. 0 ACR disabled 1 ACR enabled
14–13 SM	Supervisor mode. Allows the given ACR to be applied to references based on operating privilege mode of the ColdFire processor. The field uses the ACR for user references only, supervisor references only, or all accesses. 00 Match if user mode 01 Match if supervisor mode 1*x* Match always—ignore user/supervisor mode
12–7	Reserved, must be cleared.
6 CM	Cache mode. 0 Caching enabled 1 Caching disabled
5 BWE	Buffered write enable. Defines the value for enabling buffered writes. If BWE is cleared, the termination of an operand write cycle on the processor's local bus is delayed until the system bus cycle is completed. Setting BWE terminates the write cycle on the local bus immediately and the operation is then buffered in the bus controller. In this mode, operand write cycles are effectively decoupled between the processor's local bus and the system bus. Generally, the enabling of buffered writes provides higher system performance but recovery from access errors may be more difficult. For the V2 ColdFire core, the reporting of access errors on operand writes is always imprecise, and enabling buffered writes simply decouples the write instruction from the signaling of the fault even more. 0 Writes are not buffered. 1 Writes are buffered.
4–3	Reserved, must be cleared.
2 WP	Write protect. Defines the write-protection attribute. If the effective memory attributes for a given access select the WP bit, an access error terminates any attempted write with this bit set. 0 Read and write accesses permitted 1 Only read accesses permitted
1–0	Reserved, must be cleared.

4.3 Functional Description

The cache is physically connected to the ColdFire core's local bus, allowing it to service all fetches from the ColdFire core and certain memory fetches initiated by the debug module. Typically, the debug module's memory references appear as supervisor data accesses but the unit can be programmed to generate user-mode accesses and/or instruction fetches. The cache processes any fetch access in the normal manner.

4.3.1 Interaction with Other Modules

Because the cache and high-speed SRAM module are connected to the ColdFire core's local data bus, certain user-defined configurations can result in simultaneous fetch processing.

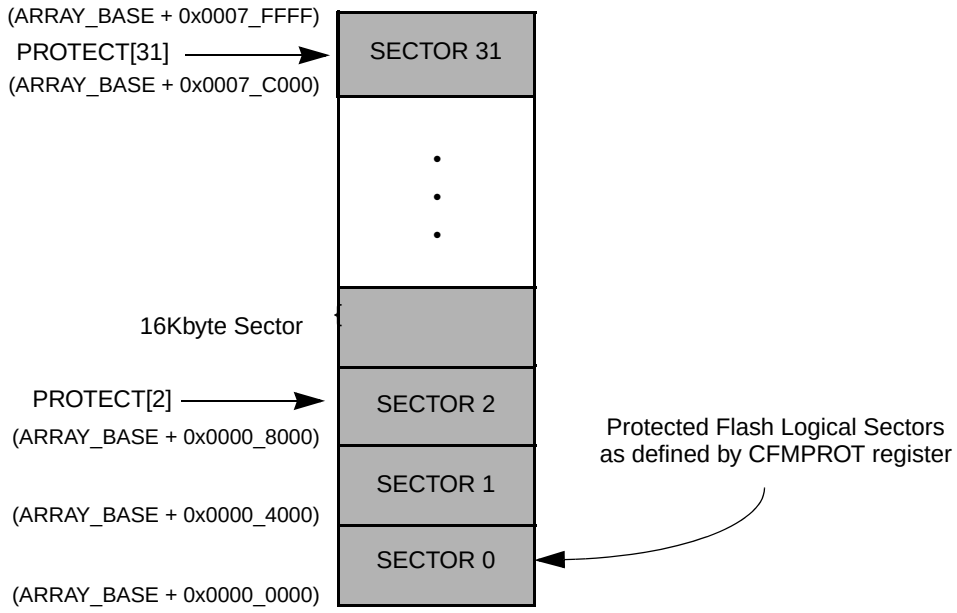

Figure 6-8. CFMPROT Protection Diagram

6.3.4.5 CFM Supervisor Access Register (CFMSACC)

The CFMSACC specifies the supervisor/user access permissions of Flash logical sectors.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	SUPV															
Reset	See Note															
R/W	R/W															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SUPV															
Reset	See Note															
R/W	R/W															
Address	IPSBAR + 0x1D_0014															

Note: The CFMPROT register is loaded at reset from the Flash Supervisor/user Space Restrictions longword stored at the array base address + 0x0000_040C.

Figure 6-9. CFM Supervisor Access Register (CFMSACC)

Chapter 8
System Control Module (SCM)

This section details the functionality of the System Control Module (SCM) which provides the programming model for the System Access Control Unit (SACU), the system bus arbiter, a 32-bit core watchdog timer (CWT), and the system control registers and logic. Specifically, the system control includes the internal peripheral system (IPS) base address register (IPSBAR), the processor's dual-port RAM base address register (RAMBAR), and system control registers that include the core watchdog timer control.

8.1 Overview

The SCM provides the control and status for a variety of functions including base addressing and address space masking for both the IPS peripherals and resources (IPSBAR) and the ColdFire core memory spaces (RAMBAR). The CPU core supports two memory banks, one for the internal SRAM and the other for the internal Flash.

The SACU provides the mechanism needed to implement secure bus transactions to the system address space.

The programming model for the system bus arbitration resides in the SCM. The SCM sources the necessary control signals to the arbiter for bus master management.

The CWT provides a means of preventing system lockup due to uncontrolled software loops via a special software service sequence. If periodic software servicing action does not occur, the CWT times out with a programmed response (interrupt) to allow recovery or corrective action to be taken.

