YOVI 2008 Core

Interrupt Controller

(INTC)

Function Specifications

Rev 0.00

08/05/01

Rev.	Revision content	Approved	Checked	Created
0.00	New created			Duong Dang 08/05/01

Index and Table

1. Scope

This document is the Interrupt Controller.

2. Features

Interrupt controller has some features:

- Two interrupt control modes.
- Priorities settable with ICR.
- Three-level interrupt mask control.
- Independent vector addresses.
- Forty-one external interrupts.
- DTC.

3. Block Diagram

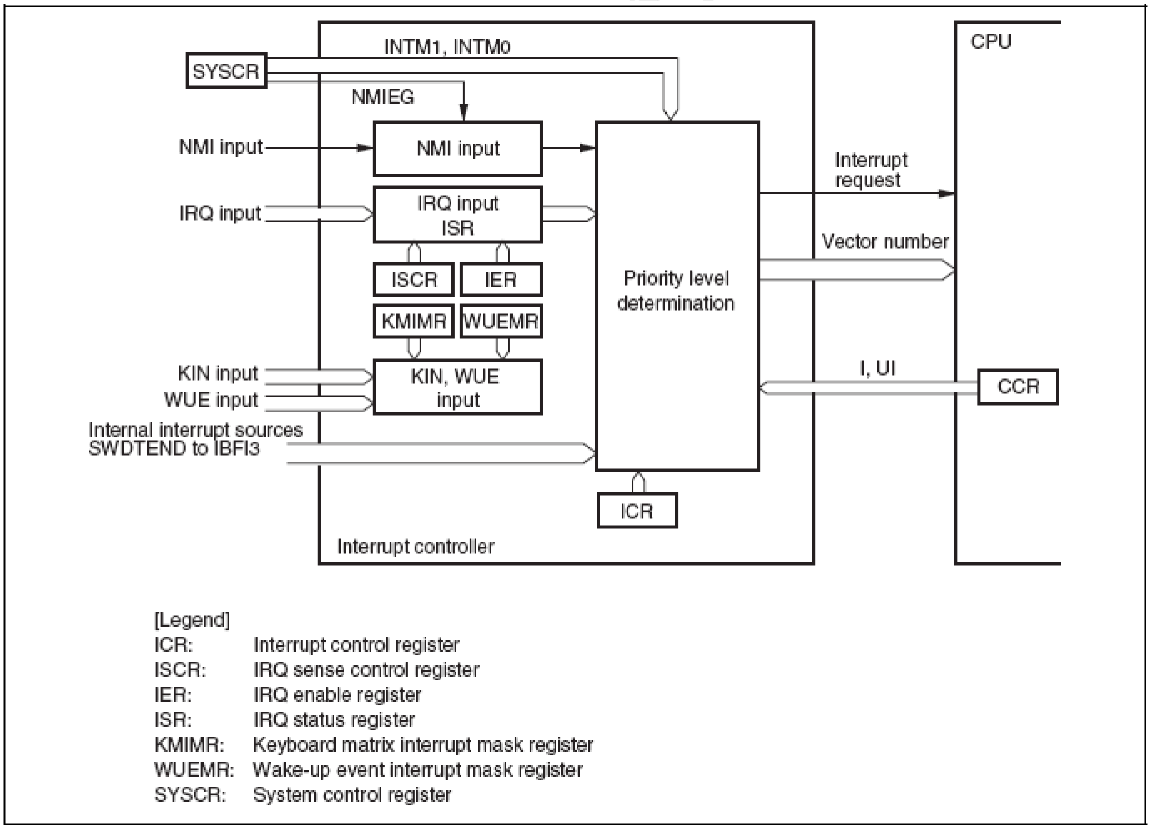


Figure 3.1: Interrupt Controller block diagram

4. Port Descriptions

Table below shows the Interrupt Controller port descriptions

Name	I/O	Function
NMI	Input	Nonmaskable external interrupt. Rising edge or falling edge can be selected.
$\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ2}}$	Input	Maskable external interrupts Rising edge, falling edge or both edges, or level sensing can be selected individually for each pin. Pin of IRQn or ExIRQn to input IQR15 to IQR12 interrupts can be selected.
$\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$	Input	Maskable external interrupts. An interrupt is requested at falling edge.
$\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$	Input	Maskable external interrupts. An interrupt is requested at falling edge.

Table 4.1: Port Description

5. Register Descriptions

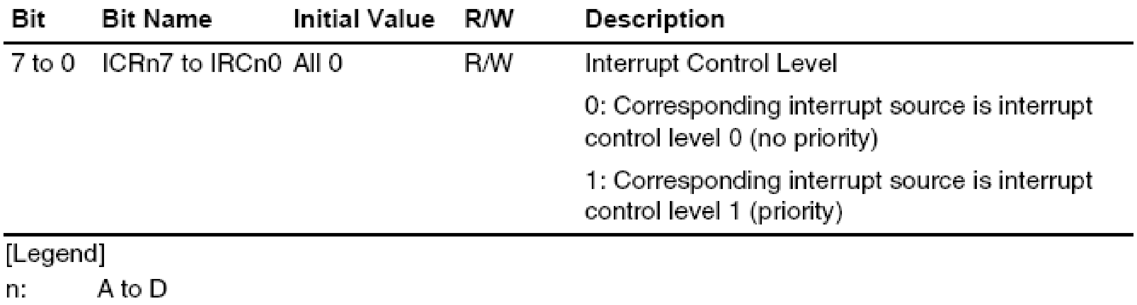
This FRC has the following registers:

- Interrupt control register A to D (ICRA to ICRD)
- Address break control register (ABRKCR)
- Break address register A to C (BARA to BARC)
- IRQ sense control registers (ISCR16H, ISCR16L, ISCRH, ISCRL)
- IRQ enable registers (IER16, IER)
- IRQ status registers (ISR16, ISR)
- Keyboard matrix interrupt mask registers (KMIMRA, KMIMR6)
- Wake-up event interrupt mask register (WUEMR3)

5.1 Interrupt Control Register A to D (ICRA to ICRD)

The ICR registers set interrupt control levels for interrupts other than NMI.

Bit	7	6	5	4	3	2	1	0
	ICR7	**ICR6**	**ICR5**	**ICR4**	**ICR3**	**ICR2**	**ICR1**	**ICR0**
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

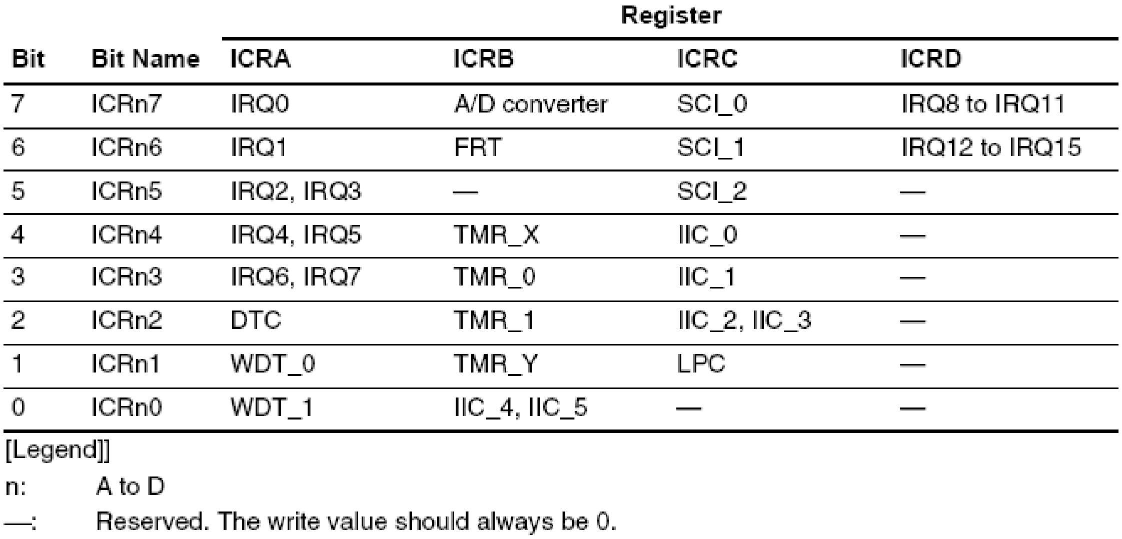


Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ICRn7 to IRCn0	All 0	R/W	Interrupt Control Level 0: Corresponding interrupt source is interrupt control level 0 (no priority) 1: Corresponding interrupt source is interrupt control level 1 (priority)

[Legend]

n: A to D

Figure 5.1: The correspondence between interrupt sources ICRA to ICRD settings



		Register			
Bit	Bit Name	ICRA	ICRB	ICRC	ICRD
7	ICRn7	IRQ0	A/D converter	SCI_0	IRQ8 to IRQ11
6	ICRn6	IRQ1	FRT	SCI_1	IRQ12 to IRQ15
5	ICRn5	IRQ2, IRQ3	—	SCI_2	—
4	ICRn4	IRQ4, IRQ5	TMR_X	IIC_0	—
3	ICRn3	IRQ6, IRQ7	TMR_0	IIC_1	—
2	ICRn2	DTC	TMR_1	IIC_2, IIC_3	—
1	ICRn1	WDT_0	TMR_Y	LPC	—
0	ICRn0	WDT_1	IIC_4, IIC_5	—	—

[Legend]]

n: A to D

—: Reserved. The write value should always be 0.

