32-Bit RISC Microcontroller

TXZ+ Family TMPM4L Group (1)

Reference Manual Exception (EXCEPT-M4L(1))

Revision 1.0

2026-01

Toshiba Electronic Devices & Storage Corporation

Contents

List of Figures

List of Tables

Preface

Related Document

Document name
Oscillation Frequency Detector
Clock Selective Watchdog Timer
Voltage Detection Circuit
Secure Access Memory
Clock Control and Operation Mode
Arm ® Cortex®-M4 Processor Technical Reference Manual

Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal: 0xABC
  Decimal: 123 or 0d123 - Only when it needs to be explicitly shown that they are decimal numbers.
  Binary: 0b111 - It is possible to omit the "0b" when the number of bits can be distinctly understood from a sentence.
- "_N" is added to the end of signal names to indicate low active signals.
- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.
- When two or more signal names are referred, they are described like as [m:n].
  Example: S[3:0] shows four signal names S3, S2, S1 and S0 together.
- The characters surrounded by ***[ ]*** defines the register.
  Example: ***[ABCD]***
- "N" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: ***[XYZ1]***, ***[XYZ2]***, ***[XYZ3]*** → ***[XYZn]***
- "x" substitutes suffix number or character of units and channels in the register list.
- In case of unit, "x" means A, B, and C, ...
  Example: ***[ADACR0]***, ***[ADBCR0]***, ***[ADCCR0]*** → ***[ADxCR0]***
- In case of channel, "x" means 0, 1, and 2, ...
  Example: ***[T32A0RUNA]***, ***[T32A1RUNA]***, ***[T32A2RUNA]*** → ***[T32AxRUNA]***
- The bit range of a register is written like as [m: n].
  Example: Bit[3: 0] expresses the range of bit 3 to 0.
- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: ***[ABCD]***<EFG> = 0x01 (hexadecimal), ***[XYZn]***<VW> = 1 (binary)
- Word and byte represent the following bit length.
  Byte: 8 bits
  Half word: 16 bits
  Word: 32 bits
  Double word: 64 bits
- Properties of each bit in a register are expressed as follows:
  R: Read only
  W: Write only
  R/W: Read and write are possible.
- Unless otherwise specified, register access supports only word access.
- The register defined as "Reserved" must not be rewritten. Moreover, do not use the read value.
- The value read from the bit having default value of "-" is unknown.
- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, In the cases that default is "-", follow the definition of each register.
- Reserved bits of the write-only register should be written with their default value. In the cases that default is "-", follow the definition of each register.
- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

Terms and Abbreviations

Some of abbreviations used in this document are as follows:

IA	Interrupt Control Register A
IB	Interrupt Control Register B
IMCxx	Interrupt Mode Control xx
IMNFLGNMI	Interrupt Monitor Flag NMI
IMNFLGx	Interrupt Monitor Flag x
INT	Interrupt
INTIF	Interrupt Interface Logic
NICxx	Non-maskable Interrupt Control xx
NVIC	Nested Vectored Interrupt Controller
POR	Power-on Reset Circuit
PORF	Power-on Reset Circuit for Flash and Debug
RLMRSTFLGx	RLM Reset Flag x
SAM	Secure Access Memory

1. Outlines

Exceptions require CPU to suspend the currently executing process, and to start another process.

There are two types of exceptions: those that are generated when some error condition occurs or when an instruction to generate an exception is executed; and those that are generated by hardware, such as an interrupt request signal from an external pin or peripheral function.

All exceptions are handled by the Nested Vectored Interrupt Controller (NVIC) in the CPU according to the respective priority levels. When an exception occurs, the CPU stores the current state to the stack and branches to the corresponding interrupt service routine (ISR). Upon completion of the ISR, the information stored to the stack is automatically restored.

1.1. Exception Types

This product has the following types of exceptions.

For detailed descriptions on each exception, refer to "Arm Cortex-M4 Processor Technical reference manual".

- Reset
- Non-maskable interrupt (NMI)
- Hard Fault
- Memory Management
- Bus Fault
- Usage Fault
- SVCall (Supervisor Call)
- Debug Monitor
- PendSV
- SysTick
- External interrupt

1.2. Exception Handling Flowchart

The following shows how an exception/interrupt is handled. In the following descriptions, exception handling by hardware and that by software are explained.

Each step is described later in this reference manual.

Process	Description	Reference
Detection by INTIF/CPU	The INTIF or CPU detects the exception request.	Section 1.2.1
⇩		
Handling exception by CPU	The CPU executes exception process.	Section 1.2.2
⇩		
Branch to ISR	The CPU branches to the interrupt service routine (ISR) corresponding to the exception.	Section 1.2.2
⇩		
Execution of ISR	Necessary processing is executed	Section 1.2.3
⇩		
Return from exception	The CPU branches to another ISR or returns to the previous program.	Section 1.2.4

1.2.1. Exception Request and Detection

1.2.1.1. Exception Occurrence

Exception occurs by the sources include instruction execution by the CPU, memory accesses, and interrupt requests from external interrupt pins or peripheral functions.

An exception by the instruction execution occurs when the CPU executes an instruction that causes an exception or when an error condition occurs during instruction execution.

An exception also occurs by an instruction fetch from the Execute Never region or an access violation to the Fault region.

The request of the exception by the external interrupt pin or the peripheral function occurs by each functional factor. Regarding to interrupt which connected via INTIF, the setup of the interrupt control register is needed. For details, refer to the chapter, "4. Interrupts".

1.2.1.2. Exception Detection

If several exceptions occur simultaneously, the CPU takes the exception with the highest priority depending on the priority level of exceptions.

Table 1.1 shows the priority of exceptions. "Configurable" means that the exception can be set priority level. Memory Management, Bus Fault and Usage Fault exceptions can be enabled or disabled.

If the disabled exception occurs, it is handled as Hard Fault.

