To our customers,

# Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

# Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

# H8SX Family

## Interrupt Controller

## Introduction

The interrupt controller of H8SX microcontrollers applies either of two kinds of interrupt control, referred to as mode 0 and mode 2. Differences between the two are described in this application note.

## Target Device

H8SX/1638, H8SX/1648, H8SX/1658, H8SX/1663, H8SX/1668 Groups

## Preface

Although the writing of this application note is in accord with the hardware manual for the H8SX/1663 Group, the program covered in this application note can be run on the target devices indicated above. However, since some functional modules may be changed for the addition of functionality etc., be sure to perform a thorough evaluation by confirming the details with the hardware manual for the target device.

## Contents

## 1. Specification

For the interrupt controller of H8SX microcontrollers, control in either interrupt control mode 0 or interrupt control mode 2 is selectable. The sample task for this application note illustrates differences in operation according to the interrupt control mode when IRQ2 to IRQ0 interrupt requests are simultaneously generated.

1. As shown in figure 1, when the processing of IRQ2 to IRQ0 interrupts is initiated, high-level trigger pulses are output on the respective pins from P22 to P20.
2. In the state where IRQ2 to IRQ0 interrupt requests have been disabled, the interrupts are held pending when the interrupt source signals for IRQ2 to IRQ0 are generated.
3. The IRQ2 to IRQ0 interrupt requests are simultaneously enabled and the order of the processing routines for the IRQ2 to IRQ0 interrupts is confirmed.
4. Table 1 shows the order of priority for the IRQ2 to IRQ0 interrupts in the case of interrupt control mode 0. Table 2 shows the setting example of the priority order for the IRQ2 to IRQ0 interrupts in interrupt control mode 2.

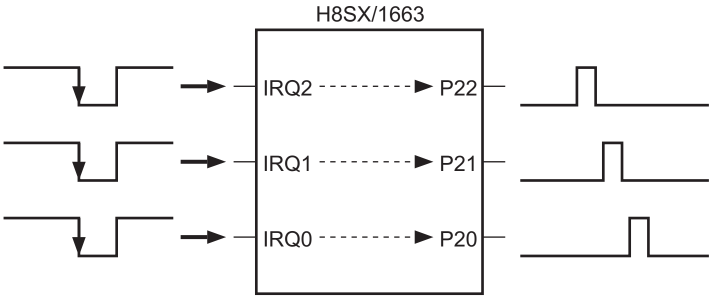


**Figure 1 Confirmation of Interrupt Controller Operation**

**Table 1 Order of Priority in Interrupt Control Mode 0**

| Interrupt | Vector Number | Priority Order |
|---|---|---|
| IRQ0 | 64 | Higher |
| IRQ1 | 65 | ↑ |
| IRQ2 | 66 | Lower |

**Table 2 Settings for Order of Priority in Interrupt Control Mode 2**

| Interrupt | IPR Setting | Priority Order | Setting for the Interrupt-Request Mask Level Bits (I2 to I0) in the EXR |
|---|---|---|---|
| IRQ0 | 4 | Lower | 4 |
| IRQ1 | 5 | ↓ | |
| IRQ2 | 6 | Higher | |

## 2. Applicable Conditions

**Table 3 Applicable Conditions**

| Item | Description |
|---|---|
| Operating frequency | Input clock: 12 MHz<br>System clock (Iϕ): 48 MHz<br>Peripheral-module clock (Pϕ): 24 MHz<br>External bus clock (Bϕ): 48 MHz |
| Operating mode | Mode 6 (MD2 = 1, MD1 = 1, MD0 = 0) |

## 3. Description of Modules Used

### 3.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests other than the NMI are maskable by the I bit of the CPU’s CCR. Figure 2 is a flowchart that illustrates the operation of interrupt acceptance in this mode.

1. If the interrupt enable bit has been set to 1 for a generated interrupt condition, the interrupt request is sent to the interrupt controller.
2. If the I bit of the CCR has been set to 1, the interrupt controller holds any interrupt other than the NMI pending. Once the I bit has been cleared to 0, the interrupt request is accepted.
3. In the case of multiple interrupts, the interrupt controller selects the interrupt request with the highest priority level in accord with the order of priority, requests that the CPU process that interrupt, and holds other requests pending.
4. When the CPU receives the interrupt request, it starts processing the interrupt exception on completion of processing for any instruction being executed at the time.
5. By interrupt exception handling, the contents of the PC and CCR are saved on the stack. In the case of the PC, the value saved is the address of the first instruction to be executed on return from interrupt handling.
6. The I bit of the CCR is set to 1. The interrupt requests other than the NMI are masked.
7. The CPU generates the address of the vector that corresponds to the interrupt request it has accepted, reads the first address of the interrupt handling routine from the vector table, and starts processing the interrupt.