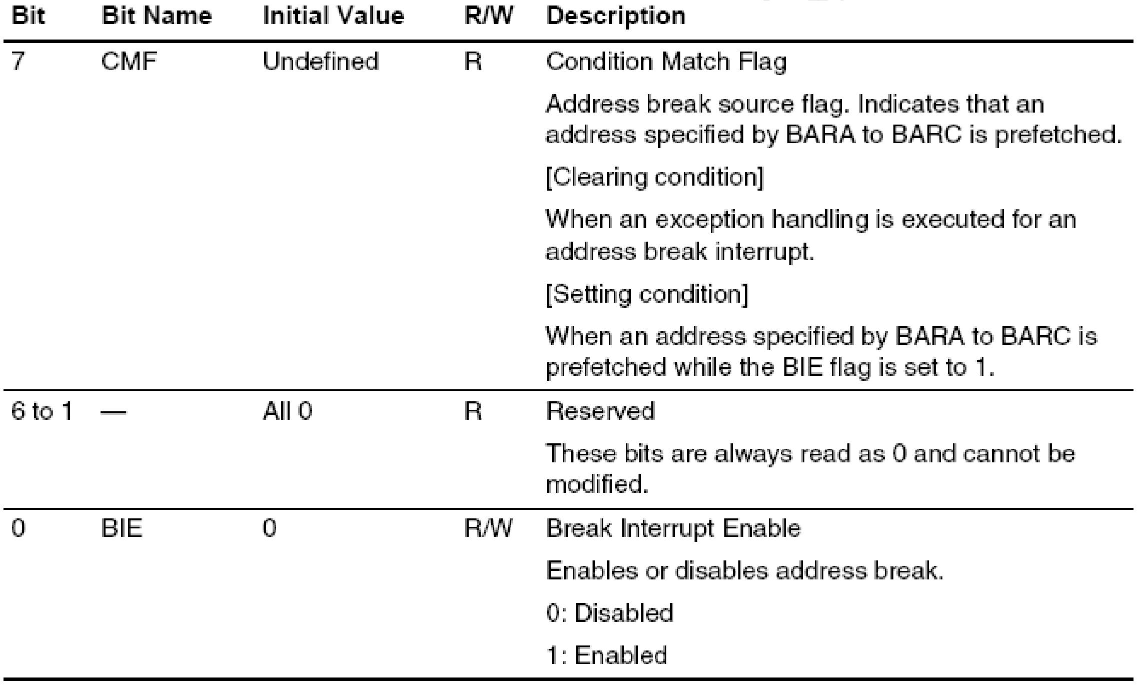
Figure 5.2: Correspondence between Interrupt Source and ICR

5.2 Address Break Control Registers (ABRKCR)

ABRKCR:

- Controls the address breaks.
- An address break is requested, CMF flag = BIE flag = 1

Bit	7	6	5	4	3	2	1	0
	CMF	—	—	—	—	—	—	BIE
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R/W



Bit	Bit Name	Initial Value	R/W	Description
7	CMF	Undefined	R	Condition Match Flag Address break source flag. Indicates that an address specified by BARA to BARC is prefetched. [Clearing condition] When an exception handling is executed for an address break interrupt. [Setting condition] When an address specified by BARA to BARC is prefetched while the BIE flag is set to 1.
6 to 1	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
0	BIE	0	R/W	Break Interrupt Enable Enables or disables address break. 0: Disabled 1: Enabled

5.3 Break Registers A to C (BARA to BARC)

BARA to BARC:

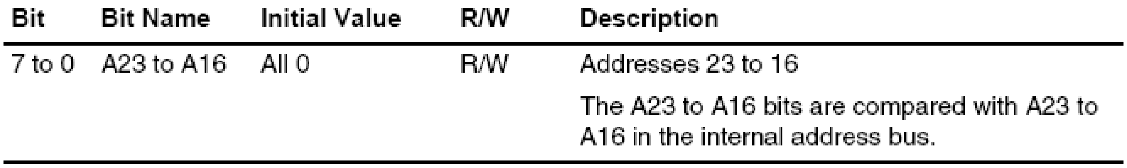
- Specify an address that is to be a break address.

An address in which the first byte of an instruction exists should be set as a break address.

In normal mode, addresses A23 to A16 are not compared.

BARA

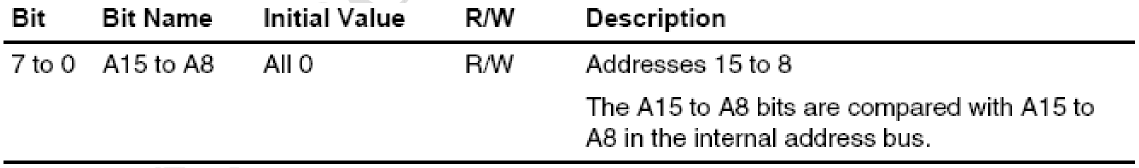
Bit	7	6	5	4	3	2	1	0
	A23	A22	A21	A20	A19	A18	A17	A16
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	A23 to A16	All 0	R/W	Addresses 23 to 16 The A23 to A16 bits are compared with A23 to A16 in the internal address bus.

BARB

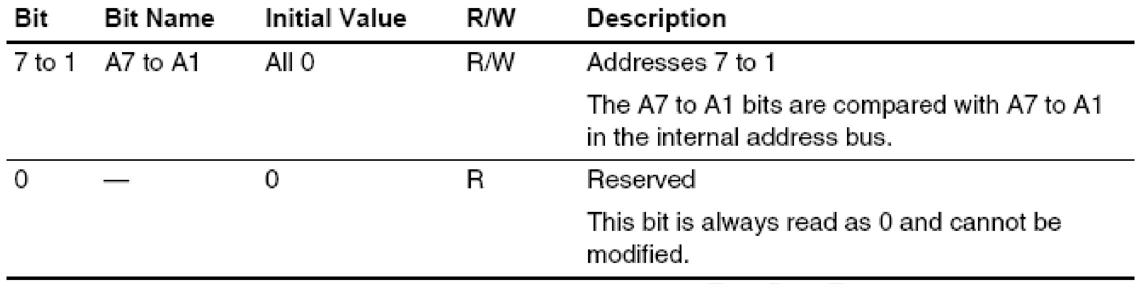
Bit	7	6	5	4	3	2	1	0
	A15	A14	A13	A12	A11	A10	A9	A8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	A15 to A8	All 0	R/W	Addresses 15 to 8 The A15 to A8 bits are compared with A15 to A8 in the internal address bus.

BARC

Bit	7	6	5	4	3	2	1	0
	A7	A6	A5	A4	A3	A2	A1	A0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R



Bit	Bit Name	Initial Value	R/W	Description
7 to 1	A7 to A1	All 0	R/W	Addresses 7 to 1 The A7 to A1 bits are compared with A7 to A1 in the internal address bus.
0	—	0	R	Reserved This bit is always read as 0 and cannot be modified.

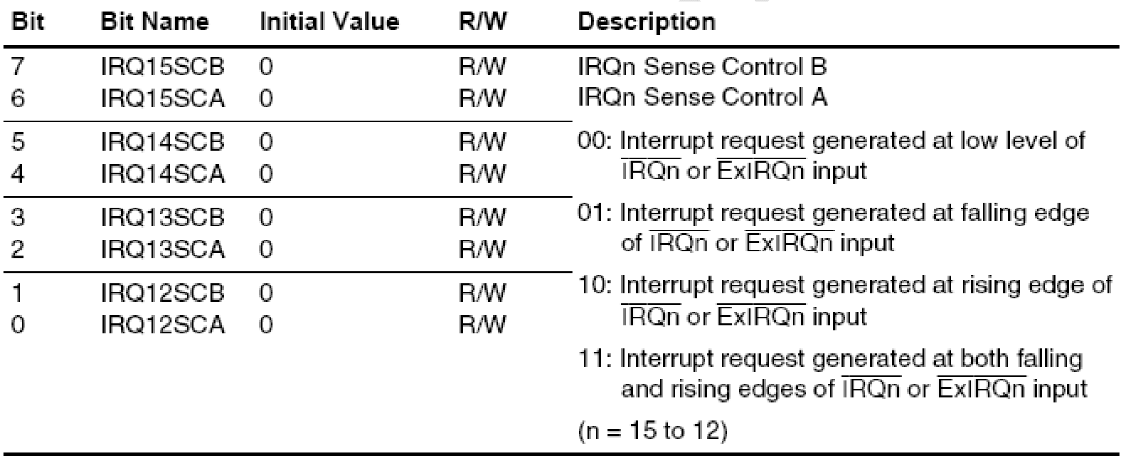
5.4 IRQ Sense Control Registers (ISCR16H, ISCR16L, ISCRH, ISCRL)

IRQ Sense Control Registers:

- Select the source that generates an interrupt request at pins $\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$ or pins $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ2}}$.