Table 1.1 Exception Types and Priority Level

Exception type	Priority level	Factor generates exception	Offset
Reset	-3 (highest)	Reset pin, POR reset, PORF reset, OFD reset, SIWDT reset, LVD reset, SAM reset, SYSRESETREQ reset, and LOCKUP reset	0x00
Non-maskable interrupt	-2	SIWDT, LVD	0x08
Hard Fault	-1	Fault that cannot activate because a higher-priority fault is being handled or it is disabled	0x0C
Memory Management	Configurable	Exception from the Memory Protection Unit (MPU) Instruction fetch from the Execute Never (XN) region	0x10
Bus Fault	Configurable	Access violation to the Hard Fault region of the memory map	0x14
Usage Fault	Configurable	Undefined instruction execution or other faults related to instruction execution	0x18
Reserved	-	-	0x1C to 0x28
SVCall	Configurable	System service call with SVC instruction	0x2C
Debug Monitor	Configurable	Debug monitor when the CPU is not faulting	0x30
Reserved	-	-	0x34
PendSV	Configurable	Pending system service request	0x38
SysTick	Configurable	Notification from system timer	0x3C
External interrupt	Configurable	External interrupt pin or peripheral function (Note)	0x40

Note: External interrupt s have different sources and numbers in each product. For details, see "4.4 List of Interrupt Sources".

1.2.1.3. Priority Setting

(a) Priority Level

The external interrupt priority level is set to the interrupt priority register and other exceptions are set to <PRI_n> bit in the system handler priority register.
The configuration <PRI_n> can be changed, and the number of bits required for setting the priority level from 3 bits to 8 bits depending on products. Thus, the settable range of priority level is different depending on products.
In the case of 8-bit configuration, the priority level can be configured in the range from 0 to 255. The highest priority is "0". If several factors are set the same priority, the factor which has the smaller number becomes the higher priority level.
TMPM4L Group (1) has the upper 4 bits of <PRI_n>. The priority level can be configured in the range from 0 to 15.

(b) Priority Grouping

The priority level can be grouped. By setting the <PRIGROUP> of the application interrupt and reset control register, <PRI_n> can be split into the pre-emption priority and the sub priority.
At first, the priorities are compared with the pre-emption priority. If the priorities are the same as the pre-emption priority, then they are compared with the sub priority. If the priorities are the same as the sub priority, oner of them which has the smaller exception number becomes the higher priority.
Table 1.2 shows the priority group setting. The number of pre-emption priority and the number of sub priority in the table are shown in the case that <PRI_n> is defined as an 8-bit configuration.

Table 1.2 Priority Grouping Setting

<PRIGROUP[2:0]> setting	<PRI_n[7:0]>		Number of pre-emption priorities	Number of sub priorities
	Pre-emption field	Sub priority field		
000	[7:1]	[0]	128	2
001	[7:2]	[1:0]	64	4
010	[7:3]	[2:0]	32	8
011	[7:4]	[3:0]	16	16
100	[7:5]	[4:0]	8	32
101	[7:6]	[5:0]	4	64
110	[7]	[6:0]	2	128
111	None	[7:0]	1	256

Note: If the configuration of <PRI_n> is less than 8 bits, the lower bit is "0". For the example in the case of 4-bit configuration, the priority is set as <PRI_n[7:4]> and <PRI_n[3:0]> is "0000".

1.2.2. Exception Handling and Branch to Interrupt Service Routine (Pre-emption)

When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

1.2.2.1. Stacking

When the CPU detects an exception, it pushes the contents of the following eight registers to the stack in the following order:

(a) Program Counter (PC)
(b) Program Status Register (xPSR)
(c) r0 to r3
(d) r12
(e) Link Register (LR)

The SP is decremented by eight words after the completion of the stack push. The following shows the state of the stack after the register contents have been pushed.

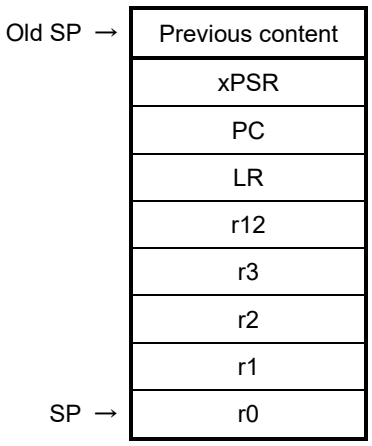


Figure 1.1 Stack After Register Push

1.2.2.2. Fetching ISR

The CPU performs instruction fetch for the ISP simultaneously with the pushing of registers.

Prepare a vector table containing the top addresses of ISRs for each exception. After reset, the vector table is located at address 0x00000000 in the Code area. The vector table is located at any address in the Code or SRAM space by setting Vector Table Offset Register.

The vector table should be also set the initial value of the main stack.

1.2.2.3. Late-arriving

If the CPU detects a higher priority exception before executing the ISR for a previous exception, the CPU handles the higher priority exception first. This is called "late-arriving".

A late-arriving exception causes the CPU to fetch an instruction for the new detected ISR once again and. the CPU branches to it. But the CPU does not newly push the register contents to the stack.

1.2.2.4. Vector Table Configuration

The vector table is configured as shown below.
Set the first four words (initial value of stack, reset ISR address, NMI ISR address, and Hard Fault ISR address).

For other exceptions, the ISR addresses are prepared if necessary.

Offset	Exception	Contents	Setting
0x00	Reset	Initial value of the main stack	Required
0x04	Reset	ISR address	Required
0x08	Non-maskable interrupt	ISR address	Required
0x0C	Hard Fault	ISR address	Required
0x10	Memory Management	ISR address	Optional
0x14	Bus Fault	ISR address	Optional
0x18	Usage Fault	ISR address	Optional
0x1C to 0x28	Reserved	-	-
0x2C	SVCall	ISR address	Optional
0x30	Debug Monitor	ISR address	Optional
0x34	Reserved	-	-
0x38	PendSV	ISR address	Optional
0x3C	SysTick	ISR address	Optional
0x40	External interrupt	ISR address	Optional

1.2.3. Handling ISR

An ISR performs the necessary processing for the corresponding the generated exception. ISRs must be prepared by the user.