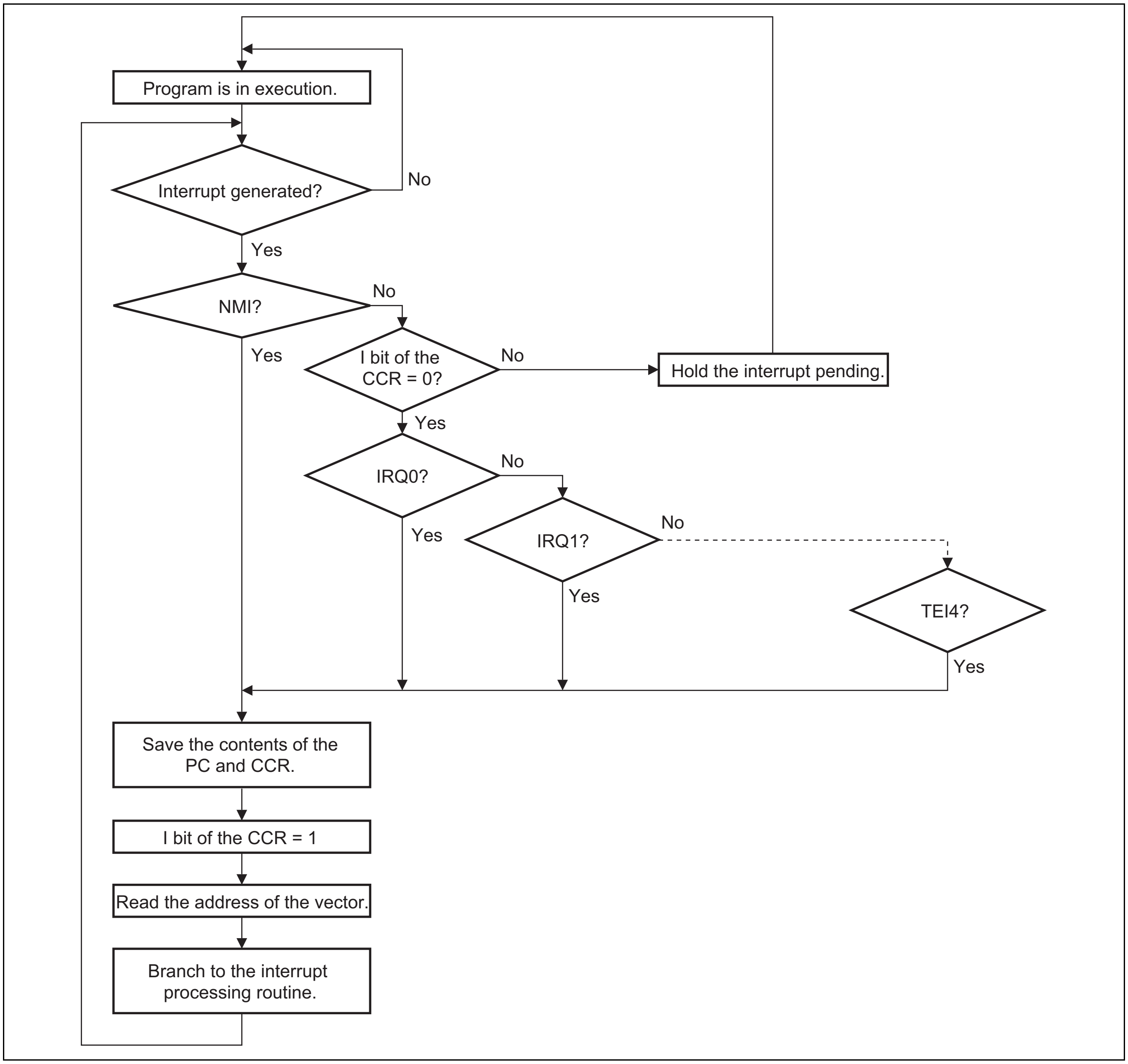

Program is in execution.
Interrupt generated?
No
Yes
NMI?
No
Yes
I bit of the CCR = 0?
No
Hold the interrupt pending.
Yes
IRQ0?
No
Yes
IRQ1?
No
Yes
TEI4?
Yes
Save the contents of the PC and CCR.
I bit of the CCR = 1
Read the address of the vector.
Branch to the interrupt processing routine.

**Figure 2 Flow up to Interrupt Acceptance in Interrupt Control Mode 0**

## 3.2 Interrupt Control Mode 2

In interrupt control mode 2, control of interrupt requests other than the NMI is performed on the basis of eight possible mask levels through comparison of the IPR and the interrupt mask level in the EXR (bits I2 to I0) of the CPU. Figure 3 is a flowchart that illustrates the operation of interrupt acceptance in this mode.

1. If the interrupt enable bit has been set to 1 for a generated interrupt condition, the interrupt request is sent to the interrupt controller.
2. In the case of multiple interrupt requests, the interrupt with the highest priority level according to the interrupt priority order for the interrupts as set in the IPRs of the interrupt controller is selected, and interrupt requests with lower priority levels are held pending. When the priority levels are the same, one interrupt request is selected in accord with the default order of priority.
3. The priority level of the selected interrupt request is compared with the interrupt mask level set in bits I2 to I0 of the EXR. If the priority level is below the mask level that has been set, the interrupt request is held pending. If the priority level is higher than the interrupt mask level, interrupt processing by the CPU is requested.
4. When the CPU receives the interrupt request, it starts processing the interrupt exception on completion of processing for any instruction being executed at the time.
5. By interrupt exception handling, the contents of the CCR, PC, and EXR are saved on the stack. In the case of the PC, the value saved is the address of the first instruction to be executed on return from interrupt handling.
6. The T bit of the EXR is cleared. The interrupt mask level can be overwritten by the priority level of the received interrupt. When a received interrupt is the NMI, the interrupt mask level is set to H'7.
7. The CPU generates the address of the vector that corresponds to the interrupt request it has accepted, reads the first address of the interrupt handling routine from the vector table, and starts processing the interrupt.

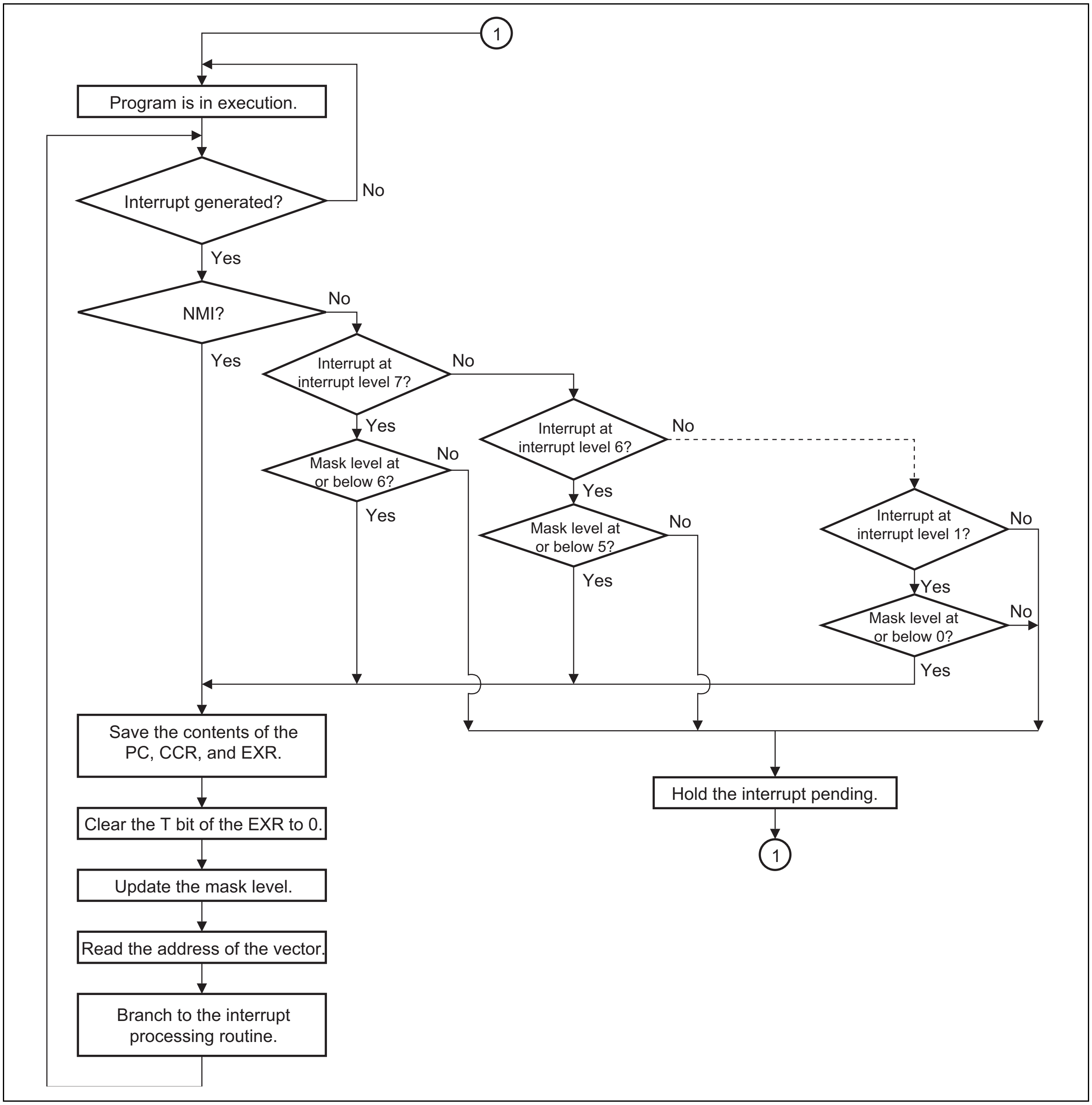

1
Program is in execution.
Interrupt generated?
No
Yes
NMI?
No
Yes
Interrupt at interrupt level 7?
No
Yes
Mask level at or below 6?
No
Yes
Interrupt at interrupt level 6?
No
Yes
Mask level at or below 5?
No
Yes
Interrupt at interrupt level 1?
No
Yes
Mask level at or below 0?
No
Yes
Save the contents of the PC, CCR, and EXR.
Clear the T bit of the EXR to 0.
Update the mask level.
Read the address of the vector.
Branch to the interrupt processing routine.
Hold the interrupt pending.
1

**Figure 3 Flow up to Interrupt Acceptance in Interrupt Control Mode 2**

## 3.3 IRQ Interrupts

An IRQn interrupt is requested by a signal input on pin $\overline{\text{IRQn}}$ (n = 11 to 0). IRQn (n = 11 to 0) interrupts have the following features.