ISCR16H

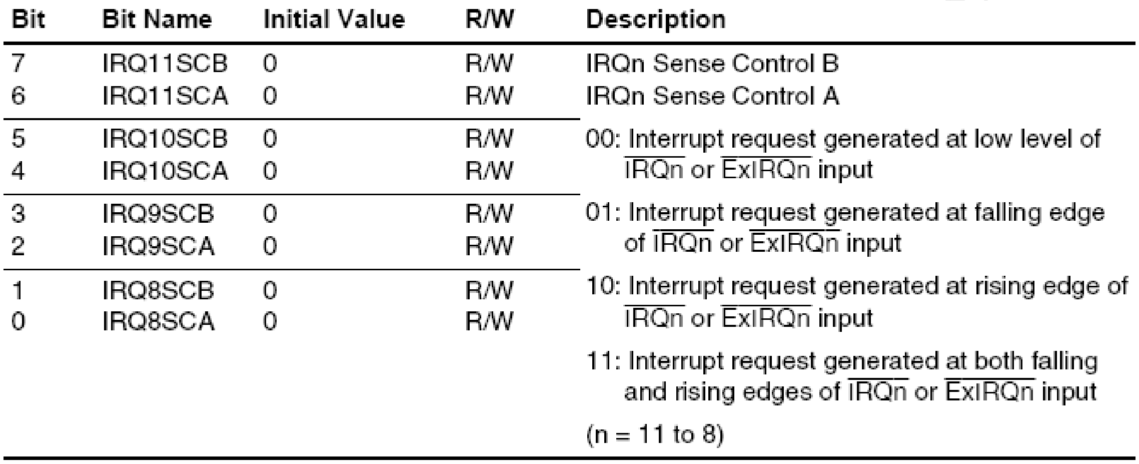
Bit	7	6	5	4	3	2	1	0
	IRQ 15SCB	IRQ 15SCA	IRQ 14SCB	IRQ 14SCA	IRQ 13SCB	IRQ 12SCA	IRQ 12SCB	IRQ 12SCA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



Bit	Bit Name	Initial Value	R/W	Description
7	IRQ15SCB	0	R/W	IRQn Sense Control B
6	IRQ15SCA	0	R/W	IRQn Sense Control A
5	IRQ14SCB	0	R/W	00: Interrupt request generated at low level of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
4	IRQ14SCA	0	R/W	
3	IRQ13SCB	0	R/W	01: Interrupt request generated at falling edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
2	IRQ13SCA	0	R/W	
1	IRQ12SCB	0	R/W	10: Interrupt request generated at rising edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
0	IRQ12SCA	0	R/W	11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input (n = 15 to 12)

ISCR16L

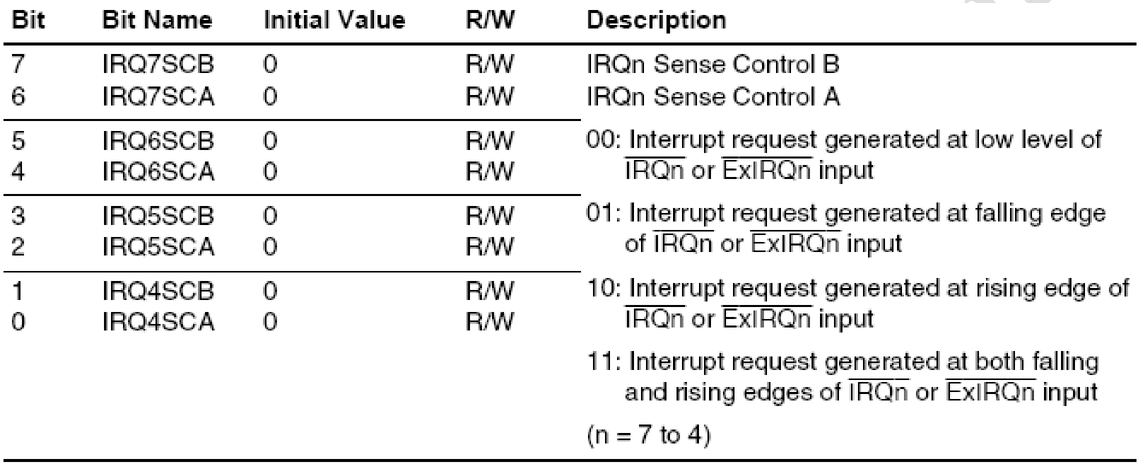
Bit	7	6	5	4	3	2	1	0
	IRQ 11SCB	IRQ 11SCA	IRQ 10SCB	IRQ 10SCA	IRQ 9SCB	IRQ 9SCA	IRQ 8SCB	IRQ 8SCA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



Bit	Bit Name	Initial Value	R/W	Description
7	IRQ11SCB	0	R/W	IRQn Sense Control B
6	IRQ11SCA	0	R/W	IRQn Sense Control A
5	IRQ10SCB	0	R/W	00: Interrupt request generated at low level of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
4	IRQ10SCA	0	R/W	
3	IRQ9SCB	0	R/W	01: Interrupt request generated at falling edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
2	IRQ9SCA	0	R/W	
1	IRQ8SCB	0	R/W	10: Interrupt request generated at rising edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
0	IRQ8SCA	0	R/W	
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input
				(n = 11 to 8)

ISCRH

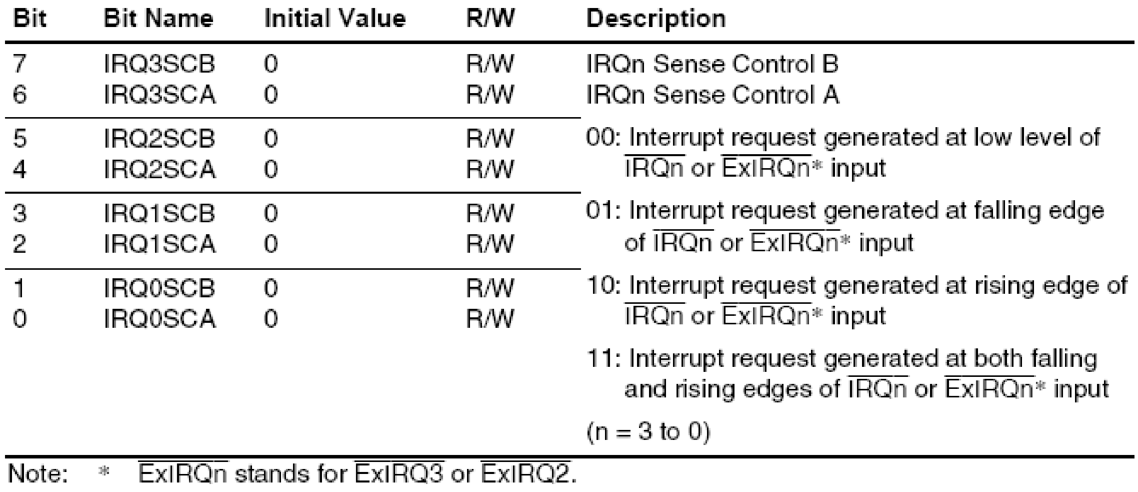
Bit	7	6	5	4	3	2	1	0
	IRQ 7SCB	IRQ 7SCA	IRQ 6SCB	IRQ 6SCA	IRQ 5SCB	IRQ 5SCA	IRQ 4SCB	IRQ 4SCA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7SCB	0	R/W	IRQn Sense Control B
6	IRQ7SCA	0	R/W	IRQn Sense Control A
5	IRQ6SCB	0	R/W	00: Interrupt request generated at low level of $\overline{IRQn}$ or $\overline{ExIRQn}$ input
4	IRQ6SCA	0	R/W	
3	IRQ5SCB	0	R/W	01: Interrupt request generated at falling edge of $\overline{IRQn}$ or $\overline{ExIRQn}$ input
2	IRQ5SCA	0	R/W	
1	IRQ4SCB	0	R/W	10: Interrupt request generated at rising edge of $\overline{IRQn}$ or $\overline{ExIRQn}$ input
0	IRQ4SCA	0	R/W	
				11: Interrupt request generated at both falling and rising edges of $\overline{IRQn}$ or $\overline{ExIRQn}$ input (n = 7 to 4)

ISCRL

Bit	7	6	5	4	3	2	1	0
	IRQ 3SCB	IRQ 3SCA	IRQ 2SCB	IRQ 2SCA	IRQ 1SCB	IRQ 1SCA	IRQ 0SCB	IRQ 0SCA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



Bit	Bit Name	Initial Value	R/W	Description
7	IRQ3SCB	0	R/W	IRQn Sense Control B
6	IRQ3SCA	0	R/W	IRQn Sense Control A
5 4	IRQ2SCB IRQ2SCA	0 0	R/W R/W	00: Interrupt request generated at low level of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$* input
3 2	IRQ1SCB IRQ1SCA	0 0	R/W R/W	01: Interrupt request generated at falling edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$* input
1 0	IRQ0SCB IRQ0SCA	0 0	R/W R/W	10: Interrupt request generated at rising edge of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$* input
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$* input
				(n = 3 to 0)

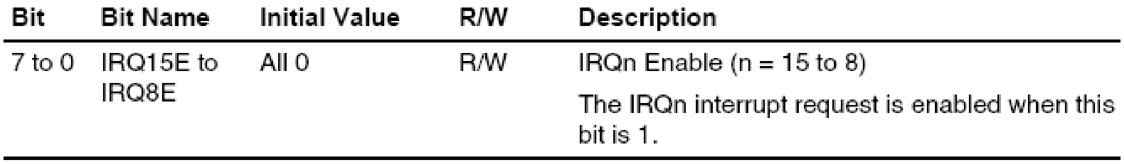
Note: * $\overline{\text{ExIRQn}}$ stands for $\overline{\text{ExIRQ3}}$ or $\overline{\text{ExIRQ2}}$.

5.5 IRQ Enable Register (IER16, IER)

Enable or disable of interrupt requests IRQ15 to IRQ0

IER16

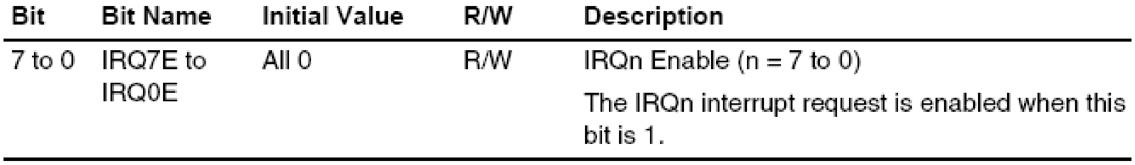
Bit	7	6	5	4	3	2	1	0
	IRQ 15E	IRQ 14E	IRQ 13E	IRQ 12E	IRQ 11E	IRQ 10E	IRQ 9E	IRQ 8E
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	IRQ15E to IRQ8E	All 0	R/W	IRQn Enable (n = 15 to 8) The IRQn interrupt request is enabled when this bit is 1.