8.2 Features

The SCM includes these distinctive features:

- IPS base address register (IPSBAR)
  - — Base address location for 1-Gbyte peripheral space
  - — User control bits
- Processor-local memory base address register (RAMBAR)
- System control registers
  - — Core reset status register (CRSR) indicates type of last reset
  - — Core watchdog control register (CWCR) for watchdog timer control
  - — Core watchdog service register (CWSR) to service watchdog timer
- System bus master arbitration programming model (MPARK)
- System access control unit (SACU) programming model
  - — Master privilege register (MPR)
  - — Peripheral access control registers (PACRs)
  - — Grouped peripheral access control registers (GPACR0, GPACR1)

Table 9-4. SYNCR Field Descriptions (continued)

<table>
<tr><th>Bit(s)</th><th>Name</th><th>Description</th></tr>
<tr><td>14–12</td><td>MFD</td><td>Multiplication Factor Divider. Contain the binary value of the divider in the PLL feedback loop. The MFD[2:0] value is the multiplication factor applied to the reference frequency. When MFD[2:0] are changed or the PLL is disabled in stop mode, the PLL loses lock. In 1:1 PLL mode, MFD[2:0] are ignored, and the multiplication factor is one.
Note: In external clock mode, the MFD[2:0] bits have no effect.
The following table illustrates the system frequency multiplier of the reference frequency[1] in normal PLL mode.
<table>
<tr><th></th><th></th><th colspan="8">MFD[2:0]</th></tr>
<tr><th></th><th></th><th>000[2] (4x)</th><th>001 (6x)</th><th>010 (8x)[(3)]</th><th>011 (10x)</th><th>100 (12x)</th><th>101 (14x)</th><th>110 (16x)</th><th>111 (18x)</th></tr>
<tr><td rowspan="8">RFD[2:0]</td><td>000 (÷ 1)</td><td>4</td><td>6</td><td>8</td><td>10</td><td>12</td><td>14</td><td>16</td><td>18</td></tr>
<tr><td>001 (÷ 2)[3]</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr>
<tr><td>010 (÷ 4)</td><td>1</td><td>3/2</td><td>2</td><td>5/2</td><td>3</td><td>7/2</td><td>4</td><td>9/2</td></tr>
<tr><td>011 (÷ 8)</td><td>1/2</td><td>3/4</td><td>1</td><td>5/4</td><td>3/2</td><td>7/4</td><td>2</td><td>9/4</td></tr>
<tr><td>100 (÷ 16)</td><td>1/4</td><td>3/8</td><td>1/2</td><td>5/8</td><td>3/4</td><td>7/8</td><td>1</td><td>9/8</td></tr>
<tr><td>101 (÷ 32)</td><td>1/8</td><td>3/16</td><td>1/4</td><td>5/16</td><td>3/8</td><td>7/16</td><td>1/2</td><td>9/16</td></tr>
<tr><td>110 (÷ 64)</td><td>1/16</td><td>3/32</td><td>1/8</td><td>5/32</td><td>3/16</td><td>7/32</td><td>1/4</td><td>9/32</td></tr>
<tr><td>111 (÷ 128)</td><td>1/32</td><td>3/64</td><td>1/16</td><td>5/64</td><td>3/32</td><td>7/64</td><td>1/8</td><td>9/64</td></tr>
</table>
1 $f_{sys} = \frac{f_{ref} \times 2(MFD + 2)}{2^{RFD}}$; $f_{ref} \times 2(MFD + 2) \leq f_{sys(max)}$; $f_{sys} \leq f_{sys(max)}$,
where $f_{sys(max)}$ is the maximum system frequency for the particular MCF5282 device (66 MHz or 80 MHz).
2 MFD = 000 not valid for f_{ref} < 3 MHz
3 Default value out of reset</td></tr>
<tr><td>11</td><td>LOCRE</td><td>Loss-of-clock reset enable. Determines how the system handles a loss-of-clock condition. When the LOCEN bit is clear, LOCRE has no effect. If the LOCS flag in SYNSR indicates a loss-of-clock condition, setting the LOCRE bit causes an immediate reset. To prevent an immediate reset, the LOCRE bit must be cleared before entering stop mode with the PLL disabled.
1 Reset on loss-of-clock
0 No reset on loss-of-clock
Note: In external clock mode, the LOCRE bit has no effect.</td></tr>
<tr><td>10–8</td><td>RFD</td><td>Reduced frequency divider field. The binary value written to RFD[2:0] is the PLL frequency divisor. See table in MFD bit description. Changing RFD[2:0] does not affect the PLL or cause a relock delay. Changes in clock frequency are synchronized to the next falling edge of the current system clock. To avoid surpassing the allowable system operating frequency, write to RFD[2:0] only when the LOCK bit is set.</td></tr>
</table>

Table 9-10. Stop Mode Operation (Sheet 3 of 5)

MODE In	LOCEN	LOCRE	LOLRE	PLL	OSC	EWKUP	Expected PLL Action at Stop	PLL Action During Stop	MODE Out	LOCKSS	LOCK	LOCS	Comments
NRM	0	0	1	On	On	X	—	—	NRM	‘LK	1	‘LC	
								Lose lock or clock	RESET	—	—	—	Reset immediately
NRM	1	0	0	Off	Off	0	Lose lock, f.b. clock, reference clock	Regain	NRM	‘LK	1	‘LC	REF not entered during stop; SCM entered during stop only during oscillator startup
								No regain	Stuck	—	—	—	
NRM	1	0	0	Off	On	0	Lose lock, f.b. clock	Regain	NRM	‘LK	1	‘LC	REF mode not entered during stop
								No f.b. clock or lock regain	Stuck	—	—	—	
								Lose reference clock	SCM	0	0	1	Wakeup without lock
NRM	1	0	0	Off	On	1	Lose lock, f.b. clock	Regain f.b. clock	Unstable NRM	0–>‘LK	0–>1	‘LC	REF mode not entered during stop
								No f.b. clock regain	Stuck	—	—	—	
								Lose reference clock	SCM	0	0	1	Wakeup without lock
NRM	1	0	0	On	On	0	—	—	NRM	‘LK	1	‘LC	
								Lose reference clock	SCM	0	0	1	Wakeup without lock
								Lose f.b. clock	REF	0	X	1	Wakeup without lock
								Lose lock	Stuck	—	—	—	
								Lose lock, regain	NRM	0	1	‘LC	
NRM	1	0	0	On	On	1	—	—	NRM	‘LK	1	‘LC	
								Lose reference clock	SCM	0	0	1	Wakeup without lock
								Lose f.b. clock	REF	0	X	1	Wakeup without lock
								Lose lock	Unstable NRM	0	0–>1	‘LC	