- The ISCR registers can be used to select whether each interrupt request is generated by the input of a low level, falling edge, rising edge, or either edge on the corresponding $\overline{\text{IRQn}}$ pin.
- An IRQn interrupt request is enabled or disabled by the setting of the corresponding bit in the IER.
- The interrupt priority levels are set in the IER registers.
- The states of all IRQn interrupt requests are indicated in the ISR. ISR flags can be cleared to 0 by software. Either bit-manipulation or memory-operation instructions can be used to clear the flags.

Detection of IRQn interrupts is enabled through the P1ICR, P2ICR, and P5ICR register settings, and this is independent of setting of the corresponding pins as outputs. However, when a pin is in use as an external interrupt input pin, do not clear the corresponding DDR bit to 0 so that it can be used as an I/O pin for another function.

## 3.4 Stack Configuration

The stack configuration for the PC (program counter), CCR, and EXR values during exception processing (interrupt processing) in advanced mode is shown below.

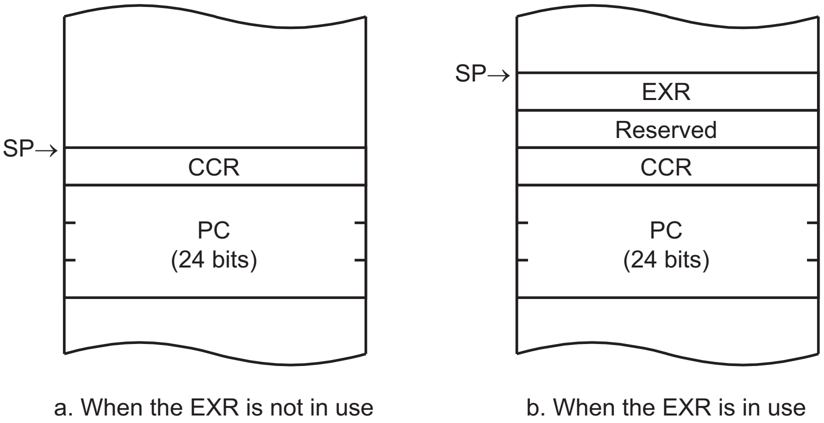


**Figure 4 Stack Configuration for Exception Processing in Advanced Mode**

## 3.5 Extended Control Register (EXR)

The EXR is an 8-bit register that contains the trace bit (T) and three interrupt mask bits (I2 to I0). Operations can be performed on the EXR bits by the LDC, STC, ANDC, ORC, and XORC instructions. For details, see section 4, Exception Handling.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | T | 0 | R/W | Trace Bit<br>When this bit is set to 1, a trace exception is generated each time an instruction is executed. When this bit is cleared to 0, instructions are executed in sequence. |
| 6 to 3 | — | All 1 | R/W | Reserved; these bits are always read as 1. |
| 2 to 0 | I2 | 1 | R/W | These bits designate the interrupt mask level (from 0 to 7). |
| | I1 | 1 | R/W | |
| | I0 | 1 | R/W | |

# 4. Principles of Operation

## 4.1 Interrupt Control Mode 0

When multiple interrupt requests are generated while the controller is in interrupt control mode 0, requests for interrupt processing by the CPU are in accord with the default order of interrupt priority. When the IRQ2 to IRQ0 interrupts are simultaneously generated in this sample task, operation is in the order given against points 1 to 3 below.

1. Execution branches to the interrupt processing routine for IRQ0 (vector number 64), which has the highest priority level, and a high-level trigger pulse is output on pin P20.
2. Execution branches to the interrupt processing routine for IRQ1 (vector number 65) and a high-level trigger pulse is output on pin P21.
3. Execution branches to the interrupt processing routine for IRQ2 (vector number 66) and a high-level trigger pulse is output on pin P22.

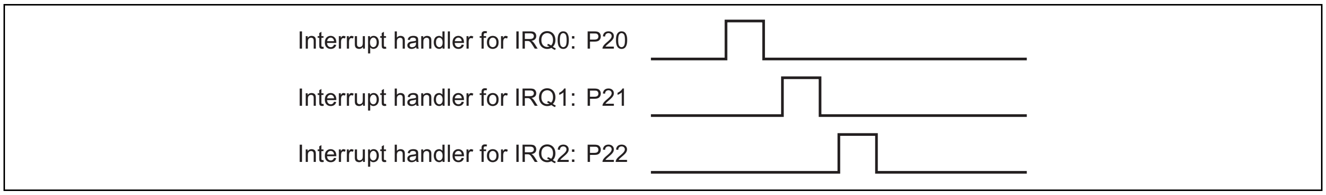

Interrupt handler for IRQ0: P20
Interrupt handler for IRQ1: P21
Interrupt handler for IRQ2: P22

**Figure 5 Example of Operation in Interrupt Control Mode 0**

## 4.2 Interrupt Control Mode 2

When multiple interrupt requests are generated while the controller is in interrupt control mode 2, the selected request for interrupt processing is that with the highest priority in accord with the order of interrupt priority set in IPR. The priority level of the selected interrupt request is compared with the interrupt mask level in EXR. If the priority level is at or below the mask level, the request is held pending; if the priority level is above the mask level, interrupt processing by the CPU is requested. When the IRQ2 to IRQ0 interrupts are simultaneously generated in this sample task, operation is in the order given against points 1 to 3 below.

1. Execution branches to the interrupt processing routine for IRQ2 (which has the highest priority level with an IPR setting of 6), and a high-level trigger pulse is output on pin P22.
2. Execution branches to the interrupt processing routine for IRQ1 (with an IPR setting of 5) and a high-level trigger pulse is output on pin P21.
3. Processing of IRQ0 is held pending because its IPR setting of 4 is at or below the interrupt mask level (EXR = 4), and there is no branch to the interrupt processing routine at this time.

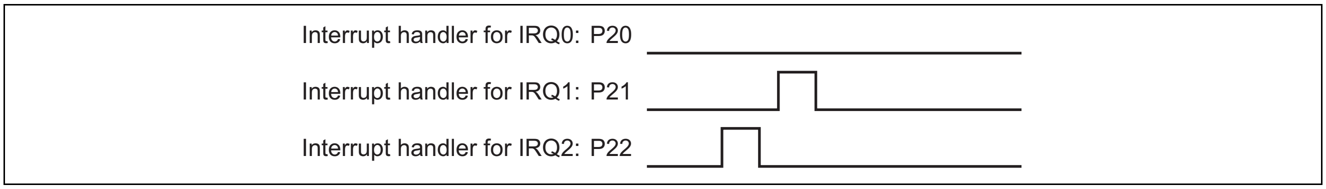

Interrupt handler for IRQ0: P20
Interrupt handler for IRQ1: P21
Interrupt handler for IRQ2: P22

**Figure 6 Example of Operation in Interrupt Control Mode 2**

# 5. Description of Software

## 5.1 Operating Environment

Table 4 Operating Environment

| Item | Details |
|---|---|
| Development tool | High-performance Embedded Workshop Ver.4.01.01 |
| C/C++ compiler | H8S, H8/300 SERIES C/C++ Compiler Ver.6.01.02<br>(manufactured by Renesas Technology) |
| Compiler options | -cpu = h8sxa:24:md, -code = machinecode, -optimize = 1, -regparam = 3<br>-speed = (register, shift, struct, expression) |

Table 5 Section Settings

| Address | Section Name | Description |
|---|---|---|
| H'001000 | P | Program area |