IER

Bit	7	6	5	4	3	2	1	0
	IRQ 7E	IRQ 6E	IRQ 5E	IRQ 4E	IRQ 3E	IRQ 2E	IRQ 1E	IRQ 0E
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	IRQ7E to IRQ0E	All 0	R/W	IRQn Enable (n = 7 to 0) The IRQn interrupt request is enabled when this bit is 1.

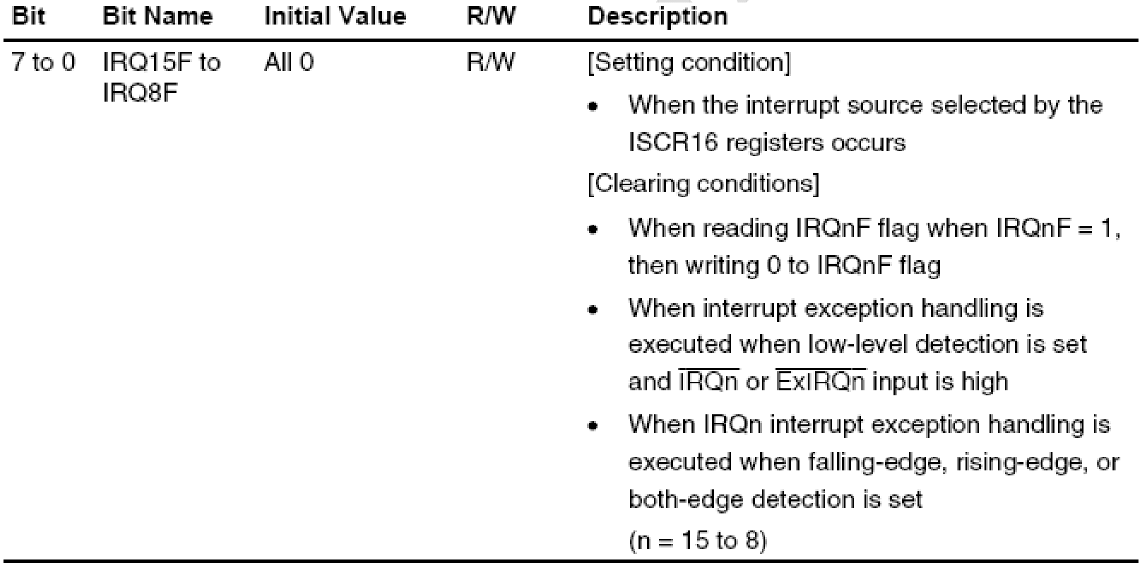
5.6 IRQ Status Registers (ISR16, ISR)

IRQ status registers:

- Are flag registers.
- Indicate the status IRQ15 to IRQ0 interrupt requests.

ISR16

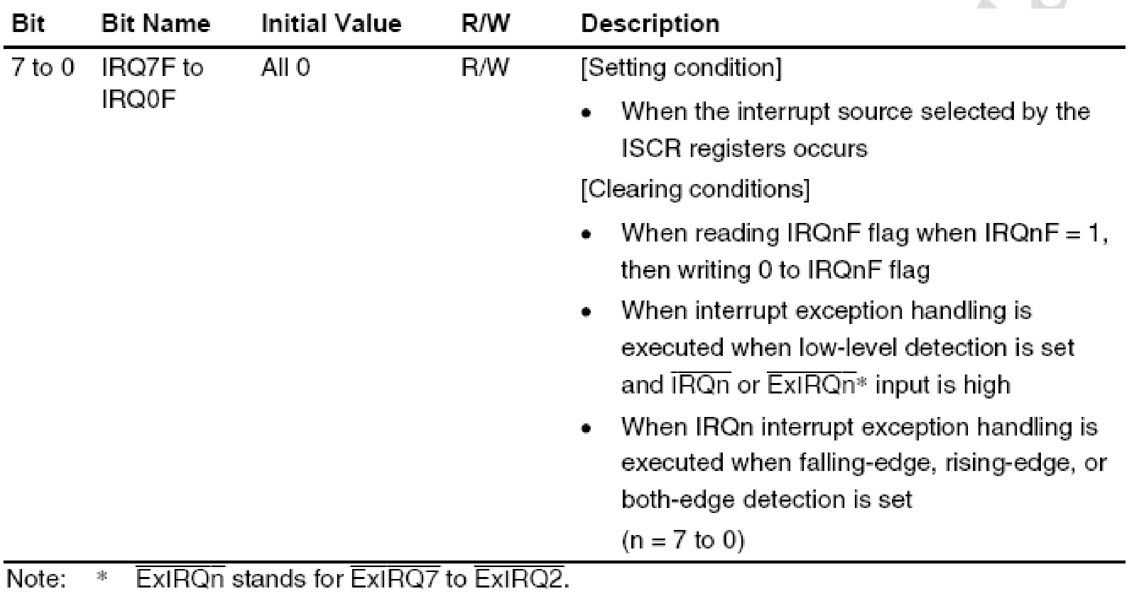
Bit	7	6	5	4	3	2	1	0
	IRQ15F	IRQ14F	IRQ13F	IRQ12F	IRQ11F	IRQ10F	IRQ9F	IRQ8F
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	IRQ15F to IRQ8F	All 0	R/W	[Setting condition] • When the interrupt source selected by the ISCR16 registers occurs [Clearing conditions] • When reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • When interrupt exception handling is executed when low-level detection is set and $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$ input is high • When IRQn interrupt exception handling is executed when falling-edge, rising-edge, or both-edge detection is set (n = 15 to 8)

ISR

Bit	7	6	5	4	3	2	1	0
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	IRQ7F to IRQ0F	All 0	R/W	[Setting condition] • When the interrupt source selected by the ISCR registers occurs [Clearing conditions] • When reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • When interrupt exception handling is executed when low-level detection is set and $\overline{\text{IRQn}}$ or $\overline{\text{ExIRQn}}$* input is high • When IRQn interrupt exception handling is executed when falling-edge, rising-edge, or both-edge detection is set (n = 7 to 0)

Note: * $\overline{\text{ExIRQn}}$ stands for $\overline{\text{ExIRQ7}}$ to $\overline{\text{ExIRQ2}}$.

5.7 Keyboard Matrix Interrupt Mask Registers (KMIMRA, KMIMR6)

Wake-Up Event Interrupt Mask Register (WUEMR3)

KMIMR and WUEMR:

- Enable and disable key-sensing interrupt inputs ($\overline{\text{KIN15}}$ to $\overline{\text{KIN0}}$).
- Wake-up event interrupt inputs ($\overline{\text{WUE15}}$ to $\overline{\text{WUE8}}$).

KMIMRA, KMIMR6 and WUEMR3 registers can be accessed when the KINWUE bit in SYSCR is set to 1.

KMIMRA

Bit	7	6	5	4	3	2	1	0
	KMIM15	KMIM14	KMIM13	KMIM12	KMIM11	KMIM10	KMIM9	KMIM8
Initial Value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	KMIM15 to KMIM8	All 1	R/W	Keyboard Matrix Interrupt Mask These bits enable or disable a key-sensing input interrupt request (KIN15 to KIN8). 0: Enables a key-sensing input interrupt request 1: Disables a key-sensing input interrupt request

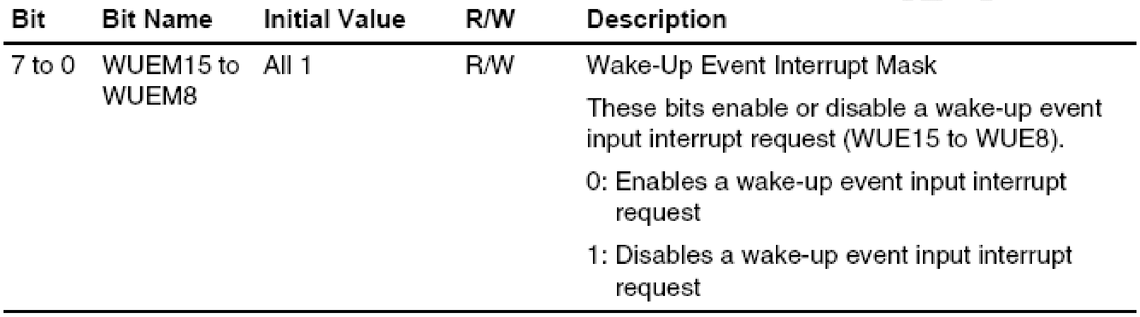
KMIMR6

Bit	7	6	5	4	3	2	1	0
	KMIM7	KMIM6	KMIM5	KMIM4	KMIM3	KMIM2	KMIM1	KMIM0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	KMIM7 to KMIM0	All 1	R/W	Keyboard Matrix Interrupt Mask These bits enable or disable a key-sensing input interrupt request (KIN7 to KIN0). 0: Enables a key-sensing input interrupt request 1: Disables a key-sensing input interrupt request

WUERM3

Bit	7	6	5	4	3	2	1	0
	WUEM15	WUEM14	WUEM13	WUEM12	WUEM11	WUEM10	WUEM9	WUEM8
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	WUEM15 to WUEM8	All 1	R/W	Wake-Up Event Interrupt Mask These bits enable or disable a wake-up event input interrupt request (WUE15 to WUE8). 0: Enables a wake-up event input interrupt request 1: Disables a wake-up event input interrupt request

