SECTION 6
PERIPHERAL CONTROL UNIT

The peripheral control unit (PCU) consists of the following submodules:

- Software watchdog — provides system protection.
- Interrupt controller — controls the interrupts that external peripherals and internal modules send to the CPU.
- Port Q — provides for digital I/O on pins that are not being used as interrupt inputs.
- Test submodule — allows factory testing of the MCU.
- L-bus/IMB2 interface (LIMB) — provides an interface between the load/store bus and the second generation intermodule bus (IMB2). The IMB2 connects on-chip peripherals to the processor via the LIMB.

6.1 PCU Block Diagram

Figure 6-1 shows a block diagram of the PCU.

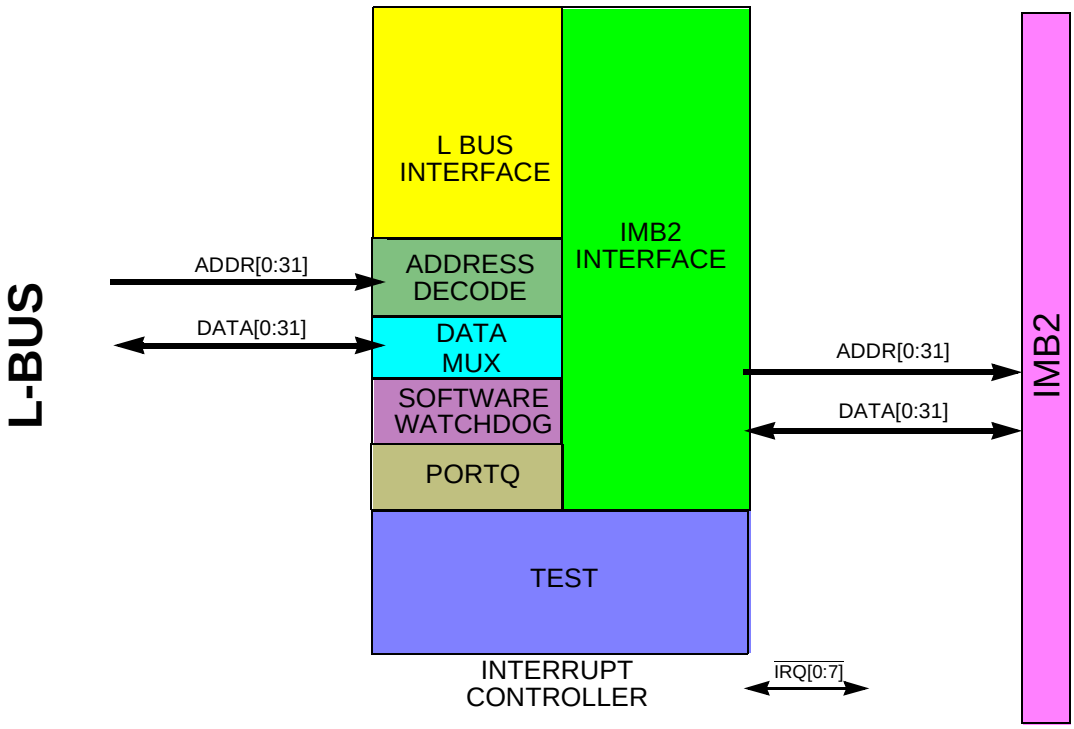

Figure 6-1 Peripherals Control Unit Block Diagram

6.2 PCU Address Map

Table 6-1 shows the address map for the PCU. An entry of “S” in the Access column indicates that the register is accessible in supervisor mode only. “S/U” indicates that the register can be programmed to the desired privilege level. “Test” indicates that the register is accessible in test mode only.