Table 6 Vector Table for Interrupt Exception Handling

| Source for Exception Handling | Vector Number | Address in Vector Table | Destination Function |
|---|---|---|---|
| Reset | 0 | H'000000 | init |
| IRQ0 | 64 | H'000100 | irq0_int |
| IRQ1 | 65 | H'000104 | irq1_int |
| IRQ2 | 66 | H'000108 | irq2_int |

## 5.2 List of Functions

Table 7 Functions in File main.c

| Function Name | Purpose |
|---|---|
| init | Initialization routine<br>Sets the CCR and configures the clocks, releases the required modules from the module stop mode, and calls the main function. |
| main | Main routine<br>Makes interrupt settings and initializes interrupts IRQ0 to IRQ2. |
| irq0_int | Interrupt handler for IRQ0; produces a high-level trigger pulse on pin P20. |
| irq1_int | Interrupt processing for IRQ1; produces a high-level trigger pulse on pin P21. |
| irq2_int | Interrupt processing for IRQ2; produces a high-level trigger pulse on pin P22. |

## 5.3 Macro Definitions

Table 8 Macro Definitions

| Defined Name | Description | Used in Function |
|---|---|---|
| INTMODE_0 | Create the program for confirming operation in interrupt control mode 0. | main |
| INTMODE_2 | Create the program for confirming operation in interrupt control mode 2. | main |

## 5.4 Description of Functions

### 5.4.1 Function init

1. Functional overview

   Initialization routine, which releases the required modules from module stop mode, configures the clocks, and calls the main function.

2. Arguments

   None

3. Return value

   None

4. Description of internal registers

   The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Mode control register (MDCR) Number of bits: 16 Address: H'FFFDC0

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 15 | MDS7 | Undefined* | R | Indicates the value set by a mode pin (MD3).<br>When MDCR is read, the input level on the MD3 pin is latched. This latch is released by a reset. |
| 11 | MDS3 | Undefined* | R | Mode Select 3 to 0<br>These bits indicate the operating mode selected by the mode pins (MD2 to MD0; see table 9). When MDCR is read, the signal levels input on pins MD2 to MD0 are latched into these bits. The latch is released by a reset. |
| 10 | MDS2 | Undefined* | R | |
| 9 | MDS1 | Undefined* | R | |
| 8 | MDS0 | Undefined* | R | |

Note: * Determined by the settings on pins MD0 to MD3.

**Table 9 Values of Bits MDS3 to MDS0**

| MCU Operating Mode | Mode Pins | | | MDCR | | | |
|---|---|---|---|---|---|---|---|
| | MD2 | MD1 | MD0 | MDS3 | MDS2 | MDS1 | MDS0 |
| 2 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 5 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 6 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |

- System clock control register (SCKCR) Number of bits: 16 Address: H'FFFDC4

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 10 | ICK2 | 0 | R/W | System Clock (Iϕ) Select<br>These bits select the frequency of the system clock provided to the CPU, DMAC, and DTC.<br>000: Input clock × 4 |
| 9 | ICK1 | 0 | R/W | |
| 8 | ICK0 | 0 | R/W | |
| 6 | PCK2 | 0 | R/W | Peripheral Module Clock (Pϕ) Select<br>These bits select the frequency of the peripheral module clock.<br>001: Input clock × 2 |
| 5 | PCK1 | 0 | R/W | |
| 4 | PCK0 | 1 | R/W | |
| 2 | BCK2 | 0 | R/W | External Bus Clock (Bϕ) Select<br>These bits select the frequency of the external bus clock.<br>000: Input clock × 4 |
| 1 | BCK1 | 0 | R/W | |
| 0 | BCK0 | 0 | R/W | |

- MSTPCRA, MSTPCRB and MSTPCRC control the module stop mode. Setting a bit to 1 makes the corresponding module enter the module stop mode, while clearing the bit to 0 releases the module from module stop state.
- Module stop control register A (MSTPCRA)  Number of bits: 16  Address: H'FFFDC8

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 15 | ACSE | 0 | R/W | All-Module-Clock-Stop Mode Enable<br>Enables/disables all-module-clock-stop mode for reducing current drawn by stopping operation of the bus controller and I/O ports when the CPU executes the SLEEP instruction after the module stop state has been set for all of the on-chip peripheral modules controlled by MSTPCR.<br>0: Disables all-module-clock-stop mode.<br>1: Enables all-module-clock-stop mode. |
| 13 | MSTPA13 | 1 | R/W | DMA controller (DMAC) |
| 12 | MSTPA12 | 1 | R/W | Data transfer controller (DTC) |
| 9 | MSTPA9 | 1 | R/W | 8-bit timer unit (TMR_3 and TMR_2) |
| 8 | MSTPA8 | 1 | R/W | 8-bit timer unit (TMR_1 and TMR_0) |
| 5 | MSTPA5 | 1 | R/W | D/A converter (channels 1 and 0) |
| 3 | MSTPA3 | 1 | R/W | A/D converter (unit 0) |
| 0 | MSTPA0 | 1 | R/W | 16-bit timer pulse unit (TPU channels 5 to 0) |

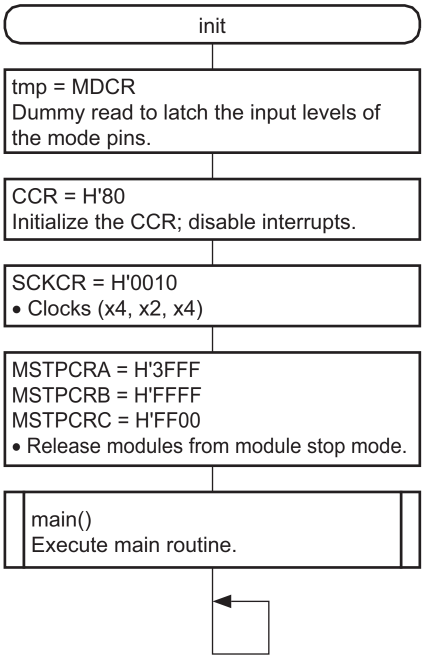
- Module stop control register B (MSTPCRB)  Number of bits: 16  Address: H'FFFDCA

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 15 | MSTPB15 | 1 | R/W | Programmable pulse generator (PPG) |
| 12 | MSTPB12 | 1 | R/W | Serial communications interface_4 (SCI_4) |
| 10 | MSTPB10 | 1 | R/W | Serial communications interface_2 (SCI_2) |
| 9 | MSTPB9 | 1 | R/W | Serial communications interface_1 (SCI_1) |
| 8 | MSTPB8 | 1 | R/W | Serial communications interface_0 (SCI_0) |
| 7 | MSTPB7 | 1 | R/W | $I^2C$ bus interface_1 (IIC_1) |
| 6 | MSTPB6 | 1 | R/W | $I^2C$ bus interface_0 (IIC_0) |

- Module stop control register C (MSTPCRC)  Number of bits: 16  Address: H'FFFDCC

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 15 | MSTPC15 | 1 | R/W | Serial communications interface_5 (SCI_5) , (IrDA) |
| 14 | MSTPC14 | 1 | R/W | Serial communications interface_6 (SCI_6) |
| 13 | MSTPC13 | 1 | R/W | 8-bit timer unit (TMR_4, TMR_5) |
| 12 | MSTPC12 | 1 | R/W | 8-bit timer unit (TMR_6, TMR_7) |
| 11 | MSTPC11 | 1 | R/W | Universal serial bus interface (USB) |
| 10 | MSTPC10 | 1 | R/W | Cyclic redundancy check |
| 4 | MSTPC4 | 0 | R/W | On-chip RAM_4 (H'FF2000 to H'FF3FFF) |
| 3 | MSTPC3 | 0 | R/W | On-chip RAM_3 (H'FF4000 to H'FF5FFF) |
| 2 | MSTPC2 | 0 | R/W | On-chip RAM_2 (H'FF6000 to H'FF7FFF) |
| 1 | MSTPC1 | 0 | R/W | On-chip RAM_1 (H'FF8000 to H'FF9FFF) |
| 0 | MSTPC0 | 0 | R/W | On-chip RAM_0 (H'FFA000 to H'FFBFFF) |

5. Flowchart



init
tmp = MDCR
Dummy read to latch the input levels of the mode pins.
CCR = H'80
Initialize the CCR; disable interrupts.
SCKCR = H'0010
• Clocks (x4, x2, x4)
MSTPCRA = H'3FFF
MSTPCRB = H'FFFF
MSTPCRC = H'FF00
• Release modules from module stop mode.
main()
Execute main routine.