An ISR may need to include code for clearing the interrupt request so that the same interrupt will not occur again upon return to normal program execution.
For details about interrupt handling, refer to "4 Interrupts".

If a higher priority exception occurs during ISR execution for the current exception, the CPU suspends the currently executing ISR and services the newly detected exception.

1.2.4. Exception Exit

1.2.4.1. Actions after Returning from ISR

When returning from an ISR, the CPU takes one of the following actions:

(f) Tail-chaining

If a pending exception exists and there are no stacked exceptions, or the pending exception has higher priority than all stacked exceptions, the CPU returns to the ISR of the pending exception.
In this case, the CPU skips the pop of eight registers and push of eight registers when exiting one ISR and entering another. This is called "tail-chaining".

(g) Returning to the last stacked ISR

If there are no pending exceptions or if the priority of stacked exception is higher priority than the pending exception, the CPU returns to the last stacked ISR.

(h) Returning to the previous program

If there are no pending or stacked exceptions, the CPU returns to the previous program.

1.2.4.2. Exception Exit Sequence

When returning from ISR, the CPU performs the following operations:

(a) Pop eight registers

Pop eight registers (PC, xPSR, r0 to r3, r12, and LR) from the stack and adjust the SP.

(b) Load current active interrupt number

Loads the current active interrupt number from the pushed xPSR. The CPU uses this to control which interrupt to return to.

(c) Select SP

If returning to an exception (Handler Mode), SP is SP_main. If returning to Thread Mode,
SP can be SP_main or SP_process.

2. Reset Exception

Reset exceptions are generated from the following sources.
Use the ***[RLMRSTFLGn]*** of the Reset Flag Register to identify the source of a reset exception.

- Reset exception by the reset pin

  A reset exception occurs when the reset pin changes from "Low" to "High".

- Reset exception by POR

  A reset exception occurs by POR. For details, refer to reference manual "Clock Control and Operation Mode"

- Reset exception by OFD

  A reset exception occurs by OFD. For details, refer to reference manual "Oscillation Frequency Detector".

- Reset exception by SIWDT

  A reset exception occurs by SIWDT. For details, refer to reference manual "Clock Selective Watchdog Timer".

- Reset exception by LVD

  A reset exception occurs by LVD. For details, refer to reference manual "Voltage Detector Circuit".

- Reset exception by PORF

  A reset exception occurs by PORF. For details, refer to reference manual "Clock Control and Operation Mode".

- SAM exception by SAM

  A reset exception occurs by SAM. In addition, in the Boot and Lib state, non-maskable interrupts become SAM reset. For details, refer to reference manual "Secure Access Memory".

- Reset exception by <SYSRESETREQ>

  A reset exception occurs by setting the <SYSRESETREQ> in the NVIC's application interrupt and reset control register.

- Reset exception by LOCKUP signal

  A reset exception occurs by the LOCKUP signal which can be output from the Cortex-M4 with FPU Processor when the un-recoverable interrupt occurs. For details of the LOCKUP signal, please refer to "Arm Cortex-M4 Processor Technical reference manual".

3. SysTick

SysTick provides interrupt function using the CPU's system timer.

When set a value to the SysTick Reload Value Register and enable the SysTick function in the SysTick Control and Status Register, the counter re-loads with the value set in the SysTick Reload Value Register and begins counting down. When the counter reaches "0", a SysTick exception occurs. The generated exception can be pended, and the flag can be monitored to know when the timer reaches "0".

4. Interrupts

This chapter explains the route which an interrupt request is transmitted, source ,and the required setup.

4.1. Non-maskable Interrupt (NMI)

Non-maskable interrupts are generated from the following sources.

- Non-maskable Interrupt by SIWDT

  The non-maskable interrupt occurs by SIWDT.
  For details, refer to reference manual "Clock Selective Watchdog Timer".

- Non-maskable Interrupt by LVD

  The non-maskable interrupt occurs by LVD.
  For details, refer to reference manual "Voltage Detector Circuit".

When SAM is enabled, non-maskable interrupts become SAM reset in Boot and Lib state. For details, refer to reference manual "Secure Access Memory".

4.2. Maskable Interrupt

Refer to interrupt control register A and interrupt control register B in "4.4 List of Interrupt Sources" for the sources of maskable interrupt .

4.3. Interrupt Request

The CPU is notified of interrupt requests by the interrupt request signal from each interrupt source. It sets priority on interrupts and handles an interrupt request with the highest priority.

4.3.1. Interrupt Route

The interrupt is available for releasing from a low power consumption mode, and a route varies according to a source.

Figure 4.1 shows the interrupt transfer route diagram and Table 4.1 shows the explanation of each interrupt transfer route.

- The interrupt that is releasable from IDLE, STOP1, and STOP2 mode

  Interrupt which can release from IDLE, STOP1, and STOP2 mode is controlled by the interrupt control register A in INTIF, and is notified to CPU via INTIF. (Route A, B, C)

- The interrupt that is releasable from IDLE, STOP1 mode

  Interrupt which can release from IDLE and the STOP1 mode is controlled by the interrupt control register B in INTIF, and is notified to CPU via INTIF. (Route D, E, F)

- The interrupt that is releasable from IDLE mode

  Although some factors of interrupt which can release from IDLE mode are controlled by the interrupt control register B via INTIF (Route G), other factors are notified to CPU directly (Route H) not passing through INTIF.

When the interrupt factor that goes by way of an interrupt regardless of low power consumption mode cancellation is used, setting of interrupt control register A or B is necessary.