Table 6-1 PCU Address Map

Access	Address	Register	
S	0x8007 EF80	Peripheral control unit module configuration register (PCUMCR)	
—	0x8007 EF84 – 0x8007 EF8C	Reserved	
Test	0x8007 EF90	Test control register (TSTMSRA)	Test control register (TSTMSRB)
Test	0x8007 EF94	Test control register (TSTCNTRAB)	Test control register (TSTREPS)
Test	0x8007 EF98	Test control register (TSTCREG1)	Test control register (TSTCREG2)
Test	0x8007 EF9C	Test control register (TSTDREG)	Reserved
S	0x8007 EFA0	Pending Interrupt request register (IRQPEND)	
S	0x8007 EFA4	Enabled active interrupt request register (IRQAND)	
S	0x8007 EFA8	Interrupt enable register (IRQENABLE)	
S	0x8007 EFAC	PIT/port Q interrupt level register (PITQIL)	
—	0x8007 EFB0 – 0x8007 EFBC	Reserved	
S	0x8007 EFC0	Software service register (SWSR)	Reserved
S	0x8007 EFC4	Software watchdog control field/timing count (SWCR/SWTC)	
S/U	0x8007 EFC8	Software watchdog register	
—	0x8007 EFCC	Reserved	
S/U	0x8007 EFD0	Port Q edge detect/data (PQEDGDAT)	Reserved
S	0x8007 EFD4	Port Q pin assignment register (PQPAR)	
—	0x8007 EFD8 – 0x8007 EFFC	Reserved	

CAUTION

Avoid writing to test location 0x8007 EF98. Setting the high-order bit in this reserved register causes the MCU to enter test mode.

6.3 Module Configuration

The peripheral control unit module configuration register (PCUMCR) contains fields for stopping the system clock to IMB2 modules, assigning certain PCU registers to either supervisor or unrestricted memory space, and assigning the number of interrupt request levels available to IMB2 peripherals.

PCUMCR — Peripheral Control Unit Module Configuration Register **0x8007 EF80**

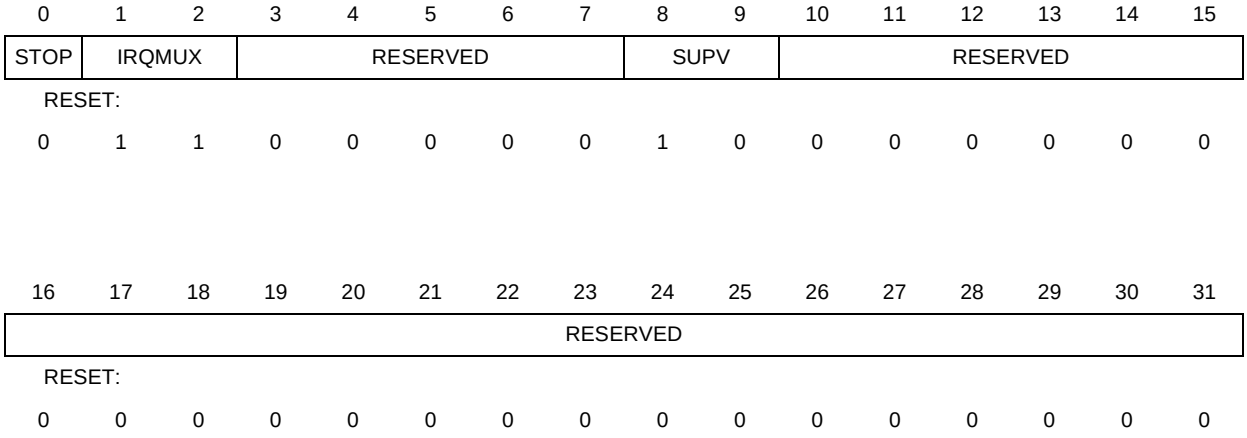

Table 6-2 PCUMCR Bit Settings

Bit(s)	Name	Description
0	STOP	Stop system clock to peripherals controller 0 = Enable system clock to IMB2 modules 1 = Disable system clock to IMB2 modules
1:2	IRQMUX	Interrupt request multiplexer control 00 = Disable multiplexing scheme (eight possible interrupt sources) 01 = 2-to-1 multiplexing (16 possible interrupt sources) 10 = 3-to-1 multiplexing (24 possible interrupt sources) 11 = 4-to-1 multiplexing (32 possible interrupt sources) Refer to **6.5.2.3 Interrupt Request Multiplexing** for details.
3:7	—	Reserved
8:9	SUPV	Supervisor access for PCU registers 00 = Supervisor/unrestricted registers respond to accesses in supervisor or user data space. 01 = Supervisor/unrestricted registers respond to accesses in supervisor space only. 10 = Supervisor/unrestricted registers respond to read accesses in either data space, but write accesses can only be performed in supervisor data space. 11 = Undefined
10:31	—	Reserved