### 5.4.2 Function main

1. Functional overview
   Makes interrupt settings and initializes interrupts IRQ2 to IRQ0

2. Arguments
   None

3. Return value
   None

4. Description of internal registers
   The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Port 2 data direction register (P2DDR) Number of bits: 8 Address: H'FFFB81

Function: Sets pins P22, P21, and P20 as output pins.
Setting: H'07

- Port 5 input buffer control register (P5ICR) Number of bits: 8 Address: H'FFFB94

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 2 | P52ICR | 1 | R/W | 0: Input buffer for pin P52 ($\overline{\text{IRQ2}}$-B) is disabled.<br>1: Input buffer for pin P52 ($\overline{\text{IRQ2}}$-B) is enabled. |
| 1 | P51ICR | 1 | R/W | 0: Input buffer for pin P51 ($\overline{\text{IRQ1}}$-B) is disabled.<br>1: Input buffer for pin P51 ($\overline{\text{IRQ1}}$-B) is enabled. |
| 0 | P50ICR | 1 | R/W | 0: Input buffer for pin P50 ($\overline{\text{IRQ0}}$-B) is disabled.<br>1: Input buffer for pin P50 ($\overline{\text{IRQ0}}$-B) is enabled. |

- Port function control register C (PFCRC) Number of bits: 8 Address: H'FFFBCC

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 2 | ITS2 | 1 | R/W | $\overline{\text{IRQ2}}$ Pin Select<br>0: Selects pin P12 as the $\overline{\text{IRQ2}}$-A input.<br>1: Selects pin P52 as the $\overline{\text{IRQ2}}$-B input. |
| 1 | ITS1 | 1 | R/W | $\overline{\text{IRQ1}}$ Pin Select<br>0: Selects pin P11 as the $\overline{\text{IRQ1}}$-A input.<br>1: Selects pin P51 as the $\overline{\text{IRQ1}}$-B input. |
| 0 | ITS0 | 1 | R/W | $\overline{\text{IRQ0}}$ Pin Select<br>0: Selects pin P10 as the $\overline{\text{IRQ0}}$-A input.<br>1: Selects pin P50 as the $\overline{\text{IRQ0}}$-B input. |

- Interrupt priority register A (IPRA)   Number of bits: 16   Address: H'FFFD40

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 14 | IPR14 | 1 | R/W | Sets the priority level for the IRQ0 interrupt |
| 13 | IPR13 | 0 | R/W | 000: Priority level 0 (lowest) |
| 12 | IPR12 | 0 | R/W | 001: Priority level 1 |
| | | | | 010: Priority level 2 |
| | | | | 011: Priority level 3 |
| | | | | 100: Priority level 4 |
| | | | | 101: Priority level 5 |
| | | | | 110: Priority level 6 |
| | | | | 111: Priority level 7 (highest) |
| 10 | IPR10 | 1 | R/W | Sets the priority level for the IRQ1 interrupt |
| 9 | IPR9 | 0 | R/W | 000: Priority level 0 (lowest) |
| 8 | IPR8 | 1 | R/W | 001: Priority level 1 |
| | | | | 010: Priority level 2 |
| | | | | 011: Priority level 3 |
| | | | | 100: Priority level 4 |
| | | | | 101: Priority level 5 |
| | | | | 110: Priority level 6 |
| | | | | 111: Priority level 7 (highest) |
| 6 | IPR6 | 1 | R/W | Sets the priority level for the IRQ2 interrupt |
| 5 | IPR5 | 1 | R/W | 000: Priority level 0 (lowest) |
| 4 | IPR4 | 0 | R/W | 001: Priority level 1 |
| | | | | 010: Priority level 2 |
| | | | | 011: Priority level 3 |
| | | | | 100: Priority level 4 |
| | | | | 101: Priority level 5 |
| | | | | 110: Priority level 6 |
| | | | | 111: Priority level 7 (highest) |

- IRQ sense control register L (ISCRL)   Number of bits: 16   Address: H'FFFD6A

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 5 | IRQ2SR | 0 | R/W | IRQ2 Sense Control Rise |
| 4 | IRQ2SF | 1 | R/W | IRQ2 Sense Control Fall |
| | | | | 01: The interrupt request is generated on falling edges of the $\overline{\text{IRQ2}}$ input. |
| 3 | IRQ1SR | 0 | R/W | IRQ1 Sense Control Rise |
| 2 | IRQ1SF | 1 | R/W | IRQ1 Sense Control Fall |
| | | | | 01: The interrupt request is generated on falling edges of the $\overline{\text{IRQ1}}$ input. |
| 1 | IRQ0SR | 0 | R/W | IRQ0 Sense Control Rise |
| 0 | IRQ0SF | 1 | R/W | IRQ0 Sense Control Fall |
| | | | | 01: The interrupt request is generated on falling edges of the $\overline{\text{IRQ0}}$ input. |

- Interrupt control register (INTCR) Number of bits: 8 Address: H'FFFF32

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 5 | INTM1 | B'00 or B'10 | R/W | Interrupt Control Selection Mode 1 and 0<br>These bits select the mode of control by the interrupt controller.<br>00: Interrupt control mode 0<br>Interrupts are controlled by the I bit in the CCR.<br>10: Interrupt control mode 2<br>Interrupts are controlled by bits I2 to I0 in EXR and by the IPR settings. |
| 4 | INTM0 |  | R/W |  |
| 3 | NMIEG | 0 | R/W | NMI Edge Selection<br>Selects the input edge for the NMI pin.<br>0: Interrupt requests are generated by falling edges of the NMI input.<br>1: Interrupt requests are generated by rising edges of the NMI input. |

- IRQ enable register (IER) Number of bits: 16 Address: H'FFFF34

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 2 | IRQ2E | 0 | R/W | IRQ2 Enable<br>0: Disables IRQ2 interrupt requests.<br>1: Enables IRQ2 interrupt requests. |
| 1 | IRQ1E | 0 | R/W | IRQ1 Enable<br>0: Disables IRQ1 interrupt requests.<br>1: Enables IRQ1 interrupt requests. |
| 0 | IRQ0E | 0 | R/W | IRQ0 Enable<br>0: Disables IRQ0 interrupt requests.<br>1: Enables IRQ0 interrupt requests. |

