SECTION 7
SYSTEM INTEGRATION MODULE

7.1 INTRODUCTION

This subsection details the operation and programming model of the System Integration Module (SIM) registers, including the interrupt controller and system-protection functions for the MCF5206. The SIM provides overall control of the internal and external buses and serves as the interface between the ColdFire core processor and the internal peripherals or external devices.

7.1.1 Features

The following list summarizes the key SIM features:

- Module Base Address Register (MBAR)
  - — Base address location of all internal peripherals and SIM resources
  - — Address space masking to internal peripherals and SIM resources
- Interrupt Controller
  - — Programmable interrupt level (1-7) for internal peripheral interrupts
  - — Programmable priority level (0-3) within each interrupt level
  - — Three external interrupts - programmable as individual Interrupt Requests or as Interrupt Priority-Level signals
- System Protection
  - — Reset status to indicate cause of last reset
  - — Bus monitor and Spurious Interrupt Monitor
  - — Software Watchdog Timer

7.2 SIM OPERATION

7.2.1 Module Base Address Register (MBAR)

The MBAR determines the base address of all internal peripherals as well as the definition of which types of accesses are allowed for these registers.

The MBAR is a 32-bit write-only supervisor control register that physically resides in the SIM. It is accessed in the CPU address space $C0F via the MOVEC instruction. (Refer to the *ColdFire Programmer’s Reference Manual* for use of MOVEC instruction). The MBAR can be read when in Debug mode using background debug commands.

At system reset, the MBAR valid bit is cleared to prevent incorrect references to resources before the MBAR is written. The remainder of the MBAR bits are uninitialized. To access the internal peripherals, you should write MBAR with the appropriate base address and set the valid bit after system reset.

All internal peripheral registers occupy a single relocatable memory block along 1-kbyte boundaries. If the MBAR valid bit is set, the base address field is compared to the upper 22 bits of the full 32-bit internal address to determine if an internal peripheral is being accessed. The MBAR masks specific address spaces using the address space fields. Any attempt to access a masked address space will result in an access being generated on the external bus.

7.2.2 Bus Time-Out Monitor

The bus monitor ensures that each external cycle terminates within a programmed period of time. It continually checks the external bus for termination and asserts the internal transfer error acknowledge that results in an access fault exception if the response time is greater than the programmed bus monitor time. If a transfer is in progress while $\overline{\text{TEA}}$ is asserted, the transfer will be aborted and the exception will occur.

You can program the bus monitor time to 128, 256, 512, or 1024 clock cycles using the bus monitor timing bits in the system protection control register (SYPCR). The value you select should be larger than the longest possible response time of the slowest peripheral of the system. The bus time-out monitor begins counting on the clock cycle $\overline{\text{TS}}$ is asserted and stops counting on the clock after the assertion of the final transfer termination signal (i.e., for a line transfer to a 32-bit port, the bus time-out monitor will start counting on the clock $\overline{\text{TS}}$ is asserted and will stop counting after $\overline{\text{TA}}$ and/or $\overline{\text{ATA}}$ have been asserted a total of four times).

The bus monitor enable bit in the SYPCR enables the bus monitor for external bus cycles. The bus monitor cannot be disabled for internal bus cycles to internal peripherals. Once the SYPCR is written using the MOVEC instruction, the state of the bus time-out monitor cannot be changed—the SYPCR may be written only once. Refer to subsection **6.3.2.7 System Protection Control Register** for programming information.

7.2.3 Spurious Interrupt Monitor

The SIM will automatically generate the spurious interrupt vector number 24 ($18), which causes the ColdFire core processor to terminate the cycle with a spurious interrupt exception if:

- An external device responds to an interrupt acknowledge cycle by asserting $\overline{\text{TEA}}$
- No interrupt is pending for the interrupt level being acknowledged when the interrupt acknowledge cycle is generated.

NOTE

> If an external device does not respond to an interrupt acknowledge cycle by asserting $\overline{\mathrm{TA}}$ or $\overline{\mathrm{ATA}}$, an access error exception will be generated, not a spurious interrupt exception. To generate a spurious interrupt exception, $\overline{\mathrm{TEA}}$ would have to be generated.