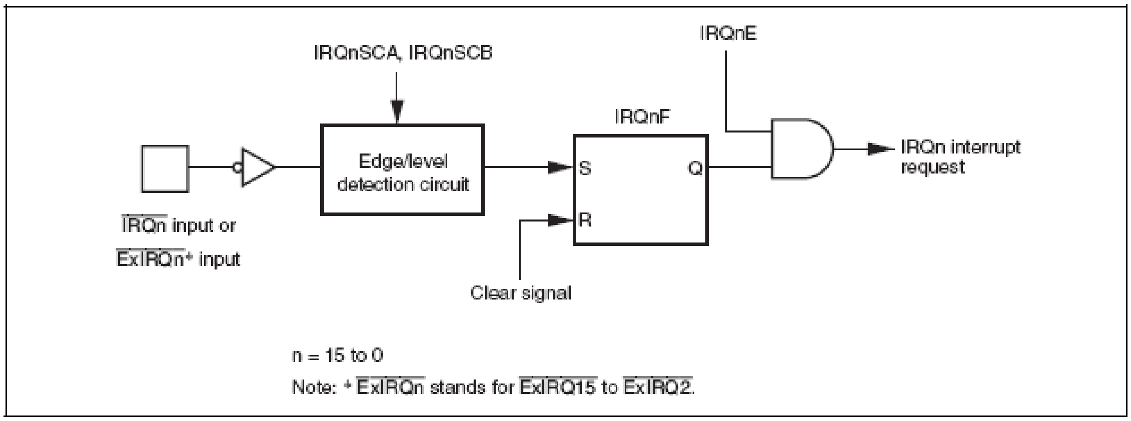
6. Interrupt Sources

6.1 External Interrupts

There are four external interrupts:

- NMI Interrupt: is the highest-priority interrupt, and is always accepted by the CPU regardless of the interrupt control mode or the status of the CPU interrupt mask bits. The NMIEG bit in SYSCR can be used to select whether an interrupt is requested at the rising edge or a falling edge on the NMI pin.
- IRQ15 to IRQ0 Interrupts: are requested by an input signal at pins IRQ15 to IRQ0 or pins ExIRQ15 to ExIRQ2. They have following features:
  - The interrupt exception handling for interrupt requests IRQ15 to IRQ0 can be started at an independent vector address.
  - Using ISCR to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges at pins IRQ15 to IRQ0 or pins ExIRQ`5 to ExIRQ2.

- Enabling or disabling of interrupt requests IRQ15 to IRQ0 can be selected with IER.
- The status of interrupt request IRQ15 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.


IRQnSCA, IRQnSCB
IRQnE
IRQnF
Edge/level detection circuit
S
Q
R
IRQn interrupt request
$\overline{\text{IRQn}}$ input or
$\overline{\text{ExIRQn}}$* input
Clear signal

n = 15 to 0

Note: * $\overline{\text{ExIRQn}}$ stands for $\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ2}}$.

Figure 6.1: Block diagram of Interrupts IRQ15 to IRQ0

- KIN15 to KIN0 Interrupts, WUE15 to WUE8 Interrupts: Interrupts KIN15 to KIN0 and WUE15 to WUE8 are requested by an input signal at pins KIN15 to KIN0 and WUE15 to WUE8. They have the following features:
  - The interrupts KIN15 and KIN8, KIN7 to KIN0 and WUE15 to WUE8 each form a group.
  - The interrupt exception handling for an interrupt request from the same group is started at the same vector address.
  - Enabling or disabling of interrupt requests can be selected with the I bit in CCR.• An interrupt is generated by a falling edge at pins KIN15 to KIN0 and WUE15 to WUE8.
  - Enabling or disabling of interrupt requests KIN15 to KIN0 and WUE15 to WUE8 can be selected using KMIMRA, KMIMR6, and WUEMR3.
  - The status of interrupt requests KIN15 to KIN0 and WUE15 to WUE8 are not indicated.

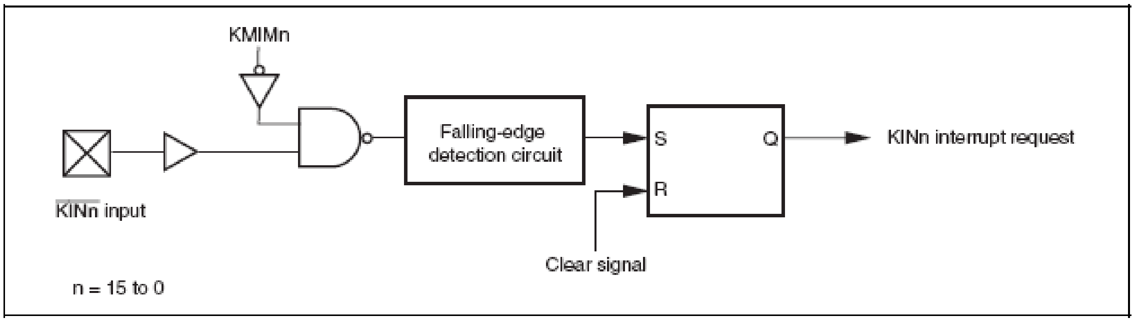


Figure 6.2: Block Diagrams of Interrupts KIN15 to KIN0 and WUE15 to WUE8 (Example of KIN15 to KIN0)

6.2 Internal Interrupts

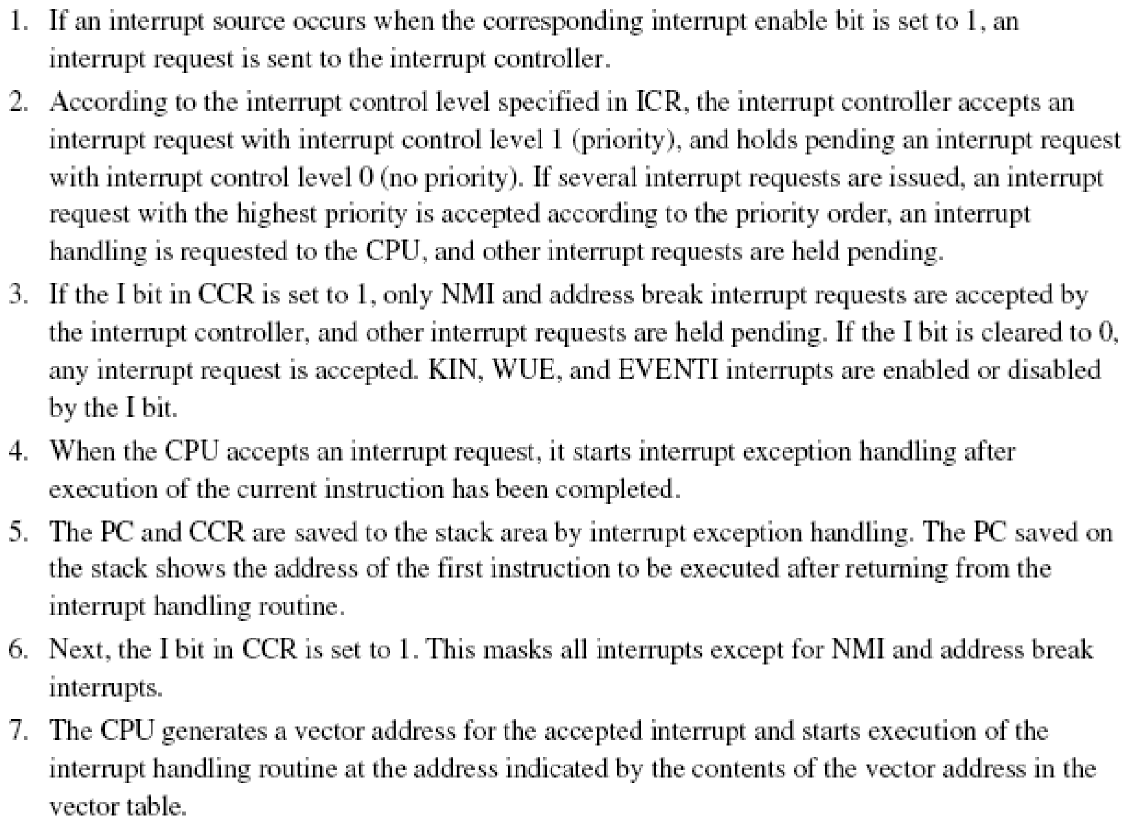
Internal interrupts issued from the on-chip peripheral modules have the following features:

- For each on-chip peripheral module there are flags that indicate the interrupt request status, and enable bits that individually select enabling or disabling of these interrupts. When the enable bit for a particular interrupt source is set to 1, an interrupt request is sent to the interrupt controller.
- The control level for each interrupt can be set by ICR.
- The DTC can be activated by an interrupt request from an on-chip peripheral module.
- An interrupt request that activates the DTC is not affected by the interrupt control mode or the status of the CPU interrupt mask bits.

7. Interrupt Control Mode and Interrupt Operation

7.1 Interrupt Mode 0

In Interrupt control mode 0, interrupts other than NMI are masked by ICR and the I bit of the CCR in the CPU. Figure below shows a flowchart of the interrupt acceptance operation.