Table 12-2. Byte Enables/Byte Write Enable Signal Settings

Transfer Size	Port Size	A1	A0	$\overline{BS3}$	$\overline{BS2}$	$\overline{BS1}$	$\overline{BS0}$
				D[31:24]	D[23:16]	D[15:8]	D[7:0]
Byte	8-bit	0	0	0	1	1	1
		0	1	0	1	1	1
		1	0	0	1	1	1
		1	1	0	1	1	1
	16-bit	0	0	0	1	1	1
		0	1	1	0	1	1
		1	0	0	1	1	1
		1	1	1	0	1	1
	32-bit	0	0	0	1	1	1
		0	1	1	0	1	1
		1	0	1	1	0	1
		1	1	1	1	1	0
Word	8-bit	0	0	0	1	1	1
		0	1	0	1	1	1
		1	0	0	1	1	1
		1	1	0	1	1	1
	16-bit	0	0	0	0	1	1
		1	0	0	0	1	1
	32-bit	0	0	0	0	1	1
		1	0	1	1	0	0
Longword	8-bit	0	0	0	1	1	1
		0	1	0	1	1	1
		1	0	0	1	1	1
		1	1	0	1	1	1
	16-bit	0	0	0	0	1	1
		1	0	0	0	1	1
	32-bit	0	0	0	0	0	0
Line	8-bit	0	0	0	1	1	1
		0	1	0	1	1	1
		1	0	0	1	1	1
		1	1	0	1	1	1
	16-bit	0	0	0	0	1	1
		1	0	0	0	1	1
	32-bit	0	0	0	0	0	0

Table 14-1. MCF5282 Signal Description (continued)

Signal Name	Abbreviation	Function	I/O	Page
Development serial input/Test data	DSI/TDI	Provides single-bit communication for debug module commands (DSI). Provides serial data port for loading JTAG boundary scan, bypass, and instruction registers (TDI).	I	14-30
Development serial output/Test data	DSO/TDO	Provides single-bit communication for debug module responses (DSO). Provides serial data port for outputting JTAG logic data (TDO).	O	14-30
Test clock	TCLK	JTAG test logic clock.	I	14-30
Debug data	DDATA[3:0]	Display captured processor addresses, data, and breakpoint status.	O	14-31
Processor status outputs	PST[3:0]	Indicate core status.	O	14-31
Test Signals				
Test	TEST	Reserved, should be connected to VSS.	I	14-31
Power and Reference Signals				
QADC analog reference	VRH, VRL	High (VRH) and low (VRL) reference potentials for the analog converter.	Ground	14-32
QADC analog supply	VDDA, VSSA	Isolate the QADC analog circuitry from digital power supply noise.	I	14-32
PLL analog supply	VDDPLL, VSSPLL	Isolate the PLL analog circuitry from digital power supply noise.	I	14-32
QADC positive supply	VDDH	Supplies positive power to the ESD structures in the QADC pads.	I	14-32
Flash erase/program power	VPP	Used for Flash stress testing.	I	14-32
Flash array power and ground	VDDF, VSSF	Supply power and ground to Flash array.	I	14-32
Standby power	VSTBY	Provides standby voltage to RAM array if VDD is lost.	I	14-32
Positive supply	VDD	Supplies positive power to the core logic and I/O pads.	I	14-32
Ground	VSS	Negative supply.		14-32

Table 14-2 lists signals in alphabetical order by abbreviated name.

Chapter 19
Programmable Interrupt Timers (PIT0–PIT3)

19.1 Introduction

This chapter describes the operation of the four programmable interrupt timer modules: PIT0–PIT3.

19.1.1 Overview

Each PIT is a 16-bit timer that provides precise interrupts at regular intervals with minimal processor intervention. The timer can count down from the value written in the modulus register or it can be a free-running down-counter.

19.1.2 Block Diagram

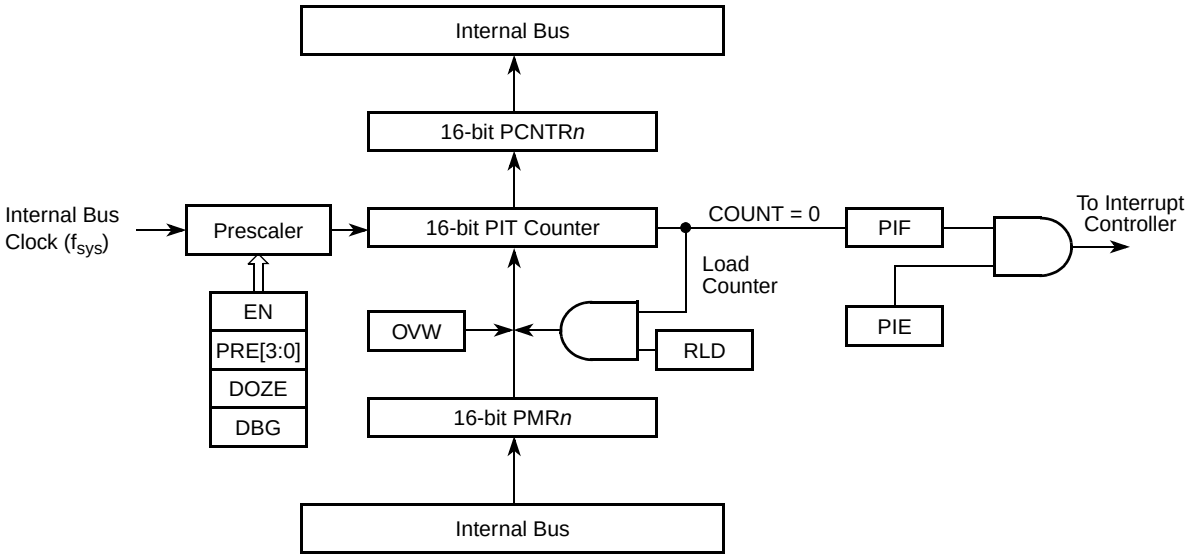

Figure 19-1. PIT Block Diagram

19.1.3 Low-Power Mode Operation

This subsection describes the operation of the PIT modules in low-power modes and debug mode of operation. Low-power modes are described in the power management module, Chapter 7, “Power Management.” Table 19-1 shows the PIT module operation in low-power modes and how it can exit from each mode.