7.2.4 Software Watchdog Timer

The software watchdog timer (SWT) prevents system lockup in case the software becomes trapped in loops with no controlled exit. The SWT can be enabled via the software watchdog enable bit in the SYPCR. If enabled, the SWT requires a special service sequence execution on a periodic basis. If this periodic servicing action does not occur, the SWT times out and results in a hardware reset or a level 7 interrupt, as programmed by the software watchdog reset/interrupt select bit in the SYPCR. If the SWT times out and is programmed to generate a hardware reset, an internal reset will be generated and the software watchdog timer reset bit in the reset status register will be set. Additionally, if the RTS2/$\overline{\mathrm{RSTO}}$ pin is programmed for $\overline{\mathrm{RSTO}}$, $\overline{\mathrm{RSTO}}$ will be asserted for 32 CLK cycles. Refer to subsection **7.3.2.10 Pin Assignment Register (PAR)** for more information on programming.

The 8-bit interrupt vector for the SWT interrupt is stored in the software watchdog interrupt vector register (SWIVR). The software watchdog prescalar (SWP) and software watchdog timing (SWT) bits in SYPCR determine the SWT time-out period.

The SWT service sequence consists of these two steps: write $55 to SWSR, and write $AA to the SWSR. Both writes must occur in the order listed prior to the SWT time-out, but any number of instructions or accesses to the SWSR can be executed between the two writes. This order allows interrupts and exceptions to occur, if necessary, between the two writes.

NOTE

> If the SYSPCR is programmed for the SWT to generate an interrupt and the SWT times out, the SWT must be serviced in the interrupt handler routine by writing the $55, $AA sequence to the SWSR.

7.2.5 Interrupt Controller

The SIM provides a centralized interrupt controller for all MCF5206 interrupt sources, including:

- External Interrupts ($\overline{\mathrm{IPL/IRQ}}$)
- Software watchdog timer
- Timer modules
- MBUS (I^{2}C) module
- UART modules

All interrupt inputs are level sensitive. An interrupt request must be held valid for at least two consecutive CLK periods to be considered a valid input. The three external interrupt inputs can be programmed to be three individual interrupt inputs (at level 1, 4, and 7) or encoded interrupt priority levels.

NOTE

> If the external interrupt inputs are programmed to individual interrupt requests (at level 1, 4, and 7), the interrupt request must be asserted until the MCF5206 acknowledges the interrupt (by generating an interrupt acknowledge cycle) to guarantee that the interrupt is recognized.

If the external interrupt inputs are programmed to be interrupt priority levels, the interrupt request must maintain the interrupt request level or a higher priority level request until the MCF5206 acknowledges the interrupt to guarantee that the interrupt is recognized.

When the external interrupts are programmed to be individual $\overline{\text{IRQ}}$ interrupts in the Pin Assignment Register (PAR), the interrupt level bits in the appropriate ICRs of the external interrupts are not user-programmable and are always set to 1, 4, and 7. The value of the interrupt level bits in the ICRs for the external interrupts cannot be changed, even by a write to the register.

If the external interrupts are programmed to be encoded interrupt priority levels, the interrupt level is that indicated as shown in Table 7-1.

Table 7-1. Interrupt Levels for Encoded External Interrupts

$\overline{\text{IPL2/IRQ7}}$	$\overline{\text{IPL1/IRQ4}}$	$\overline{\text{IPL0/IRQ1}}$	INTERRUPT LEVEL $\overline{\text{INDICATED}}$
0	0	0	7
0	0	1	6
0	1	0	5
0	1	1	4
1	0	0	3
1	0	1	2
1	1	0	1
1	1	1	No Interrupt

Although the interrupt levels of the external interrupts are fixed, customers can program the interrupt priorities of the external interrupts to any value using the IP (IP1, IP0) bits in the corresponding interrupt control registers (ICR7 - ICR1). You can program the autovector bits in the interrupt control registers and you should program them to 1 when autovector generation is preferred.

NOTE

> When an autovectored interrupt occurs, the interrupt will be serviced internally. No external IACK cycle will occur.

The SWT has a fixed interrupt level (level 7). The interrupt priority of the SWT can be programmed to any value using the IP (IP1, IP0) bits in the SWT interrupt control register, ICR8. You cannot program the SWT to generate an autovector.The autovector bit in ICR8 is reserved and is set to zero. You cannot change it. The value of the software watchdog interrupt vector register will always be used as the interrupt vector number for a SWT interrupt.

You can program the timer modules interrupt levels and priorities using the interrupt level (IL2 - IL0) and interrupt priority (IP1, IP0) bits in the appropriate interrupt control registers (ICR9, ICR10). The timer peripherals cannot provide interrupt vectors. Thus, autovector bits in ICR9 and ICR10 are set to 1. You cannot change this value. This generates autovectors in response to all timer interrupts.