Please refer to the chapter of " Release Source of Low-power Consumption Mode" of a reference manual "Clock Control and Operation Mode" for the details of a low power consumption mode release factor.

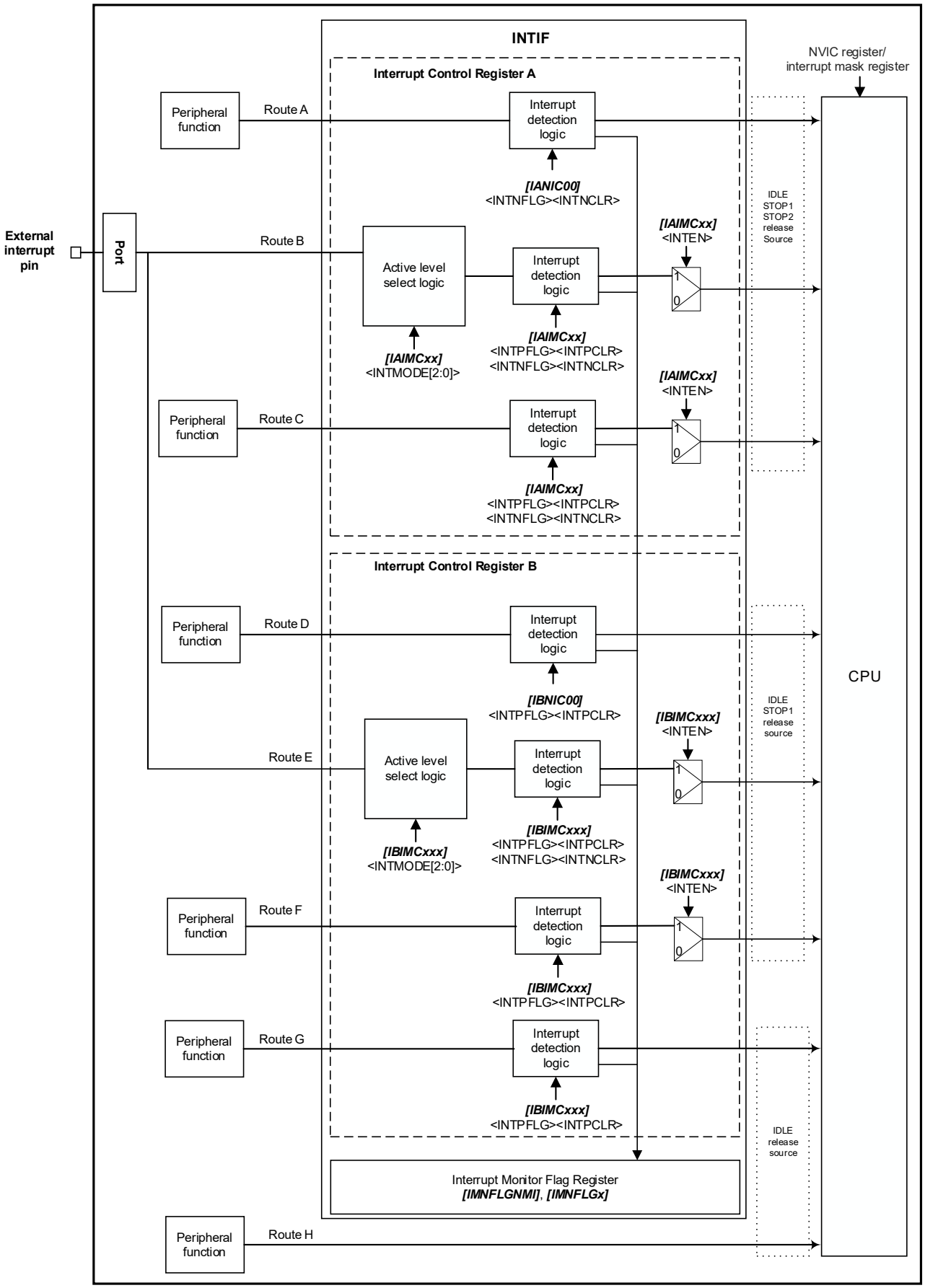


Figure 4.1 Interrupt Transfer Route Diagram

Table 4.1 Explanation of Each Interrupt Transfer Route

Route	Interrupt number	Interrupt request	Route description
A	-	LVD interrupt	This route is NMI. It is a route which is input into CPU via INTIF. The interrupt release setup is set to the interrupt control register A (***[IANIC00]***).
B	0 to 20	External interrupts (00 to 20)	The interrupt request of a port is a route which is input into CPU via INTIF. Enable/Disable of selection of interrupt detection level, interrupt release setup, and interrupt request setup are set to the interrupt control register A (***[IAIMCxx]***) for every source.
C	-	-	This is a route which is input into CPU via INTIF. Enable/Disable of interrupt release setup and interrupt request setup are set to the interrupt control register A (***[IAIMCxx]***) for every source.
D	-	SIWDT interrupt	This route is NMI. It is a route which is input into CPU via INTIF. The interrupt release setup is set to the interrupt control register B (***[IBNIC00]***).
E	-	-	The interrupt request of a port is a route which is input into CPU via INTIF. Enable/Disable of selection of interrupt detection level, interrupt release setup, and an interrupt request setup are set to the interrupt control register B (***[IBIMCxx]***) for every source.
F	-	-	This is a route which input into CPU via INTIF. Enable/Disable of interrupt request setup is set to the interrupt control register B (***[IBIMCxx]***) for every source.
G	21 to 31	DMAC transmission end interrupt (ch0 to 31) T32A ch0 to 4 timer A capture 0 interrupt T32A ch0 to 4 timer A capture 1 interrupt T32A ch0 to 4 timer C capture 0 interrupt T32A ch0 to 4 timer B capture 0 interrupt T32A ch0 to 4 timer B capture 1 interrupt T32A ch0 to 4 timer C every count interrupt	It is a route which is input into CPU via INTIF. The interrupt release setup is set to the interrupt control register B (***[IBIMCxxx]***) for every factor.