- IRQ status register (ISR) Number of bits: 16 Address: H'FFFF36

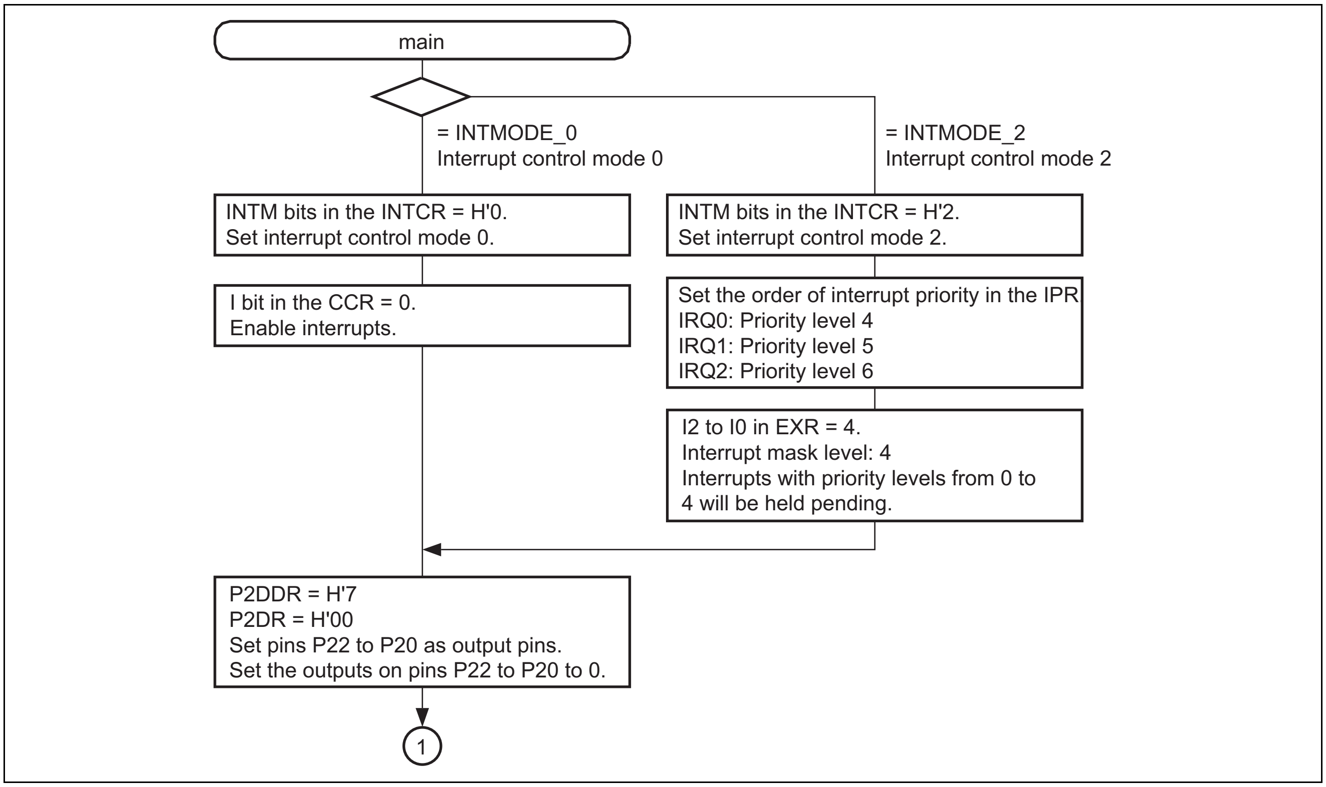
| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 11 | IRQ11F | 0 | R/(W)* | [Setting condition]<br>• Generation of an interrupt source selected in ISCR<br>[Clearing conditions]<br>• Writing 0 to IRQnF after reading it as 1<br>• Execution of interrupt exception handling while low-level sensing is selected and the $\overline{\text{IRQn}}$ input is high (n = 11 to 0)<br>• Execution of IRQn interrupt exception handling while falling-, rising-, or both-edge sensing is selected<br>• Activation of the DTC by an IRQn interrupt when the DISEL bit in MRB of the DTC is clear (0) |
| 10 | IRQ10F | 0 | R/(W)* |  |
| 9 | IRQ9F | 0 | R/(W)* |  |
| 8 | IRQ8F | 0 | R/(W)* |  |
| 7 | IRQ7F | 0 | R/(W)* |  |
| 6 | IRQ6F | 0 | R/(W)* |  |
| 5 | IRQ5F | 0 | R/(W)* |  |
| 4 | IRQ4F | 0 | R/(W)* |  |
| 3 | IRQ3F | 0 | R/(W)* |  |
| 2 | IRQ2F | 0 | R/(W)* |  |
| 1 | IRQ1F | 0 | R/(W)* |  |
| 0 | IRQ0F | 0 | R/(W)* |  |

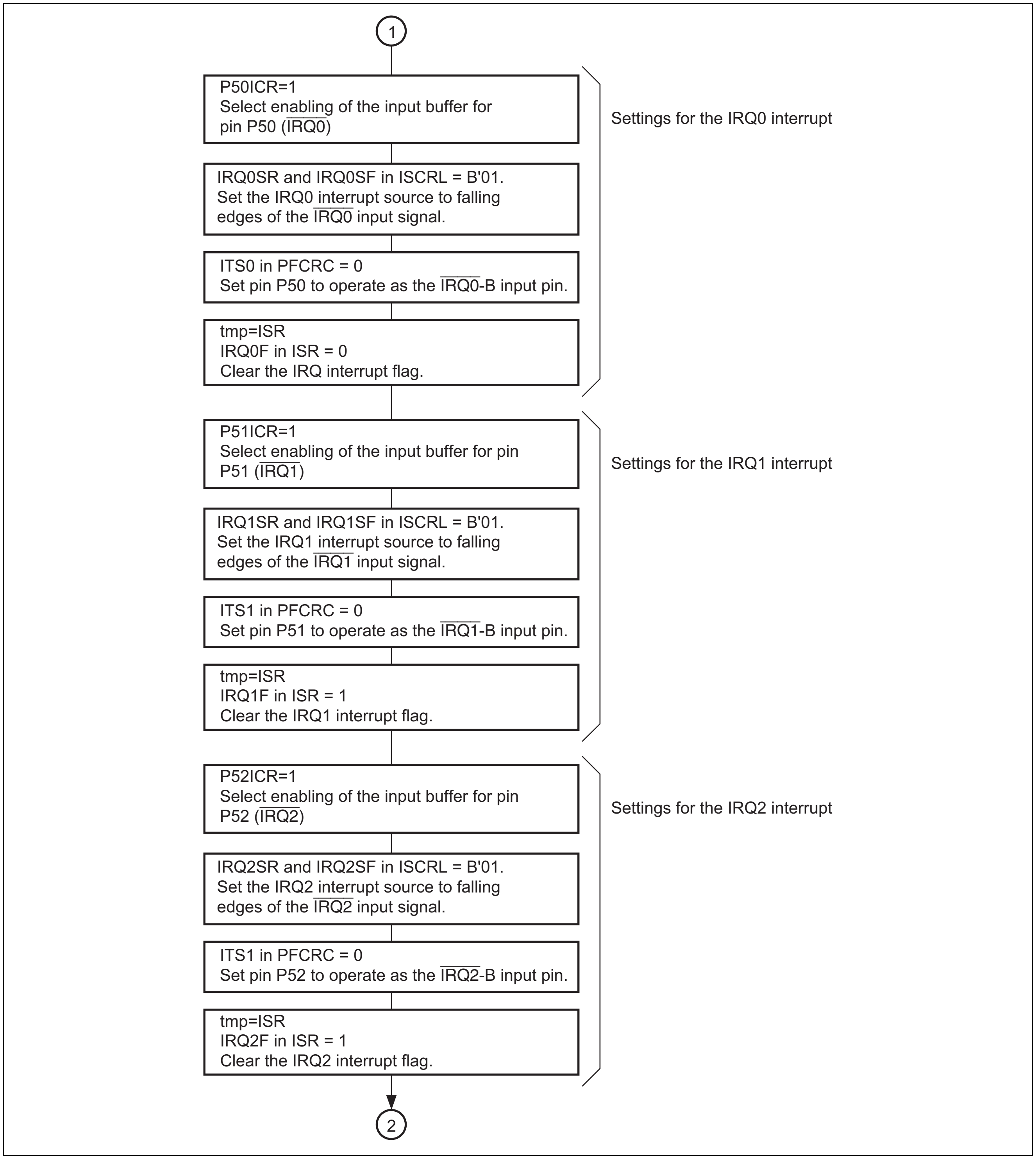
Note: * Only 0 can be written here, to clear the flag.