You can also program the MBUS module interrupt level and priority using the interrupt level (IL2 - IL0) and interrupt priority (IP1, IP0) bits in the MBUS interrupt control register, ICR11. You cannot program the MBUS module to provide an interrupt vector. Thus, the autovector bit in ICR11 is set to 1. You cannot change this value. This generates an autovector in response to an MBUS interrupt.

You can program the UART module interrupt levels and priorities using the interrupt level (IL2 - IL0) and interrupt priority (IP1, IP0) bits in the appropriate interrupt control registers (ICR12, ICR13). In addition, you can program the autovector bits in ICR12 and ICR13. If the autovector bit is set to 0, you must program the interrupt vector register in each UART module to the preferred vector number.

The interrupt controller monitors and masks individual interrupt inputs and outputs the highest priority unmasked pending interrupt to the ColdFire core. Each interrupt input has a mask bit in the interrupt mask register (IMR) and a pending bit in the interrupt pending register (IPR). The pending bits for all internal interrupts in the interrupt pending register are set the CLK cycle after the interrupt is asserted whether or not the interrupt is masked. If you program the external interrupt inputs as individual interrupt inputs, the pending bits in the interrupt pending register are set to the CLK cycle after the interrupt is asserted and internally synchronized.

If you program the external interrupt inputs to indicate interrupt priority levels, the interrupt pending bits are set and cleared as follows:

1. EINT[7:1] will be set the CLK cycle after the interrupt level has been internally synchronized and indicated the same valid level for two consecutive CLK cycles.
2. The interrupt pending bits EINT[7:1] will remain set if the external interrupt level remains the same or increases in priority.
3. The interrupt pending bits EINT[7:1] will be cleared if the external interrupt level decreases in priority or if an interrupt acknowledge cycle is completed for an external interrupt that is pending but is not the current level being driven onto the external interrupt priority level signals ($\overline{\text{IPLx}}$/$\overline{\text{IRQx}}$).

For example, if you program the external interrupts to indicate interrupt priority levels and assert to indicate a level 3 interrupt, and then assert to indicate a level 6 interrupt, both

EINT3 and EINT6 bits in the interrupt pending register will be set. This indicates that both an external level 6 and an external level 3 interrupt is pending.If the level 3 interrupt is now acknowledged (by completing an interrupt acknowledge cycle for a level 3 interrupt), EINT3 will be cleared and only EINT6 will remain set. If the interrupt priority level is now changed to indicate a level 1 interrupt, EINT6 will be cleared and EINT1 will be set, indicating that only an external interrupt level 1 is currently pending. Running an interrupt-acknowledge cycle for the level 6 interrupt returns the spurious interrupt vector as the level 6 interrupt is no longer pending. EINT1 will remain pending until the interrupt priority level signals change. For timing diagrams, refer to the electrical section.

You can assign as many as four interrupts to the same interrupt level, but you must assign unique interrupt priorities. The interrupt controller uses the interrupt priorities during an interrupt acknowledge cycle to determine which interrupt is being acknowledged. The interrupt priority bits determine the appropriate interrupt being acknowledged when multiple interrupts are assigned to the same level and are pending when the interrupt-acknowledge cycle is generated.

NOTE

> You should not program interrupts to have the same level and priority. Interrupts can have the same level but different priorities. All level and priority combinations must be unique.

If an external interrupt request is being acknowledged and the AVEC bit in the corresponding ICR is not set, an external interrupt acknowledge cycle occurs. During an external interrupt acknowledge cycle, TT[1:0] and A[27:5] are driven high; A[4:2] are set to the interrupt level being acknowledged; and A[1:0] are driven low. Additionally, ATM is asserted when $\overline{TS}$ is asserted and ATM is negated when $\overline{TS}$ is negated. For nonautovector responses, the external device places the vector number on D[31:24]. For autovector responses, the autovector is generated internally and no external interrupt acknowledge cycle is run.

An interrupt request from the SWT does not require an external interrupt acknowledge cycle because SWIVR stores its interrupt vector number.

If an internal peripheral interrupt source is being acknowledged and the AVEC bit in the corresponding ICR is cleared, an internal interrupt-acknowledge cycle occurs with the internal peripheral supplying the interrupt vector number. If the corresponding AVEC bit is set, an internal interrupt-acknowledge cycle will be run but the SIM will internally generate an autovector. No external interrupt acknowledge cycle will be run.