4.3.2. Interrupt Request Generation

An interrupt request is generated from an external interrupt pin or peripheral function which is assigned as interrupt request sources, or by setting the relevant bit of NVIC's interrupt set-pending register for interrupt request source.

- Interrupt from external interrupt pin

  Make the pin as the external interrupt pin by setting the port control registers when the external interrupt pin is used.

- Interrupt from peripheral function

  It is required that the peripheral function outputs interrupt requests.
  Refer to the reference manual of each peripheral function for details.

- By setting interrupt set-pending register (forced pending)

  An interrupt request can be forced to be generated by setting the relevant bit of the interrupt set-pending register of NVIC.

CPU will recognize the "High" level of the interrupt request as an interrupt.

4.3.3. Monitor of Interrupt Request

INTIF has the interrupt monitor flags. It can know that the interrupt request has occurred by monitoring the flag. If one request source is representing several interrupt requests, The interrupt monitor register can be used to identify the actual interrupt request source.

For detail, please refer to "4.4 List of Interrupt Sources".

4.3.4. Transmission of Interrupt Request

An interrupt request which is not passing through the Interrupt Control Register will be directly input to the CPU. The interrupts connected to the CPU through INTIF, which are used as Interrupt request sources for releasing the low-power consumption mode, will need proper setting of the Interrupt Control Register in INTIF. A "High" level interrupt signal will be sent to the CPU, when the Interrupt is used to release the low-power consumption mode.

Please setup an Interrupt detection level and Interrupt enable/disable by INTIF.

By the way, please be cautious about external Interrupt pin as in the next section.

4.3.5. Precautions when Using External Interrupt Pins

When an external Interrupt is used, take care about the following points so that an unexpected Interrupt does not occur.

If the ***[PxIE]***<PxmIE> is set to “0”, the input signal from external Interrupt pin is "Low". Then the input signal from the external Interrupt pin is “Low”. When the Interrupt Control Register A ***[IAIMCxx]***<INTMODE> is "Low", the CPU recognizes that level of the external Interrupt pin input “Low” level Interrupt request. Therefore, Interrupt occurs when the corresponding interrupts are enabled by the CPU until the CPU recognizes that the external Interrupt pin is input “Low” level Interrupt request.

Set the Interrupt pin input as "High" level and enable it. Then, enable interrupts by the CPU.

4.4. List of Interrupt Sources

4.4.1. List of Factors and Product Allocation

Table 4.2 shows the list of interrupt sources of non-maskable Interrupts. The setting for clearing the NMI sources can be done by Interrupt Control Registers A and B.

Table 4.2 List of Interrupt Sources (Non-maskable Interrupt)

M4L4	M4L2	M4L1	Interrupt number	interrupt source	Interrupt request	Interrupt control register	Interrupt monitor register
✓	✓	✓	NMI interrupt	INTLVD	LVD interrupt	***[IANIC00]***	***[IMNFLGNMI]*** <INT000FLG>
✓	✓	✓		INTWDT0	SIWDT interrupt	***[IBNIC00]***	***[IMNFLGNMI]*** <INT016FLG>

Table 4.3 shows the list of interrupt sources of interrupt control register A. These interrupt sources can be used for releasing the low-power consumption mode. The interrupt control register A will perform several settings for detecting the release of the low-power consumption mode, and interrupt enable/disable.

Table 4.3 List of Interrupt Sources (Interrupt Control Register A)

M4L4	M4L2	M4L1	Interrupt number	interrupt source	Interrupt request	Interrupt control register	Interrupt monitor register
✓	✓	✓	0	INT00	External interrupt 00	***[IAIMC00]***	***[IMNFLG1]*** <INT032FLG>
✓	✓	✓	1	INT01	External interrupt 01	***[IAIMC01]***	***[IMNFLG1]*** <INT033FLG>
✓	✓	✓	2	INT02	External interrupt 02	***[IAIMC02]***	***[IMNFLG1]*** <INT034FLG>
✓	✓	✓	3	INT03	External interrupt 03	***[IAIMC03]***	***[IMNFLG1]*** <INT035FLG>
✓	✓	✓	4	INT04	External interrupt 04	***[IAIMC04]***	***[IMNFLG1]*** <INT036FLG>
✓	✓	✓	5	INT05	External interrupt 05	***[IAIMC05]***	***[IMNFLG1]*** <INT037FLG>
✓	✓	✓	6	INT06	External interrupt 06	***[IAIMC06]***	***[IMNFLG1]*** <INT038FLG>
✓	✓	✓	7	INT07	External interrupt 07	***[IAIMC07]***	***[IMNFLG1]*** <INT039FLG>
✓	✓	✓	8	INT08	External interrupt 08	***[IAIMC08]***	***[IMNFLG1]*** <INT040FLG>
✓	✓	-	9	INT09	External interrupt 09	***[IAIMC09]***	***[IMNFLG1]*** <INT041FLG>
✓	✓	-	10	INT10	External interrupt 10	***[IAIMC10]***	***[IMNFLG1]*** <INT042FLG>
✓	✓	-	11	INT11	External interrupt 11	***[IAIMC11]***	***[IMNFLG1]*** <INT043FLG>
✓	✓	-	12	INT12	External interrupt 12	***[IAIMC12]***	***[IMNFLG1]*** <INT044FLG>
✓	-	-	13	INT13	External interrupt 13	***[IAIMC13]***	***[IMNFLG1]*** <INT045FLG>
✓	-	-	14	INT14	External interrupt 14	***[IAIMC14]***	***[IMNFLG1]*** <INT046FLG>
✓	-	-	15	INT15	External interrupt 15	***[IAIMC15]***	***[IMNFLG1]*** <INT047FLG>
✓	-	-	16	INT16	External interrupt 16	***[IAIMC16]***	***[IMNFLG1]*** <INT048FLG>
✓	-	-	17	INT17	External interrupt 17	***[IAIMC17]***	***[IMNFLG1]*** <INT049FLG>
✓	-	-	18	INT18	External interrupt 18	***[IAIMC18]***	***[IMNFLG1]*** <INT050FLG>
✓	-	-	19	INT19	External interrupt 19	***[IAIMC19]***	***[IMNFLG1]*** <INT051FLG>
✓	-	-	20	INT20	External interrupt 20	***[IAIMC20]***	***[IMNFLG1]*** <INT052FLG>

The factor list of the interrupt control registers B is shown in Table 4.4. The interrupt control register B will perform Interrupt enable/disable for several Interrupt request.