6.4 Software Watchdog

The software watchdog monitors the system software interfaces and requires the software to take periodic action in order to ensure that the program is executing properly. To protect against software error, the following service must be executed on a regular basis:

1. Write 0x556C to the SWSR
2. Write 0xAA39 to the SWSR

This sequence clears the watchdog timer, and the timing process begins again. If this periodic servicing does not occur, the software watchdog issues a reset.

Any number of instructions may occur between the two writes to the SWSR. If any value other than 0x556C or 0xAA39 is written to the SWSR, however, the entire sequence must start over.

6.4.1 Software Watchdog Service Register

A write of 0x556C followed by a write of 0xAA39 to the software watchdog service register (SWSR) causes the software watchdog register to be reloaded with the value in the software watchdog timing count (SWTC) field of the SWCR.

This register can be written at any time within the time-out period. A write of any value other than those shown above resets the servicing sequence, requiring both values to be written to the SWSR before the value in the SWTC field is reloaded into the SWSR.

Reads of the SWSR return zero.

SWSR — Software Watchdog Service Register **0x8007 EFC0**

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

6.4.2 Software Watchdog Control Register/Timing Count

The software watchdog control register/timing count consists of the software watchdog enable (SWE) and software watchdog lock (SWLK) bits and the timing count field for the software watchdog. The software watchdog timing count (SWTC) field contains the 24-bit value that is loaded into the SWSR upon completion of the software watchdog service sequence.

When the SWLK bit is cleared, this register can be written. Once the lock bit is set, further writes to this register have no effect. (In debug mode, however, the lock bit can be cleared by software.) The register can be read at any time.

SWCR/SWTC — Software Watchdog Control Field/Timing Count **0x8007 EFC4**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	SWE	SWLK	SWTC							
RESET:															
0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
SWTC															
RESET:															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 6-3 SWCR/SWTC Bit Settings

Bit(s)	Name	Description
0:5	—	Reserved
6	SWE	Software watchdog enable 0 = Disable watchdog counter 1 = Enable watchdog counter
7	SWLK	Software watchdog lock 0 = Enable changes to SWLK, SWE, SWTC 1 = Ignore writes to SWLK, SWE, SWTC
8:31	SWTC	Software watchdog timing count. This 24-bit register contains the count for the software watchdog timer, which counts at system clock frequency. If this register is loaded with zero, the maximum time-out is programmed.

6.4.3 Software Watchdog Register

The software watchdog register (SWR) is a read-only register that shows the current value of the software watchdog down counter. Writes to this register have no effect.

SWR — Software Watchdog Register **0x8007 EFC8**

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

6.5 Interrupt Controller

Interrupts provide a mechanism for asynchronous, real-time communication between I/O devices and the CPU. The MPC509 interrupt controller joins the simple interrupt structure of the CPU with the complex structure of interrupt sources in the system. The CPU has a single maskable external interrupt. A complete MPC509-based system can have multiple interrupting modules, each with multiple interrupt sources.

The interrupt controller consolidates all the interrupt sources into a single interrupt signal to the processor. Interrupt sources include the periodic interrupt timer, external interrupt pins, and any IMB2 peripherals.

External interrupt input pins are grouped into a general-purpose port (port Q). When not used as interrupt inputs, any of these pins can be used for digital input or output. Port Q operation is described in **6.6 Port Q**.