Chapter 24
I²C Interface

24.1 Introduction

This chapter describes the I²C module, clock synchronization, and I²C programming model registers. It also provides extensive programming examples.

24.1.1 Block Diagram

Figure 24-1 is a I²C module block diagram, illustrating the interaction of the registers described in Section 24.2, “Memory Map/Register Definition”.

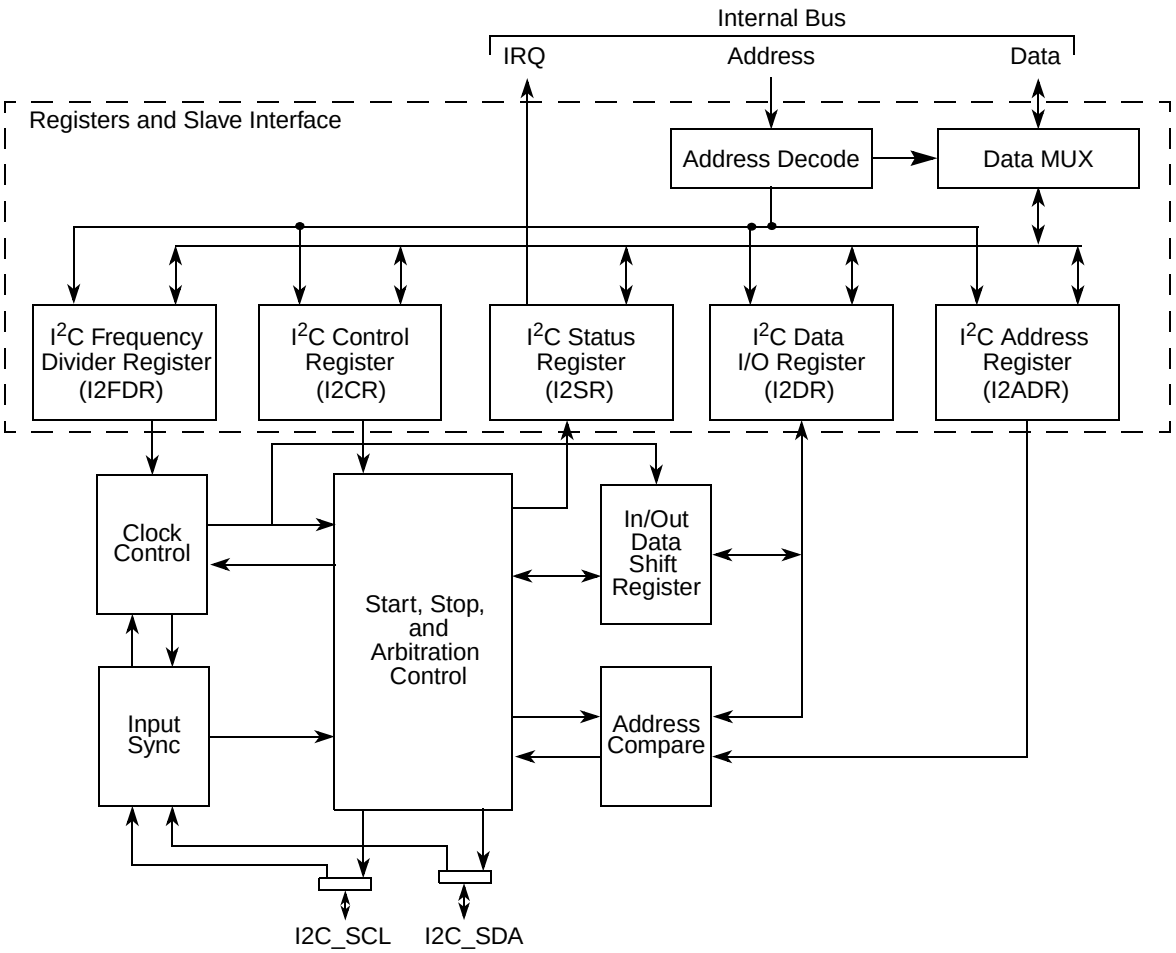

Figure 24-1. I²C Module Block Diagram

IPSBAR Offset: 0x00_0308 (I2CR) Access: User read/write

	7	6	5	4	3	2	1	0
Reset:	0	0	0	0	0	0	0	0

Figure 24-4. I²C Control Register (I2CR)

Table 24-4. I2CR Field Descriptions

Field	Description
7 IEN	I²C enable. Controls the software reset of the entire I²C module. If the module is enabled in the middle of a byte transfer, slave mode ignores the current bus transfer and starts operating when the next START condition is detected. Master mode is not aware that the bus is busy; initiating a start cycle may corrupt the current bus cycle, ultimately causing the current master or the I²C module to lose arbitration, after which bus operation returns to normal. 0 The I²C module is disabled, but registers can be accessed. 1 The I²C module is enabled. This bit must be set before any other I2CR bits have any effect.
6 IIEN	I²C interrupt enable. 0 I²C module interrupts are disabled, but currently pending interrupt condition is not cleared. 1 I²C module interrupts are enabled. An I²C interrupt occurs if I2SR[IIF] is also set.
5 MSTA	Master/slave mode select bit. If the master loses arbitration, MSTA is cleared without generating a STOP signal. 0 Slave mode. Changing MSTA from 1 to 0 generates a STOP and selects slave mode. 1 Master mode. Changing MSTA from 0 to 1 signals a START on the bus and selects master mode.
4 MTX	Transmit/receive mode select bit. Selects the direction of master and slave transfers. 0 Receive 1 Transmit. When the device is addressed as a slave, software must set MTX according to I2SR[SRW]. In master mode, MTX must be set according to the type of transfer required. Therefore, when the MCU addresses a slave device, MTX is always 1.
3 TXAK	Transmit acknowledge enable. Specifies the value driven onto I2C_SDA during acknowledge cycles for master and slave receivers. Writing TXAK applies only when the I²C bus is a receiver. 0 An acknowledge signal is sent to the bus at the ninth clock bit after receiving one byte of data. 1 No acknowledge signal response is sent (acknowledge bit = 1).
2 RSTA	Repeat start. Always read as 0. Attempting a repeat start without bus mastership causes loss of arbitration. 0 No repeat start 1 Generates a repeated START condition.
1	Reserved, must be cleared.