Table 4.4 List of Interrupt Sources (Interrupt Control Register B)

M4L4	M4L2	M4L1	Interrupt number	interrupt source	Interrupt request	Interrupt control register	Interrupt monitor register
✓	✓	✓	21	INTT32A00ACCAP (Note)	T32A ch0 timer A capture 0 interrupt	***[IBIMC000]***	***[IMNFLG3]*** <INT096FLG>
✓	✓	✓			T32A ch0 timer A capture 1 interrupt	***[IBIMC001]***	***[IMNFLG3]*** <INT097FLG>
✓	✓	✓			T32A ch0 timer C capture 0 interrupt	***[IBIMC002]***	***[IMNFLG3]*** <INT098FLG>
✓	✓	✓	22	INTT32A00BCAP (Note)	T32A ch0 timer B capture 0 interrupt	***[IBIMC003]***	***[IMNFLG3]*** <INT099FLG>
✓	✓	✓			T32A ch0 timer B capture 1 interrupt	***[IBIMC004]***	***[IMNFLG3]*** <INT100FLG>
✓	✓	✓			T32A ch0 timer C Every Count interrupt	***[IBIMC005]***	***[IMNFLG3]*** <INT101FLG>
✓	✓	✓	23	INTT32A01ACCAP (Note)	T32A ch1 timer A capture 0 interrupt	***[IBIMC006]***	***[IMNFLG3]*** <INT102FLG>
✓	✓	✓			T32A ch1 timer A capture 1 interrupt	***[IBIMC007]***	***[IMNFLG3]*** <INT103FLG>
✓	✓	✓			T32A ch1 timer C capture 0 interrupt	***[IBIMC008]***	***[IMNFLG3]*** <INT104FLG>
✓	✓	✓	24	INTT32A01BCAP (Note)	T32A ch1 timer B capture 0 interrupt	***[IBIMC009]***	***[IMNFLG3]*** <INT105FLG>
✓	✓	✓			T32A ch1 timer B capture 1 interrupt	***[IBIMC010]***	***[IMNFLG3]*** <INT106FLG>
✓	✓	✓			T32A ch1 timer C Every Count interrupt	***[IBIMC011]***	***[IMNFLG3]*** <INT107FLG>
✓	✓	✓	25	INTT32A02ACCAP (Note)	T32A ch2 timer A capture 0 interrupt	***[IBIMC012]***	***[IMNFLG3]*** <INT108FLG>
✓	✓	✓			T32A ch2 timer A capture 1 interrupt	***[IBIMC013]***	***[IMNFLG3]*** ***<INT109FLG>***
✓	✓	✓			T32A ch2 timer C capture 0 interrupt	***[IBIMC014]***	***[IMNFLG3]*** ***<INT110FLG>***
✓	✓	✓	26	INTT32A02BCAP (Note)	T32A ch2 timer B capture 0 interrupt	***[IBIMC015]***	***[IMNFLG3]*** ***<INT111FLG>***
✓	✓	✓			T32A ch2 timer B capture 1 interrupt	***[IBIMC016]***	***[IMNFLG3]*** ***<INT112FLG>***
✓	✓	✓			T32A ch2 timer C Every Count interrupt	***[IBIMC017]***	***[IMNFLG3]*** ***<INT113FLG>***
✓	✓	✓	27	INTT32A03ACCAP (Note)	T32A ch3 timer A capture 0 interrupt	***[IBIMC018]***	***[IMNFLG3]*** <INT114FLG>
✓	✓	✓			T32A ch3 timer A capture 1 interrupt	***[IBIMC019]***	***[IMNFLG3]*** <INT115FLG>
✓	✓	✓			T32A ch3 timer C Every Count interrupt	***[IBIMC020]***	***[IMNFLG3]*** <INT116FLG>
✓	✓	✓	28	INTT32A03BCAP (Note)	T32A ch3 timer B capture 0 interrupt	***[IBIMC021]***	***[IMNFLG3]*** <INT117FLG>
✓	✓	✓			T32A ch3 timer B capture 1iinterrupt	***[IBIMC022]***	***[IMNFLG3]*** <INT118FLG>
✓	✓	✓			T32A ch3 timer C Every Count interrupt	***[IBIMC023]***	***[IMNFLG3]*** <INT119FLG>
✓	✓	✓	29	INTT32A04ACCAP (Note)	T32A ch4 timer A capture 0 interrupt	***[IBIMC024]***	***[IMNFLG3]*** <INT120FLG>
✓	✓	✓			T32A ch4 timer A capture 1 interrupt	***[IBIMC025]***	***[IMNFLG3]*** <INT121FLG>
✓	✓	✓			T32A ch4 timer C Every Count interrupt	***[IBIMC026]***	***[IMNFLG3]*** <INT122FLG>