6.5.1 Interrupt Controller Operation

The following control and status registers associated with the interrupt controller indicate which of 32 possible interrupt levels are pending and control which interrupt sources are passed on to the CPU:

- The pending interrupt request register (IRQPEND) contains a status bit for each of the 32 interrupt levels.
- The interrupt enable register (IRQENABLE) contains an enable bit for each of the 32 interrupt levels.
- The interrupt request levels register (PITQIL) determines the interrupt request level assigned to each interrupt source.

If a bit in the IRQPEND register is asserted (indicating that an interrupt request at the associated level is pending) and the corresponding bit in the IRQENABLE register is asserted, then the interrupt request line to the CPU will be asserted. These registers are described in greater detail in **6.5.3 Interrupt Controller Registers**.

Figure 6-2 provides an overview of MPC509 interrupt management.

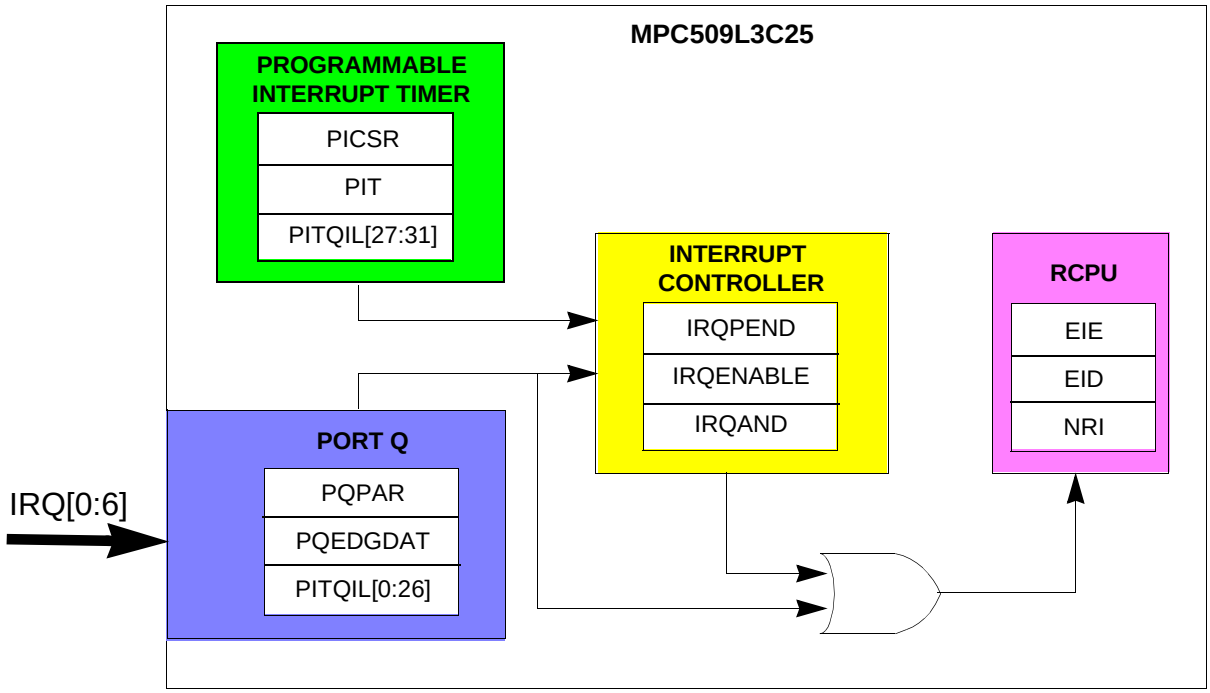

Figure 6-2 Interrupt Structure Block Diagram

The interrupt controller does not enforce a priority scheme. All interrupt priority is determined by the software. In addition, the interrupt controller does not automatically update the interrupt mask upon entering or leaving interrupt processing. Any updates to the interrupt mask are the responsibility of software. In this way, the system is not limited to a particular interrupt priority updating scheme.

In addition to the interrupt controller on the MCU, external peripheral chips in an MPC509-based system may contain their own interrupt controllers. Each interrupt input from an external peripheral chip to the MCU can be routed through the PCU interrupt controller or can be routed directly to the CPU $\overline{\text{IRQ}}$ input. This allows the system

interrupts to be structured in a cascade, where an interrupt controller on an external chip is read only if a certain interrupt on the MCU is serviced, or in parallel, where all interrupt controllers in the system are read and combined before it is determined which interrupt in the system needs servicing.