24.2.4 I²C Status Register (I2SR)

I2SR contains bits that indicate transaction direction and status.

IPSBAR Offset: 0x00_030C (I2SR) Access: User read/write

	7	6	5	4	3	2	1	0
R	ICF	IAAS	IBB	IAL	0	SRW	IIF	RXAK
W								
Reset:	1	0	0	0	0	0	0	1

Figure 24-5. I²C Status Register (I2SR)

25.2 The CAN System

A typical CAN system is shown below in Figure 25-2.

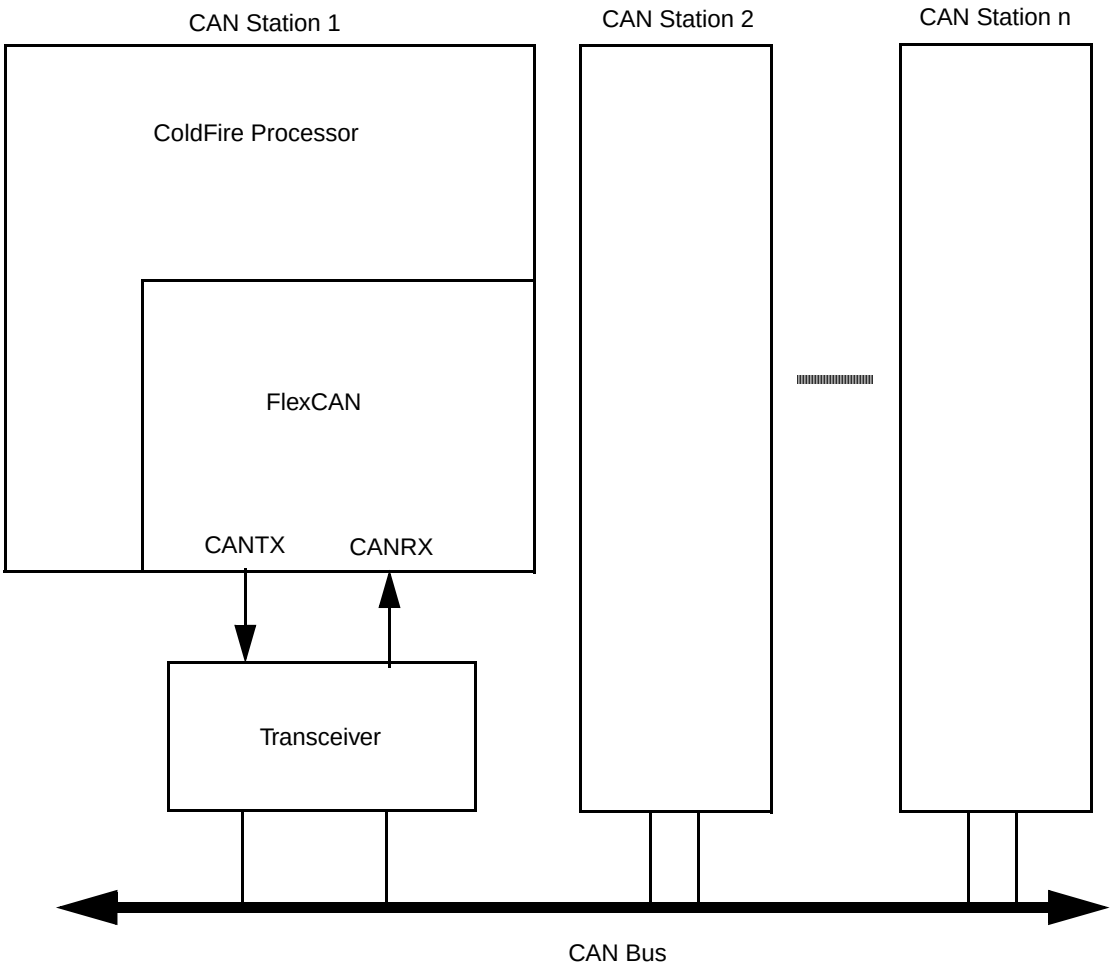

Figure 25-2. Typical CAN system

Each CAN station is connected physically to the CAN bus through a transceiver. The transceiver provides the transmit drive, waveshaping, and receive/compare functions required for communicating on the CAN bus. It can also provide protection against damage to the FlexCAN caused by a defective CAN bus or defective stations.

25.3 Message Buffers

25.3.1 Message Buffer Structure

Figure 25-3 shows the extended (29 bit) ID message buffer structure. Figure 25-4 displays the standard (11 bit) ID message buffer structure.

26.3.2.6 Port E Pin Assignment Register (PEPAR)

The PEPAR controls the pin function of port E.

The PEPAR register is read/write.

	15	14	13	12	11	10	9	8
Field	—	PEPA7	—	PEPA6	—	PEPA5	—	PEPA4
Reset	0	See Note 1	0	See Note 1	0	See Note 1	0	See Note 1
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

	7	6	5	4	3	2	1	0
Field	—	PEPA3	—	PEPA2	PEPA1		PEPA0	
Reset	0	See Note 1	0	See Note 1	See Note 1		See Note 1	
R/W:	R	R/W	R	R/W	R/W		R/W	
Address	IPSBAR + 0x10_0052							

Figure 26-20. Port E Pin Assignment Register (PEPAR)

[1] Reset state determined during reset configuration as shown in Table 26-10.