7.3 PROGRAMMING MODEL

7.3.1 SIM Registers Memory Map

Table 7-1 shows the memory map of all the SIM registers. The internal registers in the SIM are memory-mapped registers offset from the MBAR address pointer. The following list addresses several key notes regarding the programming model table:

- The Module Base Address Register can only be accessed in supervisor mode using the MOVEC instruction with an Rc value of $C0F.
- Underlined registers are status or event registers.
- Addresses not assigned to a register and undefined register bits are reserved for future expansion. Write accesses to these reserved address spaces and reserved register bits have no effect; read accesses will return zeros.
- The reset value column indicates the register initial value at reset. Certain registers may be uninitialized at reset.
- The access column indicates if the corresponding register allows both read/write functionality (R/W), read-only functionality (R), or write-only functionality (W). An attempted read access to a write-only register will return zeros. An attempted write access to a read-only register will be ignored and no write will occur.

Table 7-2. Memory Map of SIM Registers

ADDRESS	NAME	WIDTH	DESCRIPTION	RESET VALUE	ACCESS
MOVEC with $C0F	MBAR	32	Module Base Address Register	uninitialized (except V=0)	W
MBAR + $003	SIMR	8	SIM Configuration Register	$C0	R/W
MBAR + $014	ICR1	8	Interrupt Control Register 1 - External IRQ1/IPL1	$04	R/W
MBAR + $015	ICR2	8	Interrupt Control Register 2 - External IPL2	$08	R/W
MBAR + $016	ICR3	8	Interrupt Control Register 3 - External IPL3	$0C	R/W
MBAR + $017	ICR4	8	Interrupt Control Register 4 - External IRQ4/IPL4	$10	R/W
MBAR + $018	ICR5	8	Interrupt Control Register 5 - External IPL5	$14	R/W
MBAR + $019	ICR6	8	Interrupt Control Register 6 - External IPL6	$18	R/W
MBAR + $01A	ICR7	8	Interrupt Control Register 7 - External IRQ7/IPL7	$1C	R/W
MBAR + $01B	ICR8	8	Interrupt Control Register 8 - SWT	$1C	R/W
MBAR + $01C	ICR9	8	Interrupt Control Register 9 - Timer 1 Interrupt	$80	R/W
MBAR + $01D	ICR10	8	Interrupt Control Register 10 - Timer 2 Interrupt	$80	R/W
MBAR + $01E	ICR11	8	Interrupt Control Register 11 - MBUS Interrupt	$80	R/W
MBAR + $01F	ICR12	8	Interrupt Control Register 12 - UART 1 Interrupt	$00	R/W
MBAR + $020	ICR13	8	Interrupt Control Register 13 - UART2 Interrupt	$00	R/W
MBAR + $036	IMR	16	Interrupt Mask Register	$3FFE	R/W
MBAR + $03A	IPR	16	Interrupt Pending Register	$0000	R
MBAR + $040	RSR	8	Reset Status Register	$80 or $20	R/W
MBAR + $041	SYPCR	8	System Protection Control Register	$00	R/W
MBAR + $042	SWIVR	8	Software Watchdog Interrupt Vector Register	$0F	R/W
MBAR + $043	SWSR	8	Software Watchdog Service Register	uninitialized	W
MBAR + $0CB	PAR	8	Pin Assignment Register	$00	R/W

7.3.2 SIM Registers

7.3.2.1 MODULE BASE ADDRESS REGISTER (MBAR). The MBAR determines the base address location of all internal module resources such as registers as well as the definition of the types of accesses that are allowed for these resources.

The MBAR is a 32-bit write-only supervisor control register that physically resides in the SIM. It is accessed in the CPU address space via the MOVEC instruction with an Rc encoding of $C0F. The MBAR can be read and written to when in Background Debug mode (BDM). At system reset, the V-bit is cleared and the remainder of the MBAR bits are uninitialized. To access the internal modules, MBAR should be written with the appropriate base address after system reset.

Module Base Address Register(MBAR)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
RESET:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA15	BA14	BA13	BA12	BA11	BA10	-	-	-	-	-	SC	SD	UC	UD	V
RESET:	-	-	-	-	-	-	0	0	0	0	0	-	-	-	-	0

BA[31:10] - Base Address

This field defines the base address location of all internal modules and SIM resources.

SC, SD, UC, UD - Address Space Masks

This field enables or disables specific address spaces, placing the internal modules in a specific address space. If an address space is disabled, an access to the register location in that address space becomes an external bus access, and the module resource is not accessed. The address space mask bits are

SC = Supervisor code address space mask
SD = Supervisor data address space mask
UC = User code address space mask
UD = User data address space mask

For each address space bit:

0= An access to the internal module can occur for this address space.
1= Disable this address space from the internal module selection. If this address space is used, no access will occur to the internal peripheral; instead, an external bus cycle will be generated.