M4L4	M4L2	M4L1	Interrupt number	interrupt source	Interrupt request	Interrupt control register	Interrupt monitor register
✓	✓	✓	30	INTT32A04BCAP (Note)	T32A ch4 timer B capture 0 interrupt	***[IBIMC027]***	***[IMNFLG3]*** <INT123FLG>
✓	✓	✓			T32A ch4 timer B capture 1 interrupt	***[IBIMC028]***	***[IMNFLG3]*** <INT124FLG>
✓	✓	✓			T32A ch4 timer C Every Count interrupt	***[IBIMC029]***	***[IMNFLG3]*** <INT125FLG>
✓	✓	✓	31	INTDMAATC (Note)	DMAC unit A transfer end interrupt (ch0 to 31)	***[IBIMC030]*** to ***[IBIMC061]***	***[IMNFLG3]*** <INT126FLG> to <INT157FLG>
✓	✓	✓	32	INTVCN0	A-VE2 ch0 schedule end interrupt		
-	-	-	33	Reserved			
✓	✓	✓	34	INTEMG0	A-PMD2 ch0 EMG interrupt		
✓	✓	✓	35	INTOVV0	A-PMD2 ch0 OVV interrupt		
✓	✓	✓	36	INTPWM0	A-PMD2 ch0 PWM interrupt		
✓	✓	✓	37	INTENC00	A-ENC2 ch0 encoder input interrupt 0		
✓	✓	✓	38	INTENC01	A-ENC2 ch0 encoder input interrupt 1		
✓	✓	✓	39	INTCOMP0	COMP ch0 interrupt		
✓	✓	✓	40	INTCOMP1	COMP ch1 interrupt		
✓	✓	✓	41	INTCOMP2	COMP ch2 interrupt		
✓	✓	✓	42	INTCOMP3	COMP ch3 interrupt		
✓	✓	✓	43	INTCOMP4	COMP ch4 interrupt		
✓	✓	✓	44	INTCOMP5	COMP ch5 interrupt		
✓	✓	✓	45	INTADAPDA	ADC unit A PMD trigger interrupt A		
✓	✓	✓	46	INTADAPDB	ADC unit A PMD trigger interrupt B		
✓	✓	✓	47	INTADACP0	ADC unit A monitoring function 0 interrupt		
✓	✓	✓	48	INTADACP1	ADC unit A monitoring function 1 interrupt		
✓	✓	✓	49	INTADATRG	ADC unit A general purpose trigger interrupt		
✓	✓	✓	50	INTADASGL	ADC unit A single conversion interrupt		
✓	✓	✓	51	INTADACNT	ADC unit A continuous conversion interrupt		
✓	✓	✓	52	INTADBPDA	ADC unit B PMD trigger interrupt A		
✓	✓	✓	53	INTADBPDB	ADC unit B PMD trigger interrupt B		
✓	✓	✓	54	INTADBCP0	ADC unit B monitoring function 0 interrupt		
✓	✓	✓	55	INTADBCP1	ADC unit B monitoring function 1 interrupt		
✓	✓	✓	56	INTADBTRG	ADC unit B General purpose trigger interrupt		
✓	✓	✓	57	INTADBSGL	ADC unit B single conversion interrupt		
✓	✓	✓	58	INTADBCNT	ADC unit B continuous conversion interrupt		
✓	-	-	59	INTT0RX	TSPI ch0 reception interrupt		
✓	-	-	60	INTT0TX	TSPI ch0 transmission interrupt		
✓	-	-	61	INTT0ERR	TSPI ch0 error interrupt		
✓	✓	✓	62	INTT1RX	TSPI ch1 reception interrupt		
✓	✓	✓	63	INTT1TX	TSPI ch1 transmission interrupt		
✓	✓	✓	64	INTT1ERR	TSPI ch1 error interrupt		
✓	✓	✓	65	INTT2RX	TSPI ch2 reception interrupt		
✓	✓	✓	66	INTT2TX	TSPI ch2 transmission interrupt		
✓	✓	✓	67	INTT2ERR	TSPI ch2 error interrupt		
✓	✓	✓	68	INTUART0RX	UART ch0 reception interrupt		
✓	✓	✓	69	INTUART0TX	UART ch0 transmission interrupt		
✓	✓	✓	70	INTUART0ERR	UART ch0 error interrupt		