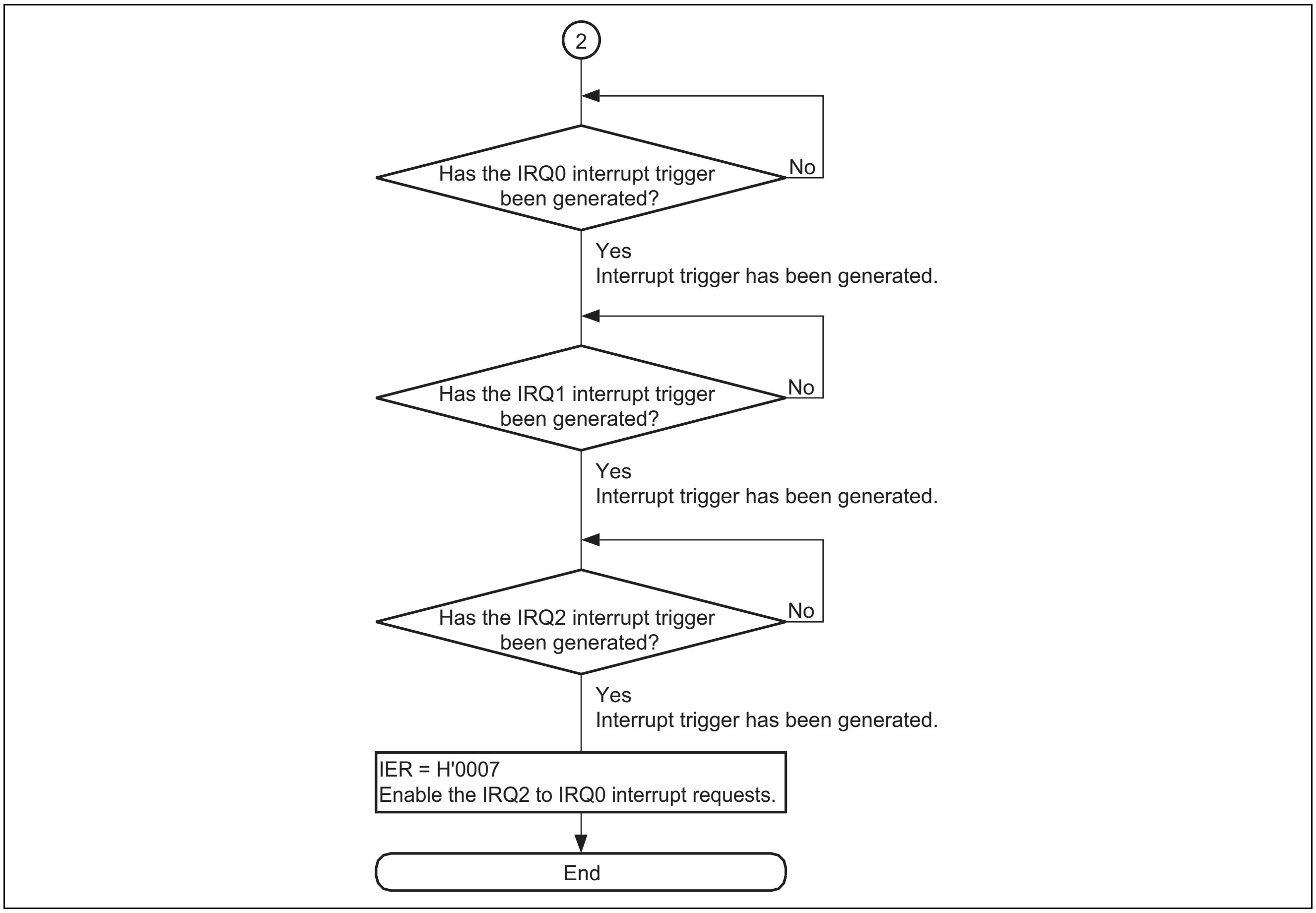
- Port 2 data register (P2DR)  Number of bits: 8  Address: H'FFFF51

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 2 | P22DR | 0 | R/W | 0: Pin P22 is at the low level.<br>1: Pin P22 is at the high level. |
| 1 | P21DR | 0 | R/W | 0: Pin P22 is at the low level.<br>1: Pin P22 is at the high level. |
| 0 | P20DR | 0 | R/W | 0: Pin P20 is at the low level.<br>1: Pin P20 is at the high level. |

5. Flowchart


main
= INTMODE_0
Interrupt control mode 0
= INTMODE_2
Interrupt control mode 2
INTM bits in the INTCR = H'0.
Set interrupt control mode 0.
INTM bits in the INTCR = H'2.
Set interrupt control mode 2.
I bit in the CCR = 0.
Enable interrupts.
Set the order of interrupt priority in the IPR
IRQ0: Priority level 4
IRQ1: Priority level 5
IRQ2: Priority level 6
I2 to I0 in EXR = 4.
Interrupt mask level: 4
Interrupts with priority levels from 0 to
4 will be held pending.
P2DDR = H'7
P2DR = H'00
Set pins P22 to P20 as output pins.
Set the outputs on pins P22 to P20 to 0.
1


1
P50ICR=1
Select enabling of the input buffer for pin P50 (IRQ0)
IRQ0SR and IRQ0SF in ISCRL = B'01.
Set the IRQ0 interrupt source to falling edges of the IRQ0 input signal.
ITS0 in PFCRC = 0
Set pin P50 to operate as the IRQ0-B input pin.
tmp=ISR
IRQ0F in ISR = 0
Clear the IRQ interrupt flag.
Settings for the IRQ0 interrupt
P51ICR=1
Select enabling of the input buffer for pin P51 (IRQ1)
IRQ1SR and IRQ1SF in ISCRL = B'01.
Set the IRQ1 interrupt source to falling edges of the IRQ1 input signal.
ITS1 in PFCRC = 0
Set pin P51 to operate as the IRQ1-B input pin.
tmp=ISR
IRQ1F in ISR = 1
Clear the IRQ1 interrupt flag.
Settings for the IRQ1 interrupt
P52ICR=1
Select enabling of the input buffer for pin P52 (IRQ2)
IRQ2SR and IRQ2SF in ISCRL = B'01.
Set the IRQ2 interrupt source to falling edges of the IRQ2 input signal.
ITS1 in PFCRC = 0
Set pin P52 to operate as the IRQ2-B input pin.
tmp=ISR
IRQ2F in ISR = 1
Clear the IRQ2 interrupt flag.
Settings for the IRQ2 interrupt
2


2
Has the IRQ0 interrupt trigger been generated?
No
Yes
Interrupt trigger has been generated.
Has the IRQ1 interrupt trigger been generated?
No
Yes
Interrupt trigger has been generated.
Has the IRQ2 interrupt trigger been generated?
No
Yes
Interrupt trigger has been generated.
IER = H'0007
Enable the IRQ2 to IRQ0 interrupt requests.
End