6.5.2 Interrupt Sources

Sources of interrupt requests to the interrupt controller include the periodic interrupt timer (PIT), two L-bus interrupt request sources, and the $\overline{\text{IRQ}}$[0:6] interrupt request pins. The request levels of PIT interrupts, IMB2 interrupts, and external $\overline{\text{IRQ}}$[0:2] interrupts are assigned by programming the PITQIL register. The request levels of $\overline{\text{IRQ}}$[3:6] external interrupts are assigned fixed values, as explained below.

CAUTION

Be sure the EE (external interrupt enable) bit in the MSR is cleared before changing the masks of any on- or off-chip interrupt sources or before negating any interrupt sources. (On-chip interrupt masks are the IRQENABLE register and the PIE bit in the PICSR.)

6.5.2.1 External Interrupt Requests

The levels of the $\overline{\text{IRQ}}$[0:2] interrupt request pins are assigned by programming the PITQIL. The remaining interrupt request pins have fixed values. $\overline{\text{IRQ3}}$ always generates a level 6 interrupt request; $\overline{\text{IRQ4}}$ generates a level 8 interrupt request; $\overline{\text{IRQ5}}$ generates a level 10 interrupt request; and $\overline{\text{IRQ6}}$ always generates a level 12 interrupt request.

6.5.2.2 Periodic Interrupt Timer Interrupts

The periodic interrupt timer (PIT) is a 16-bit counter that generates an interrupt whenever it counts down to zero, provided PIT interrupts are enabled. The PITIRQL (PIT interrupt request level) field in the PITQIL register assigns the interrupt request level for PIT interrupts. Refer to **5.7.3 Periodic Interrupt Timer (PIT)** for details of PIT operation.

6.5.2.3 Interrupt Request Multiplexing

The IMB2 has ten lines for interrupt support: eight interrupt request lines ($\overline{\text{IRQ}}$[0:7]) from the interrupting modules and two multiplexer control inputs (ILBS[0:1]). This scheme enables the peripheral control unit to transfer up to 32 levels of interrupt requests to the interrupt controller.

When the four-to-one multiplexing scheme is used, the IMB2 $\overline{\text{IRQ}}$ lines update eight of the 32 bits of the IRQPEND register during each clock cycle. A maximum latency of four clock cycles and an average latency of two clock cycles result before the interrupt request can reach the interrupt controller.

Figure 6-3 illustrates the timing for the four-to-one multiplexing scheme.

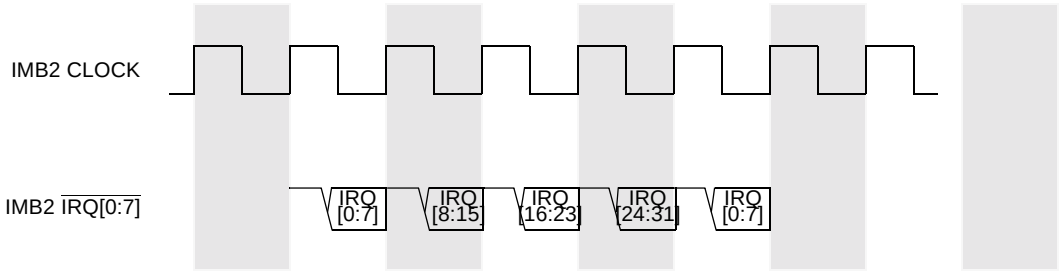

Figure 6-3 Time-Multiplexing Protocol For IRQ Pins

The IRQMUX field in the PCU module configuration register (PCUMCR) selects the type of multiplexing the interrupt controller performs. Refer to **Table 6-4**.