1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. According to the interrupt control level specified in ICR, the interrupt controller accepts an interrupt request with interrupt control level 1 (priority), and holds pending an interrupt request with interrupt control level 0 (no priority). If several interrupt requests are issued, an interrupt request with the highest priority is accepted according to the priority order, an interrupt handling is requested to the CPU, and other interrupt requests are held pending.
3. If the I bit in CCR is set to 1, only NMI and address break interrupt requests are accepted by the interrupt controller, and other interrupt requests are held pending. If the I bit is cleared to 0, any interrupt request is accepted. KIN, WUE, and EVENTI interrupts are enabled or disabled by the I bit.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except for NMI and address break interrupts.
7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

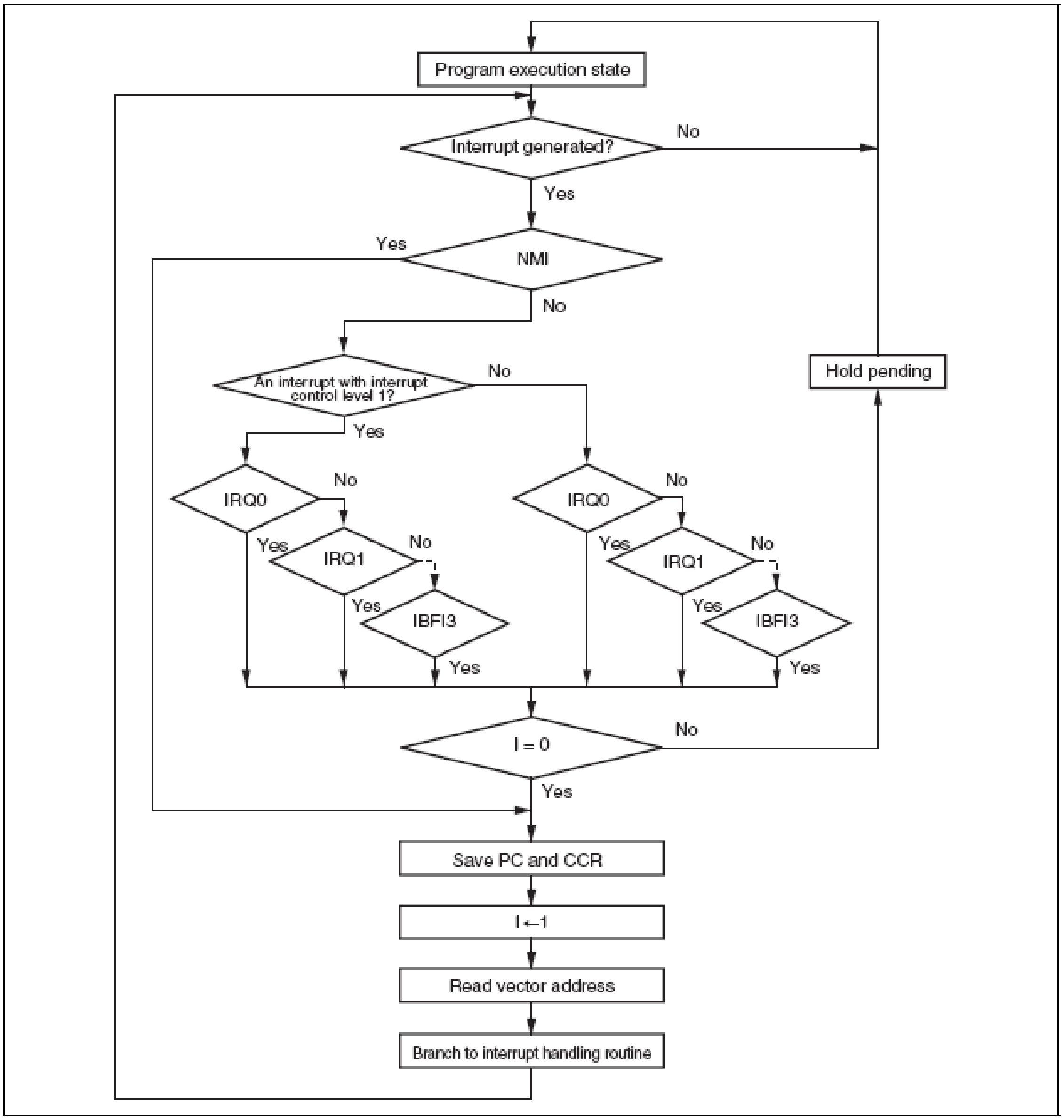


Figure 7.1: Flowchart of Procedure up to Interrupt Acceptance

in Interrupt Control Mode 0

7.2 Interrupt Mode 1

In interrupt control mode 1, mask control is applied to three levels for IRQ and on-chip peripheral module interrupt requests by comparing the I and UI bits in CCR in the CPU, and the ICR setting.

- An interrupt request with interrupt control level 0 is accepted when the I bit in CCR is cleared to 0. When the I bit is set to 1, the interrupt request is held pending. EVENTI, KIN, and WUE interrupts are enabled or disabled by the I bit.
- An interrupt request with interrupt control level 1 is accepted when the I bit or UI bit in CCR is cleared to 0. When both I and UI bits are set to 1, the interrupt request is held pending.

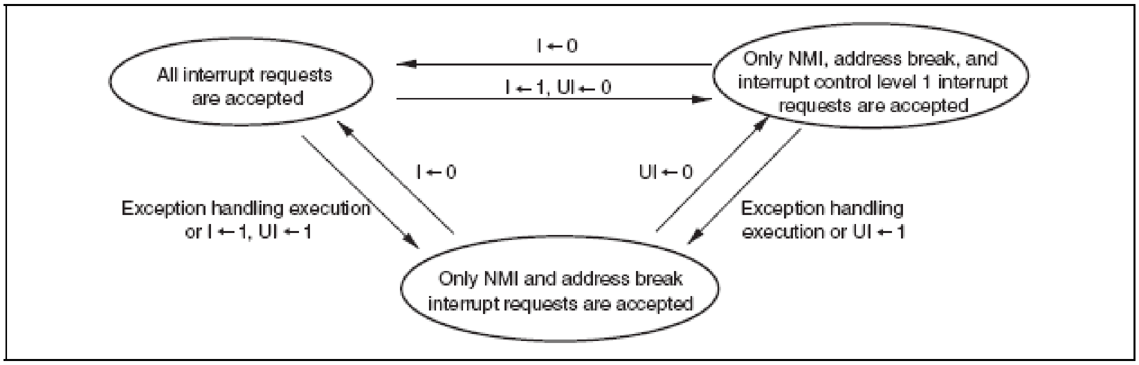
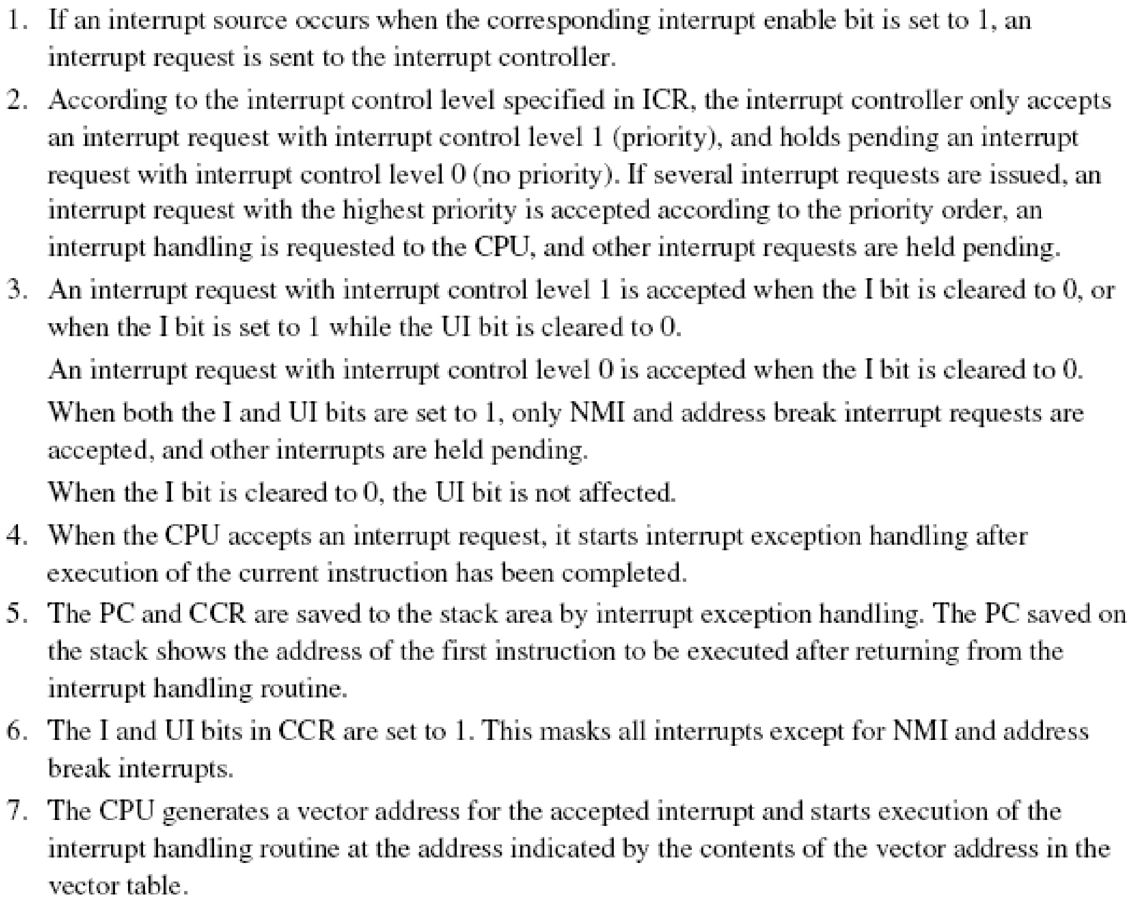

All interrupt requests are accepted
I ← 0
I ← 1, UI ← 0
Only NMI, address break, and interrupt control level 1 interrupt requests are accepted
I ← 0
UI ← 0
Exception handling execution or I ← 1, UI ← 1
Exception handling execution or UI ← 1
Only NMI and address break interrupt requests are accepted

Figure 7.2: State Transition in Interrupt Control Mode 1



1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. According to the interrupt control level specified in ICR, the interrupt controller only accepts an interrupt request with interrupt control level 1 (priority), and holds pending an interrupt request with interrupt control level 0 (no priority). If several interrupt requests are issued, an interrupt request with the highest priority is accepted according to the priority order, an interrupt handling is requested to the CPU, and other interrupt requests are held pending.
3. An interrupt request with interrupt control level 1 is accepted when the I bit is cleared to 0, or when the I bit is set to 1 while the UI bit is cleared to 0.