Table 26-9. PEPAR Field Descriptions

Bits	Name	Description
14	PEPA7	Port E pin assignment 7. This bit configures the port E7 pin for its primary function ($\overline{OE}$) or digital I/O. 1 Port E7 pin configured for primary function ($\overline{OE}$) 0 Port E7 pin configured for digital I/O
12	PEPA6	Port E pin assignment 6. This bit configures the port E6 pin for its primary function ($\overline{TA}$) or digital I/O. 1 Port E6 pin configured for primary function ($\overline{TA}$) 0 Port E6 pin configured for digital I/O
10	PEPA5	Port E pin assignment 5. This bit configures the port E5 pin for its primary function ($\overline{TEA}$) or digital I/O. 1 Port E5 pin configured for primary function ($\overline{TEA}$) 0 Port E5 pin configured for digital I/O
8	PEPA4	Port E pin assignment 4. This bit configures the port E4 pin for its primary function (R/$\overline{W}$) or digital I/O. 1 Port E4 pin configured for primary function (R/$\overline{W}$) 0 Port E4 pin configured for digital I/O

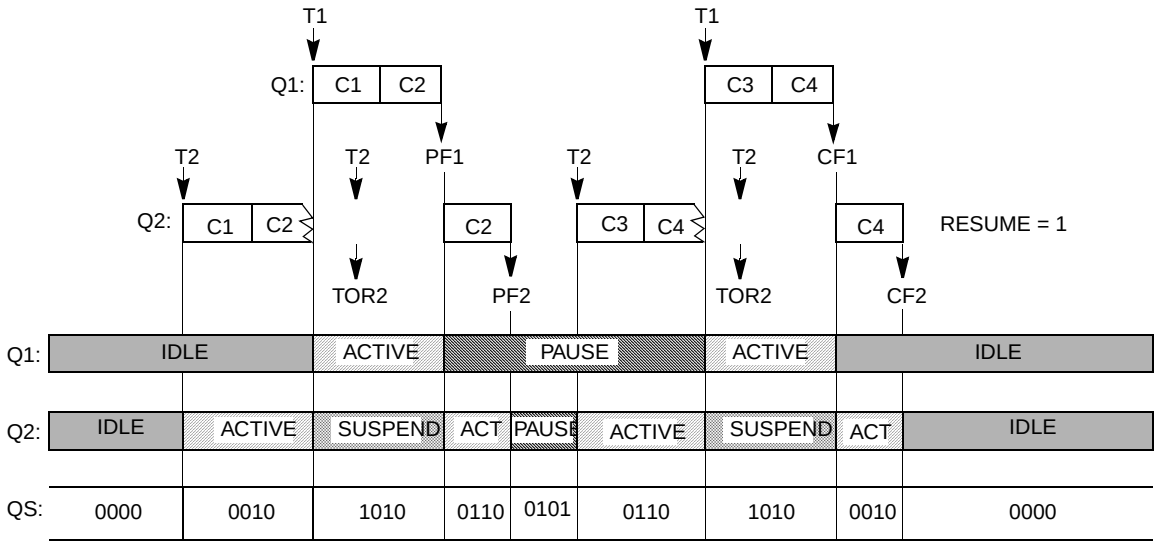

Figure 28-33. CCW Priority Situation 11

The previous situations cover normal overlap conditions that arise with asynchronous trigger events on the two queues. An additional conflict to consider is that the freeze condition can arise while the QADC is actively executing CCWs. The conventional use for the debug mode is for software/hardware debugging. When the CPU enters background debug mode, peripheral modules can cease operation. When freeze is detected, the QADC completes the conversion in progress, unlike the abort that occurs when queue 1 suspends queue 2. After the freeze condition is removed, the QADC continues queue execution with the next CCW in sequence.

Trigger events that occur during freeze are not captured. When a trigger event is pending for queue 2 before freeze begins, that trigger event is remembered when the freeze is passed. Similarly, when freeze occurs while queue 2 is suspended, after freeze, queue 2 resumes execution as soon as queue 1 is finished.

Situations 12 through 19 (Figure 28-34 to Figure 28-41) show examples of all of the freeze situations.

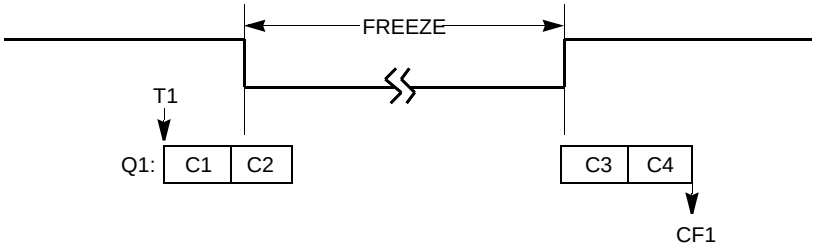

Figure 28-34. CCW Freeze Situation 12

28.9.4 Analog Supply Filtering and Grounding

Two important factors influencing performance in analog integrated circuits are supply filtering and grounding. Generally, digital circuits use bypass capacitors on every V_{DD}/V_{SS} signal pair. This applies to analog subsystems and submodules also. Equally important as bypassing is the distribution of power and ground.

Analog supplies should be isolated from digital supplies as much as possible. This necessity stems from the higher performance requirements often associated with analog circuits. Therefore, deriving an analog supply from a local digital supply is not recommended. However, if for cost reasons digital and analog power are derived from a common regulator, filtering of the analog power is recommended in addition to the bypassing of the supplies already mentioned. For example, an RC low pass filter could be used to isolate the digital and analog supplies when generated by a common regulator. If multiple high precision analog circuits are locally employed (for example, two A/D converters), the analog supplies should be isolated from each other as sharing supplies introduces the potential for interference between analog circuits.

Grounding is the most important factor influencing analog circuit performance in mixed signal systems (or in standalone analog systems). Close attention must be paid not to introduce additional sources of noise into the analog circuitry. Common sources of noise include ground loops, inductive coupling, and combining digital and analog grounds together inappropriately.