Table 6-4 IMB2 Interrupt Multiplexing

IRQMUX[0:1]	Available IRQ Levels	Type of Multiplexing	Maximum Latency
00	IRQ[0:7]	None	One clock cycle
01	IRQ[0:15]	Two to one	Two clock cycles
10	IRQ[0:23]	Three to one	Three clock cycles
11	IRQ[0:31]	Four to one	Four clock cycles

Time multiplexing is disabled during reset, but the reset default value enables time multiplexing as soon as reset is released.

6.5.3 Interrupt Controller Registers

Control and status registers associated with the interrupt controller indicate which of 32 possible interrupt levels are pending and control which interrupt sources are passed on to the CPU. **Table 6-5** lists these registers.

Table 6-5 Interrupt Controller Registers

Register	Description
Pending Interrupt Request Register (IRQPEND)	Contains a status bit for each of the 32 interrupt levels. Each bit of IRQPEND is a read-only status bit that reflects the current state of the corresponding interrupt signal.
Interrupt Enable Register (IRQENABLE)	Contains an enable bit for each of the 32 interrupt levels.
Enabled Active Interrupt Requests Register (IRQAND)	Logical AND of the IRQPEND and IRQENABLE registers. This register reflects which levels are actually causing the $\overline{IRQ}$ input to the CPU to be asserted.
Interrupt Request Levels Register (PITQIL)	Contains four 5-bit fields that determine the interrupt request levels of the PIT and the $\overline{IRQ}[0:2]$ pins.

6.5.3.1 Pending Interrupt Request Register

The pending interrupt request register (IRQPEND) is a read-only status register that reflects the state of the 32 interrupt levels.

IRQPEND — Pending Interrupt Request Register **0x8007 EFA0**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L0	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
L16	L17	L18	L19	L20	L21	L22	L23	L24	L25	L26	L27	L28	L29	L30	L31
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.5.3.2 Enabled Active Interrupt Requests Register

The enabled active interrupt requests register (IRQAND) is a read-only status register that is defined by the following equation:

$$IRQAND = IRQPEND \ \& \ IRQENABLE$$

where & is a bitwise operation.

IRQAND — Enabled Active Interrupt Requests Register **0x8007 EFA4**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L0	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
L16	L17	L18	L19	L20	L21	L22	L23	L24	L25	L26	L27	L28	L29	L30	L31
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.5.3.3 Interrupt Enable Register

The interrupt enable register (IRQENABLE) is a read/write register. The bits in this register are affected only by writes from the CPU (or other bus master) and by reset.

CAUTION

Be sure the EE (external interrupt enable) bit in the MSR is cleared before changing any masks in this register.

IRQENABLE — Interrupt Enable Register **0x8007 EFA8**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L0	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
L16	L17	L18	L19	L20	L21	L22	L23	L24	L25	L26	L27	L28	L29	L30	L31
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.5.3.4 PIT/Port Q Interrupt Levels Register

The PIT/port Q interrupt levels register (PITQIL) contains four 5-bit fields for programming the interrupt request level of the periodic interrupt timer (PIT) and the $\overline{\text{IRQ}}$[0:2] interrupt request pins. Refer to **6.5.2 Interrupt Sources** for more information.

PITQIL — PIT/Port Q Interrupt Levels Register **0x8007 EFAC**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	IRQ0L					IRQ1L					IRQ2L				
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED											PITIRQL				
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6-6 PITQIL Bit Settings

Bit(s)	Name	Description
0	—	Reserved
1:5	IRQ0L	Interrupt request level for the $\overline{\text{IRQ0}}$ pin
6:10	IRQ1L	Interrupt request level for the $\overline{\text{IRQ1}}$ pin
11:15	IRQ2L	Interrupt request level for the $\overline{\text{IRQ2}}$ pin
16:26	—	Reserved
27:31	PITIRQL	Interrupt request level for PIT interrupts

6.6 Port Q

When not used as interrupt inputs, the $\overline{\text{IRQ[0:6]}}$/PQ[0:6] pins can be used for general-purpose I/O. The following registers control port Q operation:

- Port Q pin assignment register (PQPAR) — allows the user to configure each pin as a digital input, digital output, edge- or level-sensitive interrupt request to the CPU, or edge- or level-sensitive interrupt request to the interrupt controller.
- Port Q edge detect/data register (PQEDGDAT) — contains the following fields:
  - — The port Q data field (PQ[0:6]) monitors or controls the state of port Q pins, depending on the encoding for each pin in the PQPAR.
  - — The port Q edge-detect status field (PQE[0:6]) monitors when the proper transition occurs on a port Q or interrupt request pin.