V - Valid

This bit indicates when the contents of the MBAR are valid. The base address value is not used, and all internal modules are not accessible until the V-bit is set. An external bus cycle is generated if the base address field matches the internal core address, and the V-bit is clear.

0 = Contents of MBAR are not valid.
1 = Contents of MBAR are valid.

7.3.2.2 SIM CONFIGURATION REGISTER (SIMR). The SIMR has customer-programmable bits to control the following:

- Operation of software watchdog timer when the internal freeze signal is asserted
- Operation of bus time-out monitor when the internal freeze signal is asserted
- Operation of bus lock.

The internal freeze signal is asserted when the core processor has entered into Background Debug mode (BDM) for software development purposes.

The SIMR is an 8-bit read-write register. At system reset, FRZ1 and FRZ0 are set to 1 and BL is set to 0.

SIM Configuration Register(SIMR) Address MBAR + $03

	7	6	5	4	3	2	1	0
	FRZ1	FRZ0	-	-	-	-	-	BL
RESET:	1	1	0	0	0	0	0	0
R/W								

FRZ1 - Freeze Software Watchdog Timer Enable

0 = When the internal freeze signal is asserted, the software watchdog timer continues to run.
1 = When the internal freeze signal is asserted, the software watchdog timer is disabled.

FRZ0 - Freeze Bus Monitor Enable

0 = When the internal freeze signal is asserted, the bus monitor continues to run.
1 = When the internal freeze signal is asserted, the bus monitor is disabled.

BL - Bus Lock Enable

Bus lock enable lets customers control the assertion and negation of the bus driven ($\overline{BD}$) signal. Refer to the Bus Operation section for more information.

0 = Bus Driven ($\overline{BD}$) signal will be negated by the MCF5206 and the bus will be released when bus grant ($\overline{BG}$) is negated and the current bus cycle is completed.
1 = Once bus grant ($\overline{BG}$) is asserted, the bus driven ($\overline{BD}$) signal will be asserted and can not be cleared until the BL bit is cleared.

7.3.2.3 INTERRUPT CONTROL REGISTER (ICR). The ICR contains the interrupt and priority levels assigned to each interrupt input. There will be one ICR for each interrupt input. Table 7-3 indicates the interrupt control register assigned to each interrupt input, the interrupt control register reset value, and the value of the interrupt level assigned. Each interrupt input must have a unique interrupt level and interrupt priority combination.

NOTE

> The interrupt control registers do not have valid interrupt level/interrupt priority combinations out of reset. You must program all interrupt control registers before programming the interrupt mask register (IMR). If you program the external interrupt inputs to be individual interrupts at level 1, 4 and 7, then ICR2, ICR3, ICR5 and ICR6 are not used.

Table 7-3. Interrupt Control Register Assignments

INTERRUPT SOURCE	INTERRUPT CONTROL REGISTER (ICR)	ICR RESET VALUE	ICR IINTERRUPT LEVEL (IL2 - IL0) VALUE
External Interrupt Request 1 External Interrupt Priority Level 1	ICR1	$04	$1
External Interrupt Priority Level 2	ICR2	$08	$2
External Interrupt Priority Level 3	ICR3	$0C	$3
External Interrupt Request 4 External Interrupt Priority Level 4	ICR4	$10	$4
External Interrupt Priority Level 5	ICR5	$14	$5
External Interrupt Priority Level 6	ICR6	$18	$6
External Interrupt Request 7 External Interrupt Priority Level 7	ICR7	$1C	$7
Software Watchdog Timer	ICR8	$1C	$7
Timer 1	ICR9	$80	User Programmable
Timer 2	ICR10	$80	User Programmable
MBUS (I^2C)	ICR11	$80	User Programmable
UART 1	ICR12	$00	User Programmable
UART 2	ICR13	$00	User Programmable

The ICRs are 8-bit read-write registers. See Table 7-3 for the reset values of each ICR.

Interrupt Control Register(ICR)

7	6	5	4	3	2	1	0
AVEC	-	-	IL2	IL1	IL0	IP1	IP0

R/W

AVEC - Autovector Enable

This bit determines if the interrupt acknowledge cycle for the interrupt level indicated in IL2-IL0 for each interrupt input requires an autovector response. If this bit is set, a vector number will be internally generated, which is the sum of the interrupt level, IL2-IL0, plus 24. Seven distinct autovectors can be used corresponding to the 7 levels of interrupt. If this bit is clear, the external device or internal module must return the vector number during an interrupt acknowledge cycle.

NOTE

For the SWT, the corresponding AVEC is a reserved bit and set to zero, disabling autovector generation in response to SWT generated interrupts. The SWT returns the interrupt vector in SWIVR. The AVEC bits in the interrupt control registers for the timers and MBUS peripherals are reserved and are always set to 1. The autovector value will be generated for each of these interrupts.