The problem of how and when to combine digital and analog grounds arises from the large transients which the digital ground must handle. If the digital ground is not able to handle the large transients, the associated current can return to ground through the analog ground. It is this excess current overflowing into the analog ground which causes performance degradation by developing a differential voltage between the true analog ground and the microcontroller’s ground pins. The end result is that the ground observed by the analog circuit is no longer true ground and thus skews converter performance.

Two similar approaches to improving or eliminating the problems associated with grounding excess transient currents involve star-point ground systems. One approach is to star-point the different grounds at the power supply origin, thus keeping the ground isolated. Refer to Figure 28-49.

Another approach is to star-point the different grounds near the analog ground signal on the microcontroller by using small traces for connecting the non-analog grounds to the analog ground. The small traces are meant only to accommodate dc differences, not ac transients.

NOTE

This star-point scheme still requires adequate grounding for digital and analog subsystems in addition to the star-point ground.

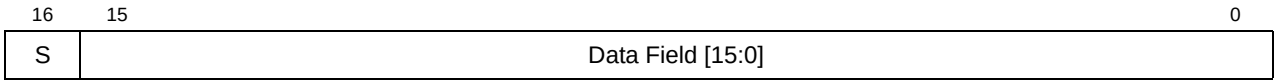

Figure 30-13. Receive BDM Packet

Table 30-15 describes receive BDM packet fields.

Table 30-15. Receive BDM Packet Field Description

Bits	Name	Description
16	S	Status. Indicates the status of CPU-generated messages listed below. The not-ready response can be ignored unless a memory-referencing cycle is in progress. Otherwise, the debug module can accept a new serial transfer after 32 processor clock periods. S DataMessage 0 xxxxValid data transfer 0 0xFFFFStatus OK 1 0x0000Not ready with response; come again 1 0x0001Error—Terminated bus cycle; data invalid 1 0xFFFFIllegal command
15–0	D	Data. Contains the message to be sent from the debug module to the development system. The response message is always a single word, with the data field encoded as shown above.

30.5.2.2 Transmit Packet Format

The basic transmit packet, Figure 30-14, consists of 16 data bits and 1 control bit.

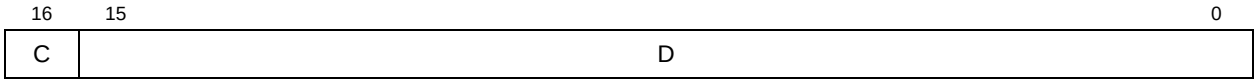

Figure 30-14. Transmit BDM Packet

Table 30-16 describes transmit BDM packet fields.

Table 30-16. Transmit BDM Packet Field Description

Bits	Name	Description
16	C	Control. This bit is reserved. Command and data transfers initiated by the development system should clear C.
15–0	D	Data bits 15–0. Contains the data to be sent from the development system to the debug module.

30.5.3 BDM Command Set

Table 30-17 summarizes the BDM command set. Subsequent paragraphs contain detailed descriptions of each command. Issuing a BDM command when the processor is accessing debug module registers using the WDEBUG instruction causes undefined behavior.

* The timings are also valid for inputs sampled on the negative clock edge.

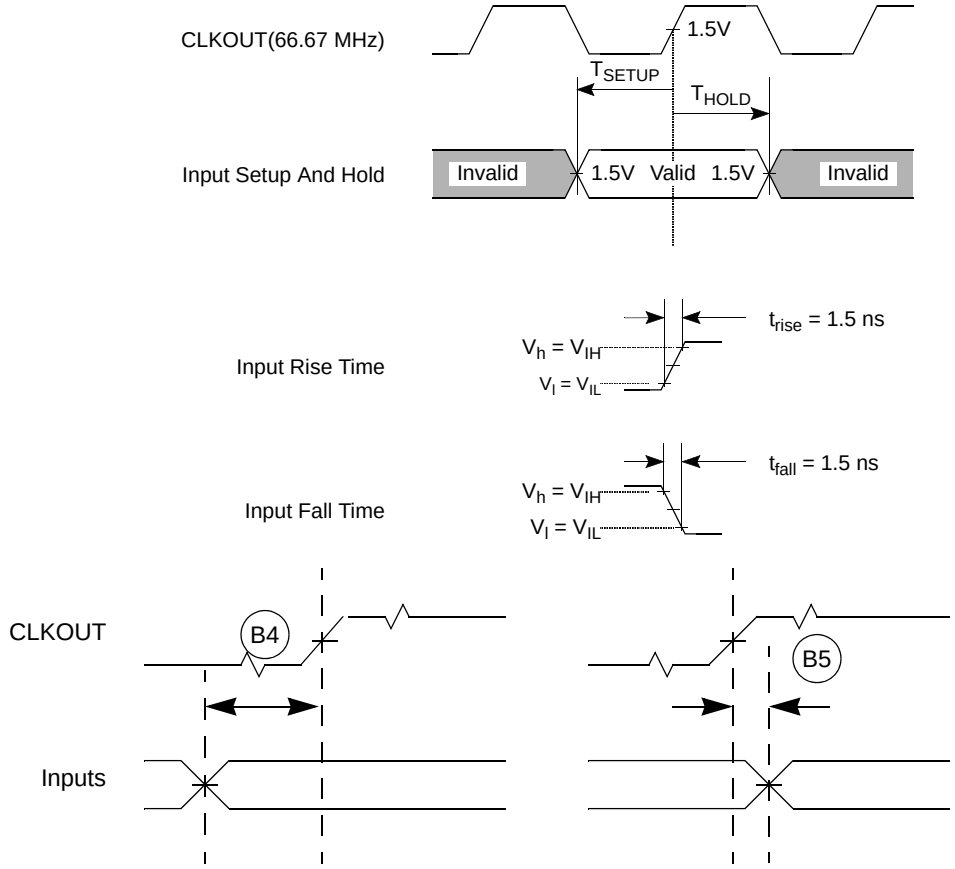

Figure 33-2. General Input Timing Requirements

33.9 Processor Bus Output Timing Specifications

Table 33-14 lists processor bus output timings.