6.6.1 Port Q Edge Detect/Data Register

The port Q edge detect/data register (PQEDGDAT) consists of the port Q edge-detect status field (PQE[0:6]) and the port Q data field (PQ[0:6]).

Port Q edge status (PQE) bits indicate when the proper transition has occurred on a port Q pin. Each pin can be configured as an interrupt input or as digital I/O. If the pin

is configured in the PQPAR as an edge-sensitive interrupt request pin, then the PQE bit acts as a status bit that indicates whether the corresponding interrupt request line is asserted. The bit also acts as a status bit if the pins are configured as general-purpose inputs or outputs. When the pin is configured in edge-detect mode, the status bit is cleared by reading the bit as a one and then writing it to zero. In level-sensitive mode, the bit remains cleared.

A write to the port Q data register is stored in the internal data latch, and if any PQ bit is configured as an output, the value latched for that bit is driven onto the pin. A read of this port returns the value at the pin only if the pin is configured as a discrete input. Otherwise, the value read is the value stored in the internal data latch. The port Q data register can be read or written at any time.

PQEDGDAT — Port Q Edge Detect/Data Register **0x8007 EFD0**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PQE0	PQE1	PQE2	PQE3	PQE4	PQE5	PQE6	0	PQ0	PQ1	PQ2	PQ3	PQ4	PQ5	PQ6	0
RESET:															
U	U	0	0	0	0	0	0	U	U	0	0	0	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED															
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

U = Unaffected by reset

6.6.2 Port Q Pin Assignment Register

The port Q pin assignment register (PQPAR) contains the port Q pin assignment fields (PQPA[0:6]) and the port Q edge fields (PQEDGE[0:6]).

PQPAR — Port Q Pin Assignment Register **0x8007 EFD4**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PQPA0		PQEDGE0		PQPA1		PQEDGE1		PQPA2		PQEDGE2		PQPA3		PQEDGE3	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
PQPA4		PQEDGE4		PQPA5		PQEDGE5		PQPA6		PQEDGE6		PQPA7		PQEDGE7	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.6.2.1 Port Q Pin Assignment Fields

The port Q pin assignment fields (PQPA[0:6]) select the basic function of each port Q pin, as shown in **Table 6-7**.

Table 6-7 Port Q Pin Assignments

PQPA Value	Pin Function	PORTQ (Port Q Data Field)	Interrupt Request Source
0b00	General-Purpose Input	A read returns the state of the pin. A write has no effect.	None
0b01	General-Purpose Output	A read returns the value in the latch. A write drives the value in the latch onto the pin.	None
0b10	$\overline{\text{IRQ}}$ to CPU	A read returns the state of the pin. A write has no effect.	From pin if PQEDG field is set to “Level”, otherwise from port Q edge detect logic
0b11	$\overline{\text{IRQ}}$ to Interrupt Controller	A read returns the state of the pin. A write has no effect.	From pin if PQEDG field is set to “Level”, otherwise from port Q edge detect logic

6.6.2.2 Port Q Edge Fields

The port Q edge (PQEDGE[0:6]) fields select whether the port/interrupt pin is edge sensitive or level sensitive. When the selected transition occurs on a port Q pin, a corresponding status bit is set in the PQEDGDAT register.

Table 6-8 explains the encodings for PQEDGE fields.

Table 6-8 Port Q Edge Select Field Encoding

PQEDGE Value	Edge Select	PORTQE (Port Q Edge Detect Field)
00	Level sensitive	Returns zero when read
01	Falling-edge sensitive	Set on rising edge
10	Rising-edge sensitive	Set on falling edge
11	Either-edge sensitive	Set on either edge