   An interrupt request with interrupt control level 0 is accepted when the I bit is cleared to 0.

   When both the I and UI bits are set to 1, only NMI and address break interrupt requests are accepted, and other interrupts are held pending.

   When the I bit is cleared to 0, the UI bit is not affected.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. The I and UI bits in CCR are set to 1. This masks all interrupts except for NMI and address break interrupts.
7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.

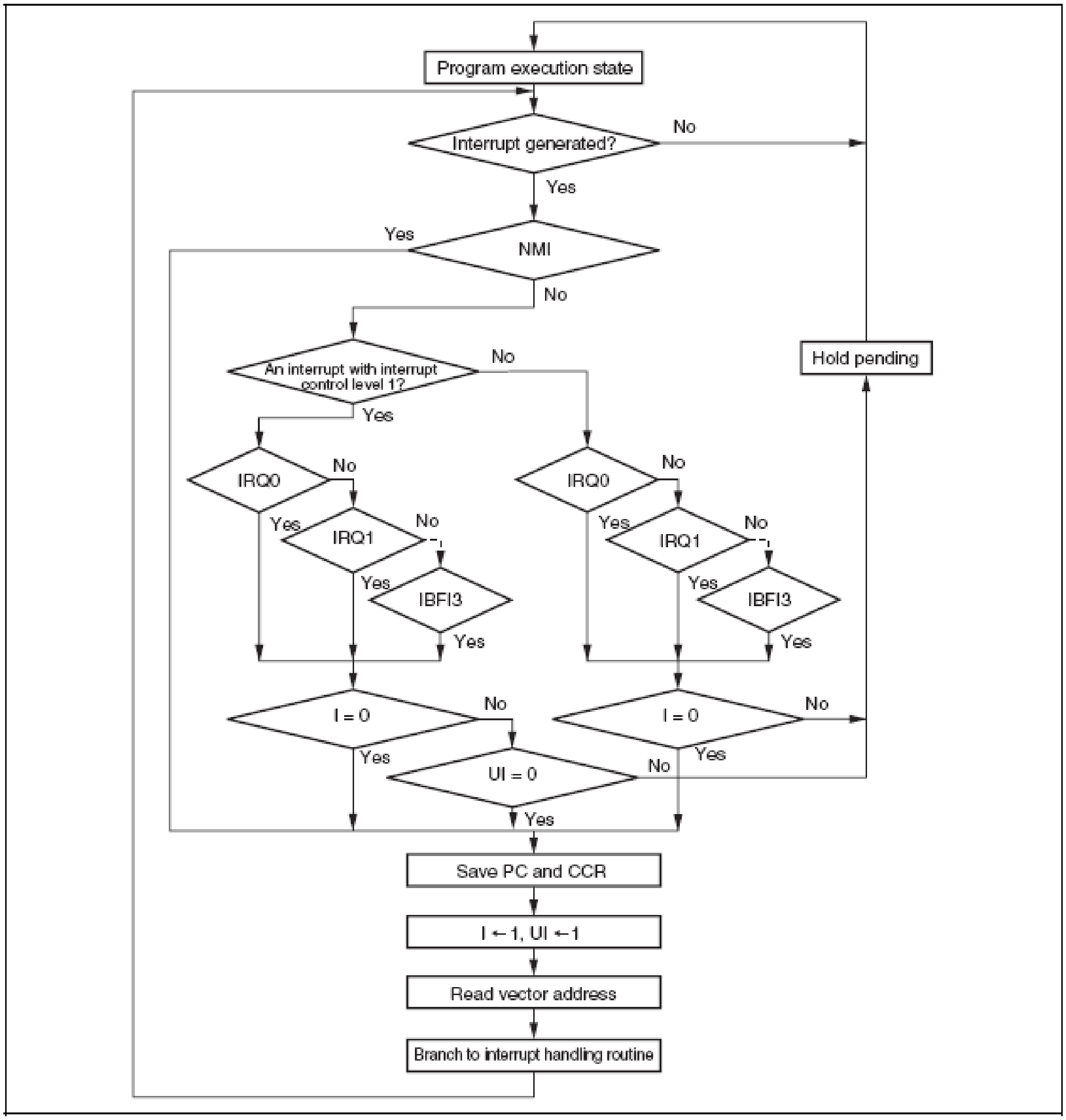


Figure 7.3: Flowchart of Procedure Up to Interrupt Acceptant in Interrupt Control Mode 1

7.3 Interrupt Exception Handling Sequence

In interrupt control mode 1, mask control is applied to three levels for IRQ and on-chip peripheral module interrupt requests by comparing the I and UI bits in CCR in the CPU, and the ICR setting.

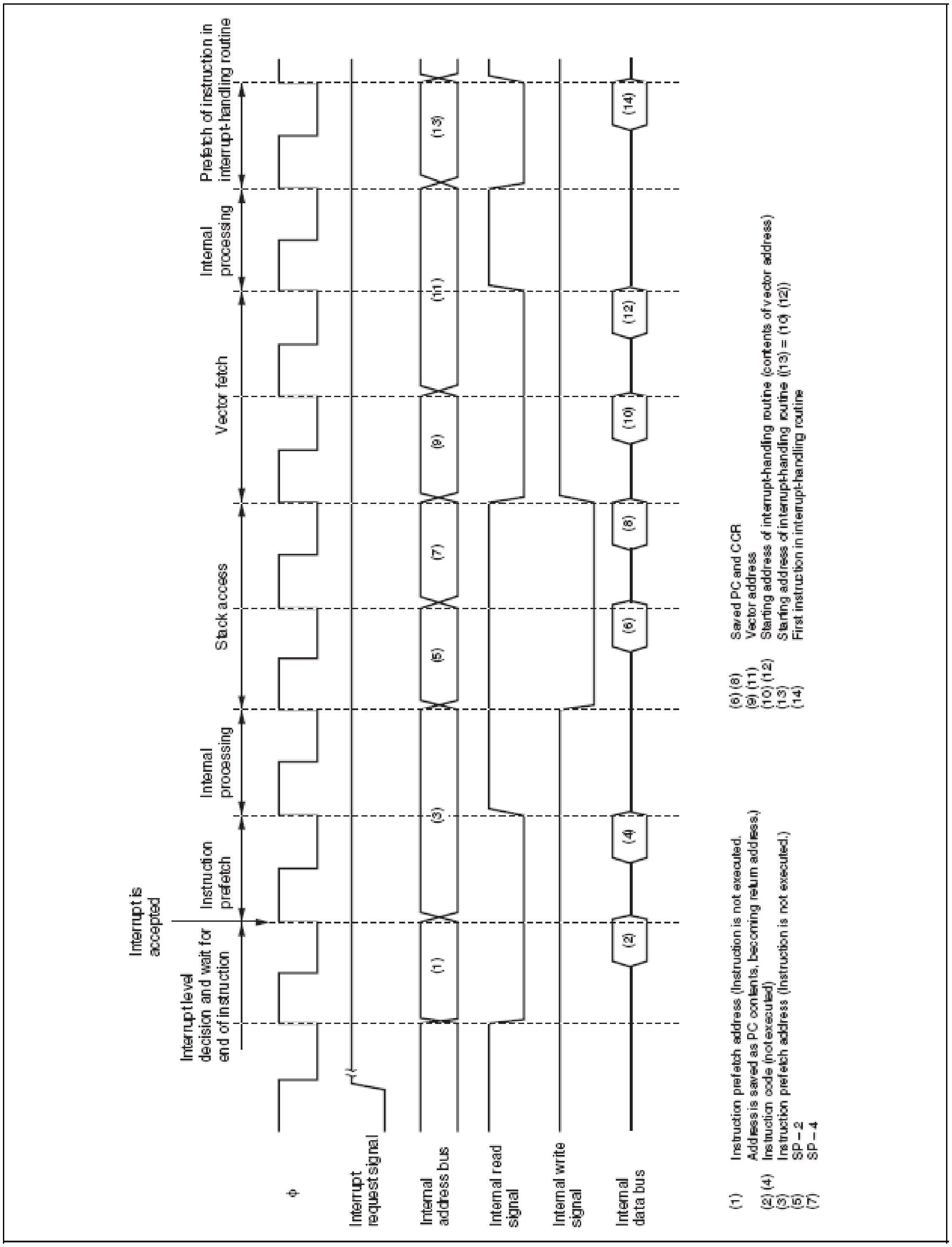
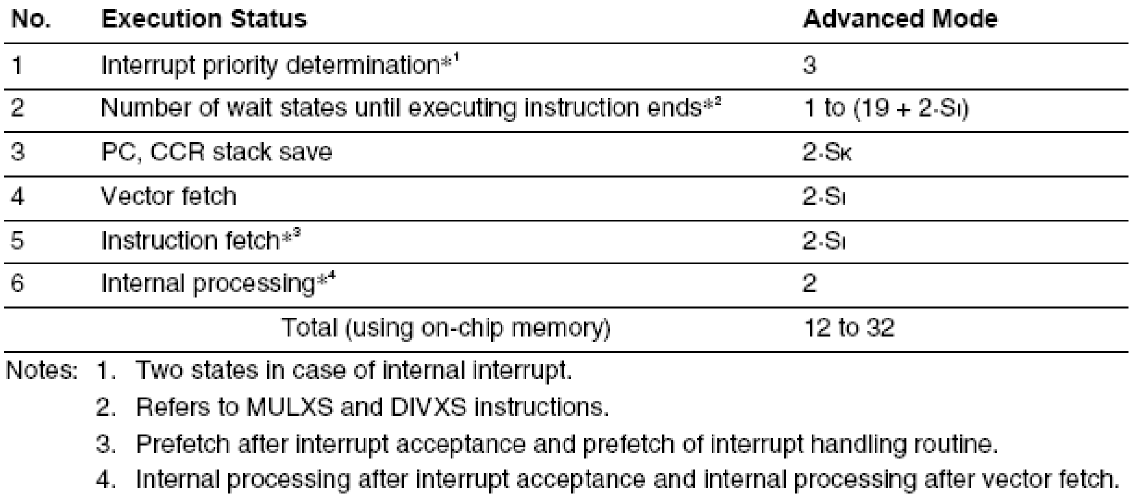