### 5.4.3 Function irq0_int

1. Functional overview
   This is the interrupt handler for IRQ0, which outputs a high-level trigger pulse on pin P20.

2. Arguments
   None

3. Return value
   None

4. Description of internal registers
   The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

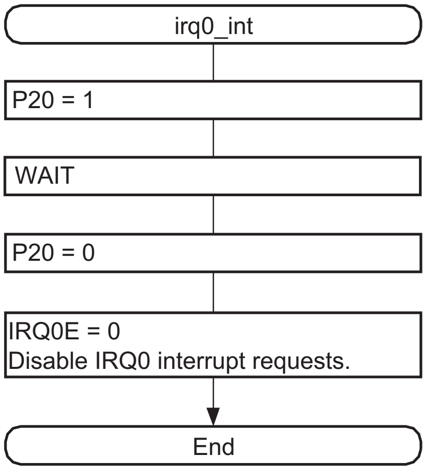
- IRQ enable register (IER) Number of bits: 16 Address: H'FFFF34

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 0 | IRQ0E | 0 | R/W | IRQ0 Enable<br>0: Disables IRQ0 interrupt requests.<br>1: Enables IRQ0 interrupt requests. |

- Port 2 data register (P2DR) Number of bits: 8 Address: H'FFFF51

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 0 | P20DR | 0/1 | R/W | 0: Pin P20 is at the low level.<br>1: Pin P20 is at the high level. |

5. Flowchart



### 5.4.4 Function irq1_int

1. Functional overview

   This is the handler for IRQ1, which outputs a high-level trigger pulse on pin P21.

2. Arguments

   None

3. Return value

   None

4. Description of internal registers

   The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

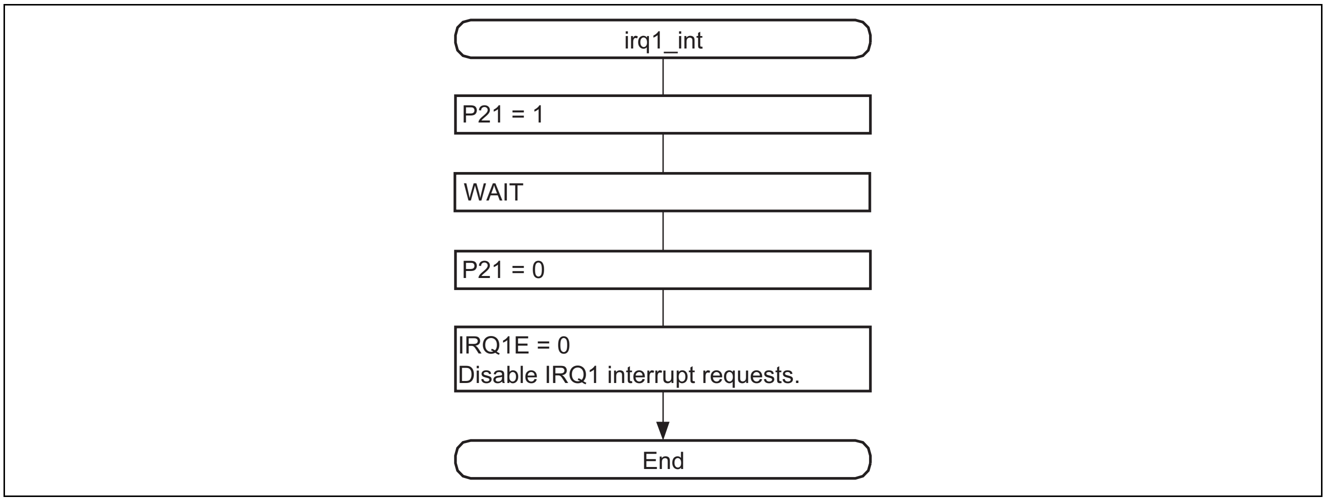
- IRQ enable register (IER) Number of bits: 16 Address: H'FFFF34

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 1 | IRQ1E | 0 | R/W | IRQ1 Enable<br>0: Disables IRQ1 interrupt requests.<br>1: Enables IRQ1 interrupt requests. |

- Port 2 data register (P2DR) Number of bits: 8 Address: H'FFFF51

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 1 | P21DR | 0/1 | R/W | 0: Pin P21 is at the low level.<br>1: Pin P21 is at the high level. |

5. Flowchart



### 5.4.5 Function irq2_int

1. Functional overview
   This is the interrupt handler for IRQ2, which outputs a high-level trigger pulse on pin P22.

2. Arguments
   None

3. Return value
   None

4. Description of internal registers
   The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

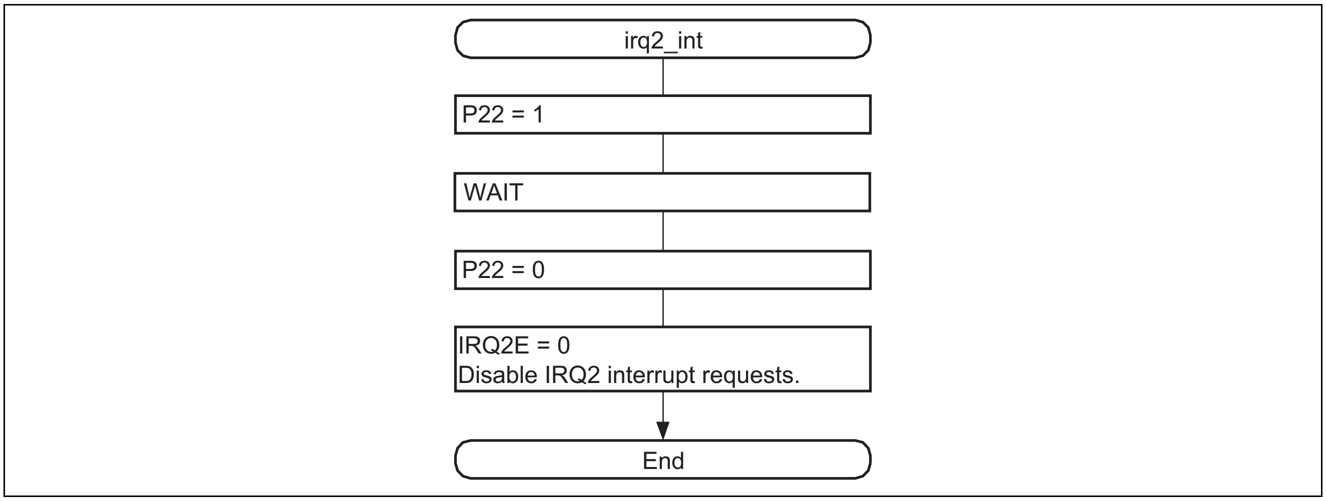
- IRQ enable register (IER) Number of bits: 16 Address: H'FFFF34

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 2 | IRQ2E | 0 | R/W | IRQ2 Enable<br>0: Disables IRQ2 interrupt requests.<br>1: Enables IRQ2 interrupt requests. |

- Port 2 data register (P2DR) Number of bits: 8 Address: H'FFFF51

| Bit | Bit Name | Setting | R/W | Description |
|---|---|---|---|---|
| 2 | P22DR | 0/1 | R/W | 0: Pin P22 is at the low level.<br>1: Pin P22 is at the high level. |

5. Flowchart



## 6. Note on Usage

When the pin of the device functions as an input for the peripheral modules, the corresponding bits of the input buffer control register (PnICR) should be set to 1. For details, see the hardware manual.

## Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

## Revision Record

| | | Description | |
|---|---|---|---|
| Rev. | Date | Page | Summary |
| 1.00 | Jul.20.07 | — | First edition issued |
| 2.00 | Mar.07.08 | 1, 23 | Page 1: Target devices added<br>Page 23: Note on usage added |
| | | | |
| | | | |
| | | | |
| | | | |

All trademarks and registered trademarks are the property of their respective owners.

## Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.