Table 33-15. External Bus Output Timing Specifications

Name	Characteristic	Symbol	Min	Max	Unit
Control Outputs					
B6a	CLKOUT high to chip selects valid [1]	t_{CHCV}	—	$0.5t_{CYC}$ +10	ns
B6b	CLKOUT high to byte enables ($\overline{BS}$[3:0]) valid[2]	t_{CHBV}	—	$0.5t_{CYC}$ +10	ns
B6c	CLKOUT high to output enable ($\overline{OE}$) valid[3]	t_{CHOV}	—	$0.5t_{CYC}$ +10	ns
B7	CLKOUT high to control output ($\overline{BS}$[3:0], $\overline{OE}$) invalid	t_{CHCOI}	$0.5t_{CYC}$ + 2	—	ns
B7a	CLKOUT high to chip selects invalid	t_{CHCI}	$0.5t_{CYC}$ + 2	—	ns
Address and Attribute Outputs					

B.4 Changes Between Rev. 2 and Rev. 2.1

Table B-4. Rev. 2 to Rev. 2.1 Changes

Location	Description
Title Page	Added MCF5280 to “Devices Supported” list on the title page.
Table 33-8/33-9	Deleted reference to “TA=TL to TH”

B.5 Changes Between Rev. 2.1 and Rev. 2.2

Table B-5. Rev. 2.1 to Rev. 2.2 Changes

Location	Description
Chapter 33	Added Power Spec info to Electricals chapter

B.6 Changes Between Rev. 2.2 and Rev. 2.3

Table B-6. Rev. 2.2 to Rev. 2.3 Changes

Location	Description
Figure 4-2/4-6	Changed bit 23 from DIDI to DISI
Table 4-6/4-9	Under ‘Configuration’ for ‘Instruction Cache’ the ‘Operation’ entry changed to “Invalidate 2 KByte data cache”
Table 4-6/4-9	Under ‘Configuration’ for ‘Data Cache’ the ‘Operation’ entry changed to “Invalidate 2 KByte instruction cache”
Figure 6-3/6-6	Changed bit 8 to write-only instead of read/write
Table 6-10/6-15	Removed “selected by BKSL[1:0]” as these are internal signal names not necessary for end-user.
10.3.2/10-8	Added note after register descriptions: ‘If an interrupt source is being masked in the interrupt controller mask register (IMR) or a module’s interrupt mask register while the interrupt mask in the status register (SR[I]) is set to a value lower than the interrupt’s level, a spurious interrupt may occur. This is because by the time the status register acknowledges this interrupt, the interrupt has been masked. A spurious interrupt is generated because the CPU cannot determine the interrupt source. To avoid this situation for interrupts sources with levels 1-6, first write a higher level interrupt mask to the status register, before setting the mask in the IMR or the module’s interrupt mask register. After the mask is set, return the interrupt mask in the status register to its previous value. Since level seven interrupts cannot be disabled in the status register prior to masking, use of the IMR or module interrupt mask registers to disable level seven interrupts is not recommended.
Table 17-2/17-5	In PALR/PAUR entry, deleted “(only needed for full duplex flow control)”
Figure 17-23/17-39	Changed FRSR to read/write instead of read-only
25.4.10/25-16	Changed CANICR to ICR*n*
Table 25-17/25-29	Added the following information to BITERR and ACKERR descriptions: “To clear this bit, first read it as a one, then write it as a one. Writing zero has no effect.”
Table 25-17/25-30	Changed bit ordering: ERRINT should be bit 2 and BOFFINT should be bit 1.

Table B-6. Rev. 2.2 to Rev. 2.3 Changes (continued)

Location	Description
Table 25-19/25-32	Changed BUF*n*I field description from “To clear an interrupt flag, first read the flag as a one, then write it as a **zero**” to “To clear an interrupt flag, first read the flag as a one, then write it as a **one**.”
Chapter 33	Updated power consumption tables.

B.7 Changes Between Rev. 2.3 and Rev. 3

Table B-7. Rev. 2.3 to Rev. 3 Changes

Location	Description
Throughout	Added MCF5214 and MCF5216 to list of the devices supported in this document. These two devices are the same as the MCF5282 except they do not have an FEC and have a rated frequency of 66 MHz. Changed title of document.
Preface	Moved revision history to this appendix
Table 2-1/Page 2-4	Remove last sentence in C bit field description.
Table 2-3/Page 2-7	Change PC's Written with MOVEC entry to “No”.
Section 2.5/Page 2-8	Change last bullet to “Use of separate system stack pointers for user and supervisor modes”
Section 2.5/Page 2-9	Change last sentence in fourth paragraph (step 2) to “The IACK cycle is mapped to special locations within the interrupt controller's address space with the interrupt level encoded in the address."
Figure 3-6/Page 3-8	Add minus sign to the exponent so that it is “–(i + 1 – N)”.
Table 4-3/Page 4-5	Change reset value of ACR0, ACR1 to “See Section” since some of the bits are undefined after reset.
Figure 4-2/Page 4-6	Change CACR fields to R/W, since they may be read via the debug module.
Table 4-5/Page 4-8	For split instruction/data cache entry, swap text in parantheses in the description field. Instruction cache uses the upper half of the arrays, while data cache uses the lower half.
Figure 4-3/Page 4-9	Change reset value of ACR: Bits 31-16, 14-13, 6-5, and 2 are undefined, and other bits are cleared. Change ACR fields to R/W, since they may be read via the debug module.
Section 4.4.2.2/Page 4-9	Change note to: **NOTE** Peripheral (IPSBAR) space should not be cached. The combination of the CACR defaults and the two ACR*n* registers must define the non-cacheable attribute for this address space.
Figure 5-1/Page 5-2	Change RAMBAR fields to R/W, since they may be read via the debug module.
Table 5-1/Page 5-2	The PRI1/PRI2 text description does not match the table below it. It should be: “If a bit is set, CPU has priority. If a bit is cleared, DMA has priority.”
Figure 6-3/Page 6-6	Changed FLASHBAR[WP] to read-only.
Table 6-2/Page 6-7	Changed bit description of FLASHBAR[WP] to read-only and that this bit is always set.