φ
Interrupt request signal
Internal address bus
Internal read signal
Internal write signal
Internal data bus
Interrupt level decision and wait for end of instruction
Interrupt is accepted
Instruction prefetch
Internal processing
Stack access
Vector fetch
Internal processing
Prefetch of instruction in interrupt-handling routine
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13)
(14)
(1) Instruction prefetch address (Instruction is not executed.
Address is saved as PC contents, becoming return address.)
(2) (4) Instruction code (not executed)
(3) Instruction prefetch address (Instruction is not executed.)
(5) SP – 2
(7) SP – 4
(6) (8) Saved PC and CCR
(9) (11) Vector address
(10) (12) Starting address of interrupt-handling routine (contents of vector address)
(13) Starting address of interrupt-handling routine ((13) = (10) (12))
(14) First instruction in interrupt-handling routine

Figure 7.2: Interrupt Exception Handling

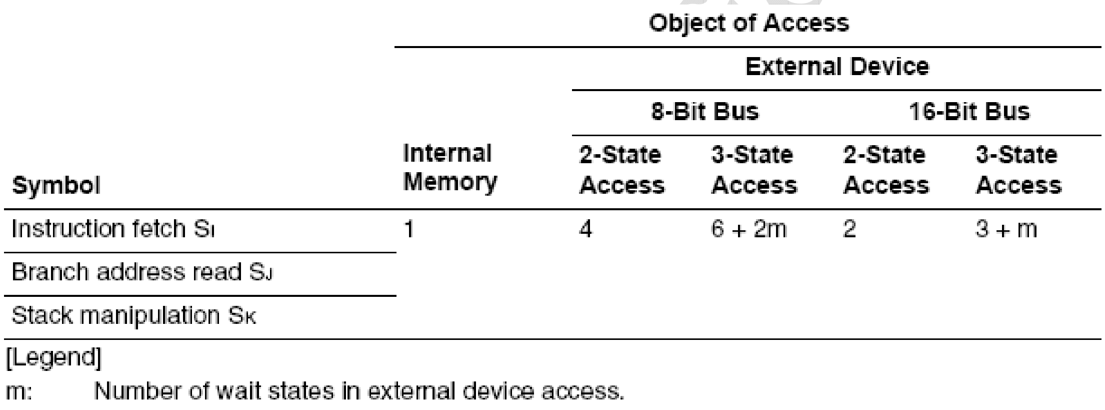
7.4 Interrupt Response Time



No.	Execution Status	Advanced Mode
1	Interrupt priority determination*1	3
2	Number of wait states until executing instruction ends*2	1 to $(19 + 2 \cdot S_I)$
3	PC, CCR stack save	$2 \cdot S_K$
4	Vector fetch	$2 \cdot S_I$
5	Instruction fetch*3	$2 \cdot S_I$
6	Internal processing*4	2
	Total (using on-chip memory)	12 to 32

Notes: 1. Two states in case of internal interrupt.
2. Refers to MULXS and DIVXS instructions.
3. Prefetch after interrupt acceptance and prefetch of interrupt handling routine.
4. Internal processing after interrupt acceptance and internal processing after vector fetch.

Figure 7.3: Interrupt Response Time



	Object of Access				
		External Device			
		8-Bit Bus		16-Bit Bus	
Symbol	Internal Memory	2-State Access	3-State Access	2-State Access	3-State Access
Instruction fetch S_I	1	4	6 + 2m	2	3 + m
Branch address read S_J					
Stack manipulation S_K					

[Legend]
m: Number of wait states in external device access.

Figure 7.4: Number of States in Interrupt Handling Routine Execution Status

7.5 DTC Activation by Interrupt

The DTC can be activated by an interrupt. In this case, the following options are available:

- Interrupt request to CPU.
- Activation request to DTC.
- Both of the above.

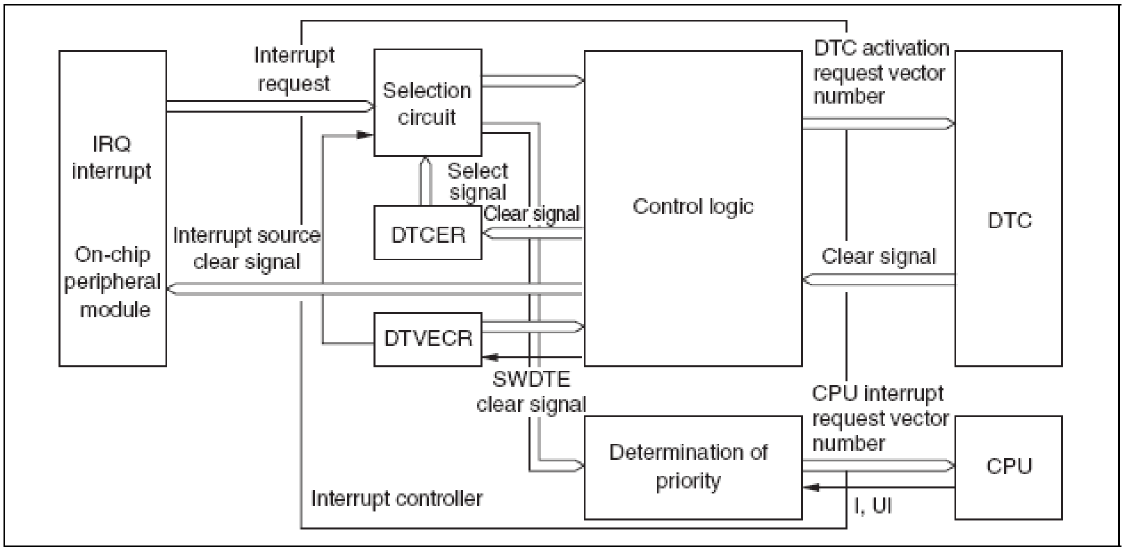
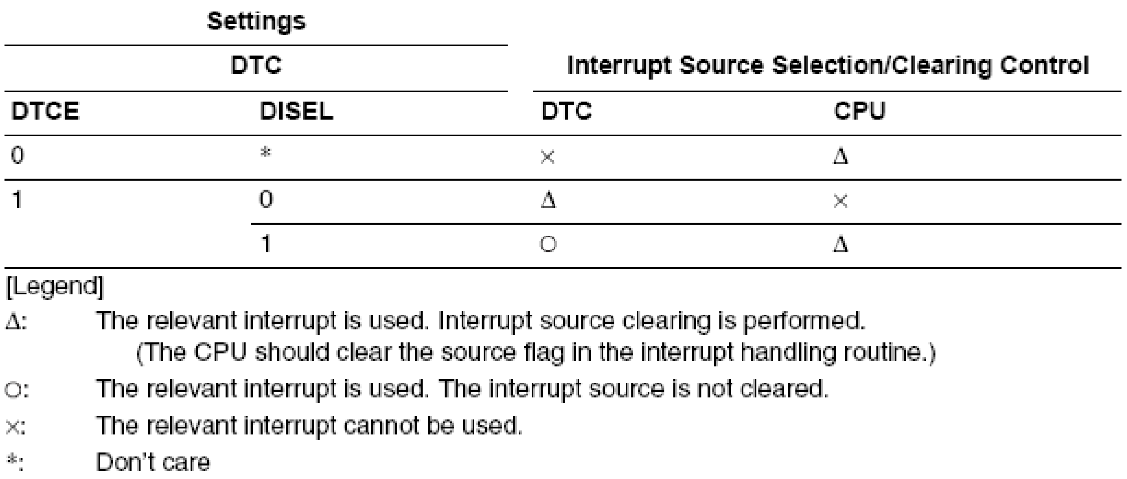


Figure 7.5: Interrupt Control for DTC

The interrupt controller has three main functions in DTC control.

- **Selection of Interrupt Source:** It is possible to select DTC activation request or CPU interrupt request with the DTCE bit of DTCERA to DTCERE in the DTC. After a DTC data transfer, the DTCE bit can be cleared to 0 and an interrupt request sent to the CPU in accordance with the specification of the DISEL bit of MRB in the DTC. When the DTC performs the specified number of data transfers and the transfer counter reaches 0, following the DTC data transfer the DTCE bit is cleared to 0 and an interrupt request is sent to the CPU.
- **Determination of Priority:** The DTC activation source is selected in accordance with the default priority order, and is not affected by mask or priority levels. See section 7.5, Location of Register Information and DTC Vector Table, for the respective priorities.
- **Operation Order:** If the same interrupt is selected as a DTC activation source and a CPU interrupt source, the DTC data transfer is performed first, followed by CPU interrupt exception handling.



Settings		Interrupt Source Selection/Clearing Control	
DTC			
DTCE	DISEL	DTC	CPU
0	*	×	Δ
1	0	Δ	×
	1	○	Δ

[Legend]

Δ: The relevant interrupt is used. Interrupt source clearing is performed.
(The CPU should clear the source flag in the interrupt handling routine.)

○: The relevant interrupt is used. The interrupt source is not cleared.

×: The relevant interrupt cannot be used.

*: Don't care

Figure 7.6: Interrupt Source Selection and Clearing Control

-End-