0 = Interrupting source returns vector number during the interrupt-acknowledge cycle.
1 = SIM internally generates vector number during the interrupt-acknowledge cycle.

IL[2:0] - Interrupt Level

These bits indicate the interrupt level assigned to each interrupt input. Level 7 is the highest priority, level 1 is the lowest, and level 0 indicates that no interrupt is requested. For external interrupts and SWT, the corresponding IL2-IL0 are reserved bits. If the ICRs are programmed to have nonunique interrupt level and priority combination, unpredictable results could occur.

NOTE

You should not program interrupts with the same level and priority. Interrupts can have the same level but different priorities. All level and priority combinations must be unique.

IP[1:0]- Interrupt Priority

These bits indicate the priority within an interrupt level assigned to each interrupt. You can assign as many as four interrupts to the same interrupt level as long as they have unique interrupt priorities. IP1-IP0 = 3 is the highest priority, and IP1-IP0 = 0 is the lowest priority for a given interrupt level. If you program the ICRs to have nonunique interrupt level and priority combination, unpredictable results could occur.

7.3.2.4 INTERRUPT MASK REGISTER (IMR). Each bit in the IMR corresponds to an interrupt source. Table 7-4 indicates which mask bit is assigned to each of the interrupt inputs.

You can mask an interrupt by setting the corresponding bit in the IMR and enable an interrupt by clearing the corresponding bit in the IMR. When a masked interrupt occurs, thecorresponding bit in the IPR is still set, regardless of the setting of the IMR bit, but no interrupt request is passed to the core processor.

Table 7-4. Interrupt Mask Register Bit Assignments

INTERRUPT SOURCE	INTERRUPT MASK REGISTER BIT LOCATION
External Interrupt Request 1 External Interrupt Priority Level 1	1
External Interrupt Priority Level 2	2
External Interrupt Priority Level 3	3

Table 7-4. Interrupt Mask Register Bit Assignments (Continued)

INTERRUPT SOURCE	INTERRUPT MASK REGISTER BIT LOCATION
External Interrupt Request 4 External Interrupt Priority Level 4	4
External Interrupt Priority Level 5	5
External Interrupt Priority Level 6	6
External Interrupt Request 7 External Interrupt Priority Level 7	7
Software Watchdog Timer	8
Timer 1	9
Timer 2	10
MBUS (I^{2}C)	11
UART 1	12
UART 2	13

The IMR is a 16-bit read/write register. At system reset, all nonreserved bits are initialized to one.

Interrupt Mask Register(IMR)

Address MBAR + $36

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UART2	UART1	MBUS	TIMER2	TIMER1	SWT	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	-
RESET:	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0
R/W																

7.3.2.5 INTERRUPT-PENDING REGISTER (IPR). Each bit in the IPR corresponds to an interrupt source. Table 7-5 indicates which pending bit is assigned to each of the interrupt inputs.

Table 7-5. Interrupt Pending Register Bit Assignments

INTERRUPT SOURCE	IINTERRUPT PENDING REGISTER BIT LOCATION
External Interrupt Request 1 External Interrupt Priority Level 1	1
External Interrupt Priority Level 2	2
External Interrupt Priority Level 3	3
External Interrupt Request 4 External Interrupt Priority Level 4	4
External Interrupt Priority Level 5	5
External Interrupt Priority Level 6	6
External Interrupt Request 7 External Interrupt Priority Level 7	7
Software Watchdog Timer	8
Timer 1	9
Timer 2	10
MBUS (I2C)	11
UART 1	12
UART 2	13

When an interrupt is received, the interrupt controller sets the corresponding bit in the IPR. For internal peripherals the corresponding bit in the IPR will be set the CLK cycle after the internal interrupt is asserted and will be cleared when the internal interrupt is cleared. If the external interrupts are programmed to be individual interrupts, the corresponding EINT bit will be set the CLK cycle after the external interrupt is internally synchronized and will be cleared when the external interrupt is cleared. If the external interrupts are programmed to be encoded interrupt priority interrupts, the IPR bit will be set the CLK after the external interrupt is internally synchronized and has been present for two CLK cycles. The IPR bit will be cleared if

- The encoded interrupt priority level being driven on interrupt pins decreases in priority
- An interrupt acknowledge cycle is completed for an external interrupt that is not the external interrupt level indicated on the external interrupt priority level signals.

The IPR bit will be cleared at the end of the interrupt acknowledge cycle. You cannot write to the IPR to clear any of the IPR bits.

An active interrupt request appears as a set bit in the IPR, regardless of the setting of the corresponding mask bit in the IMR.

The IPR is a 16-bit read-only register. At system reset, all bits are initialized to zero.

Interrupt Pending Register(IPR) Address MBAR + $3a

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UART2	UART1	MBUS	TIMER2	TIMER1	SWT	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	-
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Read Only

7.3.2.6 RESET STATUS REGISTER (RSR). The RSR contains a bit for each reset source to the SIM. A set bit indicates the last type of reset that occurred. The RSR is updated by the reset control logic when the reset is complete. Only one bit will be set at any one time in the RSR. You can clear this register by writing a one to that bit location; writing a zero has no effect.

The RSR is an 8-bit read-write register.

Reset Status Register(RSR) Address MBAR + $40

	7	6	5	4	3	2	1	0
	HRST	-	SWTR	-	-	-	-	-
RESET:	1/0	0	1/0	0	0	0	0	0

R/W

HRST - Hard Reset or System Reset

1 = The last reset was caused by an external device driving $\overline{\text{RSTI}}$. Assertion of reset by an external device causes the core processor to take a reset exception. All registers in internal peripherals and the SIM are reset.

SWTR - Software Watchdog Timer Reset

1 = The last reset was caused by the software watchdog timer. If SWRI in the SYPCR is set and the software watchdog timer times out, a hard reset occurs. $\overline{\text{RSTO}}$ is asserted as an output.

7.3.2.7 SYSTEM PROTECTION CONTROL REGISTER (SYPCR). The SYPCR controls the software watchdog timer and bus time-out monitor enables and time-out periods.

The SYPCR is an 8-bit read-write register. The register can be read at anytime, but can be written only once after system reset. Subsequent writes to the SYPCR will have no effect. At system reset, the software watchdog timer and the bus timeout monitor are disabled.

System Protection Control Register(SYPCR) Address MBAR + $41

	7	6	5	4	3	2	1	0
	SWE	SWRI	SWP	SWT1	SWT0	BME	BMT1	BMT0
RESET:	0	0	0	0	0	0	0	0

R/Write Once

SWE - Software Watchdog Enable

0 = Disable the software watchdog timer
1 = Enable the software watchdog timer

SWRI - Software Watchdog Reset/Interrupt Select

0 = Software watchdog timeout generates a level 7 interrupt to the core processor
1 = Software watchdog timeout generates an internal reset and $\overline{\text{RSTO}}$ will be asserted

NOTE

If SWRI is set to 1, you must set bit 7 in the pin assignment register (PAR) to have $\overline{\text{RSTO}}$ asserted at the pin during a software watchdog timer-generated reset. See subsection **7.3.2.10 Pin Assignment Register (PAR)** for programming information.

SWP - Software Watchdog Prescalar

0 = Software watchdog timer clock is not prescaled
1 = Software watchdog timer clock is prescaled by a value of 512

SWT[1:0] - Software Watchdog Timing

These bits (along with the SWP bit) select the timeout period for the software watchdog timer as shown in Table 7-7. At system rese,t the software watchdog timing bits are set to the minimum timeout period.

Table 7-7. SWT Timeout Period

SWP	SWT[1:0]	SWT TIMEOUT PERIOD
0	00	2^{9} / System Frequency
0	01	2^{11} / System Frequency
0	10	2^{13} / System Frequency
0	11	2^{15}/ System Frequency
1	00	2^{18} / System Frequency
1	01	2^{20} / System Frequency
1	10	2^{22} / System Frequency
1	11	2^{24}/ System Frequency

BME - Bus Timeout Monitor Enable

0 = Disable the bus timeout monitor for external bus cycles
1 = Enable timeout monitor for external bus cycles

NOTE

The bus monitor cannot be disabled for internal bus cycles to internal peripherals.

BMT[1:0] - Bus Monitor Timing

These bits select the timeout period for the bus timeout monitor as shown in Table 7-8. After system reset, the bus monitor timing bits are set to the maximum timeout value.

Table 7-8. Bus Monitor Timeout Periods

BMT[1:0]	BUS MONITOR TIMEOUT PERIOD
00	1024 system clocks
01	512 system clocks
10	256 system clocks
11	128 system clocks

7.3.2.8 SOFTWARE WATCHDOG INTERRUPT VECTOR REGISTER (SWIVR). The SWIVR contains the 8-bit interrupt vector that the SIM returns during an interrupt acknowledge cycle in response to a SWT-generated interrupt.

The SWIVR is an 8-bit write-only register, which is set to the uninitialized vector $0F at system reset.

Software Watchdog Interrupt Vector Register(SWIVR)   Address MBAR + $42

	7	6	5	4	3	2	1	0
	SWIV7	SWIV6	SWIV5	SWIV4	SWIV3	SWIV2	SWIV1	SWIV0
RESET:	0	0	0	0	1	1	1	1

7.3.2.9 SOFTWARE WATCHDOG SERVICE REGISTER (SWSR). The SWSR is the location to which the SWT servicing sequence is written. To prevent an SWT timeout, you should write a $55 followed by a $AA to this register. Although both writes must occur in the order listed prior to the SWT timeout, any number of instructions or accesses to the SWSR can be executed between the two writes. This allows interrupts and exceptions to occur, if necessary, between the two writes.

The SWSR is an 8-bit write-only register. At system reset, the contents of SWSR are uninitialized.

Software Watchdog Service Register(SWSR)   Address MBAR + $43

	7	6	5	4	3	2	1	0
	SWSR7	SWSR6	SWSR5	SWSR4	SWSR3	SWSR2	SWSR1	SWSR0
RESET:	-	-	-	-	-	-	-	-

Write Only

7.3.2.10 PIN ASSIGNMENT REGISTER (PAR). The PAR lets you select certain signal pin assignments. You can select between address, chip-select and write enables, data and parallel port signals, processor status (PST) and parallel port signals, and UART request-to-send and reset out.

The PAR is an 8-bit read-write register. At system reset, all bits are cleared.

Pin Assignment Register(PAR)   Address MBAR + $CB

	7	6	5	4	3	2	1	0
	PAR7	PAR6	PAR5	PAR4	PAR3	PAR2	PAR1	PAR0
RESET:	0	0	0	0	0	0	0	0

R/W

PAR7 - Pin Assignment Bit 7

This bit lets you select the signal output on the $\overline{\text{RTS2}}$/$\overline{\text{RSTO}}$ pin as follows:

0 = Output reset out ($\overline{\text{RSTO}}$) signal on $\overline{\text{RTS2}}$/$\overline{\text{RSTO}}$ pin

1 = Output UART 2 request to send signal on RTS2/RSTO pin

PAR6 - Pin Assignment Bit 6

This bit lets you select how the external interrupt inputs will be used by the MCF5206 as follows:

0 = Set IRQ7/IPL2, IRQ4/IPL1 and IRQ1/IPL0 to be used as individual interrupt inputs at level 7, 4 and 1, respectively.
1 = Set IRQ7/IPL2, IRQ4/IPL1 and IRQ1/IPL0 to be used as encoded interrupt priority level inputs

PAR5 - Pin Assignment Bit 5

This bit lets you select the signals output on the pins PP[7:4]/PST[3:0] as follows:

0 = Output general purpose I/O signals PP7 - PP4 on PP[7:4]/PST[3:0] pins

1 = Output background debug mode signals PST3 - PST0 on PP[7:4]/PST[3:0] pins

PAR4 - Pin Assignment Bit 4

This bit lets you select the signals output on the pins PP[3:0]/DDATA[3:0] as follows:

0 = Output Parallel Port output signals PP3 - PP0 on PP[3:0]/DDATA[3:0] pins
1 = Output background debug mode signals DDATA3 - DDATA0 on PP[3:0]/DDATA[3:0] pins

PAR3 - PAR0 - Pin Assignment Bits 3 - 0

These bits let you select the signal output on the pins CS[7:4]/A[27:24]/WE[3:0]. You can select the signals as shown in Table 7-9.

Table 7-9. PAR3 - PAR0 Pin Assignment

PAR[3:0]	A27/CS7/WE0	A26/CS6/WE1	A25/CS5/WE2	A24/CS4/WE3
0000	WE0	WE1	WE2	WE3
0001	WE0	WE1	CS5	CS4
0010	WE0	WE1	CS5	A24
0011	WE0	WE1	A25	A24
0100	WE0	CS6	CS5	CS4
0101	WE0	CS6	CS5	A24
0110	WE0	CS6	A25	A24
0111	WE0	A26	A25	A24
1000	CS7	CS6	CS5	CS4
1001	CS7	CS6	CS5	A24
1010	CS7	CS6	A25	A24
1011	CS7	A26	A25	A24
1100	A27	A26	A25	A24

Table 7-9. PAR3 - PAR0 Pin Assignment (Continued)

PAR[3:0]	A27/CS7/WE0	A26/CS6/WE1	A25/CS5/WE2	A24/CS4/WE3
1101	Reserved			
1110	Reserved			
1111	Reserved			