M4L4	M4L2	M4L1	Interrupt number	interrupt source	Interrupt request	Interrupt control register	Interrupt monitor register
✓	✓	✓	71	INTUART1RX	UART ch1 reception interrupt		
✓	✓	✓	72	INTUART1TX	UART ch1 transmission interrupt		
✓	✓	✓	73	INTUART1ERR	UART ch1 error interrupt		
✓	✓	✓	74	INTUART2RX	UART ch2 reception interrupt		
✓	✓	✓	75	INTUART2TX	UART ch2 transmission interrupt		
✓	✓	✓	76	INTUART2ERR	UART ch2 error interrupt		
✓	✓	✓	77	INTUART3RX	UART ch3 reception interrupt		
✓	✓	✓	78	INTUART3TX	UART ch3 transmission interrupt		
✓	✓	✓	79	INTUART3ERR	UART ch3 error interrupt		
✓	-	-	80	INTUART4RX	UART ch4 reception interrupt		
✓	-	-	81	INTUART4TX	UART ch4 transmission interrupt		
✓	-	-	82	INTUART4ERR	UART ch4 error interrupt		
✓	✓	-	83	INTUART5RX	UART ch5 reception interrupt		
✓	✓	-	84	INTUART5TX	UART ch5 transmission interrupt		
✓	✓	-	85	INTUART5ERR	UART ch5 error interrupt		
✓	✓	✓	86	INTSMI0	SMIF ch0 interrupt		
✓	✓	✓	87	INTI2C0BF	EI2C ch0 receive buffer full interrupt		
✓	✓	✓	88	INTI2C0AL	EI2C ch0 transmit buffer empty interrupt		
✓	✓	✓	89	INTI2C0	EI2C ch0 status interrupt		
✓	✓	-	90	INTI2C1BF	EI2C ch1 receive buffer full interrupt		
✓	✓	-	91	INTI2C1AL	EI2C ch1 transmit buffer empty interrupt		
✓	✓	-	92	INTI2C1	EI2C ch1 status interrupt		
✓	✓	✓	93	INTT32A00AC (Note)	T32A ch0 timer A match, overflow, and Underflow interrupt		
✓	✓	✓			T32A ch0 timer C match, overflow, and underflow interrupt		
✓	✓	✓	94	INTT32A00B (Note)	T32A ch0 timer B match, overflow, and underflow interrupt		
✓	✓	✓			T32A ch0 timer C capture 1 interrupt		
✓	✓	✓	95	INTT32A01AC (Note)	T32A ch1 timer A match, overflow, and underflow interrupt		
✓	✓	✓			T32A ch1 timer C match, overflow, and underflow interrupt		
✓	✓	✓	96	INTT32A01B (Note)	T32A ch1 timer B match, overflow, and underflow interrupt		
✓	✓	✓			T32A ch1 timer C capture 1 interrupt		
✓	✓	✓	97	INTT32A02AC (Note)	T32A ch2 timer A match, overflow, and underflow interrupt		
✓	✓	✓			T32A ch2 timer C match, overflow, and underflow interrupt		
✓	✓	✓	98	INTT32A02B (Note)	T32A ch2 timer B match, overflow, and underflow		
✓	✓	✓			T32A ch2 timer C capture 1 interrupt		
✓	✓	✓	99	INTT32A03AC (Note)	T32A ch3 timer A match, overflow, and underflow interrupt		
✓	✓	✓			T32A ch3 timer C match, overflow, and underflow interrupt		
✓	✓	✓	100	INTT32A03B (Note)	T32A ch3 timer B match, overflow, and underflow		
✓	✓	✓			T32A ch3 timer C capture 1 interrupt		

M4L4	M4L2	M4L1	Interrupt number	interrupt source	Interrupt request	Interrupt control register	Interrupt monitor register
✓	✓	✓	101	INTT32A04AC (Note)	T32A ch4 timer A match, overflow, and underflow interrupt		
✓	✓	✓			T32A ch4 timer C match, overflow, and underflow interrupt		
✓	✓	✓	102	INTT32A04B (Note)	T32A ch4 timer B match, overflow, and underflow		
✓	✓	✓			T32A ch4 timer C capture 1 interrupt		
✓	✓	✓	103	INTPARI	T32A ch4 timer AC capture 1 interrupt		
✓	✓	✓	104	INTDMAAERR	DMAC unit A transmission error interrupt		
✓	✓	✓	105	INTFLCRDY	Flash ready interrupt		

✓: Supported, -: Not Supported

Note: Please refer to "4.4.2 About Joint Interrupt".

4.4.2. About Joint Interrupt

The details of interrupts which are jointed in TMPM4L Group (1) are as follows.

Table 4.5 List of Joint Interrupt (1/3)

Interrupt number	interrupt source	Interrupt request	Interrupt control register	Interrupt monitor register
21	INTT32A00ACCAP	INTT32A00ACAP0 (T32A ch0 timer A capture0 interrupt)	***[IBIMC000]***	***[IMNFLG3]*** <INT096FLG>
		INTT32A00ACAP1 (T32A ch0 timer A capture1 interrupt)	***[IBIMC001]***	***[IMNFLG3]*** <INT097FLG>
		INTT32A00CCAP0 (T32A ch0 timer C capture0 interrupt)	***[IBIMC002]***	***[IMNFLG3]*** <INT098FLG>
22	INTT32A00BCAP	INTT32A00BCAP0 (T32A ch0 timer B capture0 interrupt)	***[IBIMC003]***	***[IMNFLG3]*** <INT099FLG>
		INTT32A00BCAP1 (T32A ch0 timer B capture1 interrupt)	***[IBIMC004]***	***[IMNFLG3]*** <INT100FLG>
		INTT32A00EVRYC (T32A ch0 timer C every count interrupt)	***[IBIMC005]***	***[IMNFLG3]*** <INT101FLG>
23	INTT32A01ACCAP	INTT32A01ACAP0 (T32A ch1 timer A capture0 interrupt)	***[IBIMC006]***	***[IMNFLG3]*** <INT102FLG>
		INTT32A01ACAP1 (T32A ch1 timer A capture1 interrupt)	***[IBIMC007]***	***[IMNFLG3]*** <INT103FLG>
		INTT32A01CCAP0 (T32A ch1 timer C capture0 interrupt)	***[IBIMC008]***	***[IMNFLG3]*** <INT104FLG>
24	INTT32A01BCAP	INTT32A01BCAP0 (T32A ch1 timer B capture0 interrupt)	***[IBIMC009]***	***[IMNFLG3]*** <INT105FLG>
		INTT32A01BCAP1 (T32A ch1 timer B capture1 interrupt)	***[IBIMC010]***	***[IMNFLG3]*** <INT106FLG>
		INTT32A01EVRYC (T32A ch1 timer C every count interrupt)	***[IBIMC011]***	***[IMNFLG3]*** <INT107FLG>
25	INTT32A02ACCAP	INTT32A02ACAP0 (T32A ch2 timer A capture0 interrupt)	***[IBIMC012]***	***[IMNFLG3]*** <INT108FLG>
		INTT32A02ACAP1 (T32A ch2 timer A capture1 interrupt)	***[IBIMC013]***	***[IMNFLG3]*** <INT109FLG>
		INTT32A02CCAP0 (T32A ch2 timer C capture0 interrupt)	***[IBIMC014]***	***[IMNFLG3]*** <INT110FLG>
26	INTT32A02BCAP	INTT32A02BCAP0 (T32A ch2 timer B capture0 interrupt)	***[IBIMC015]***	***[IMNFLG3]*** <INT111FLG>
		INTT32A02BCAP1 (T32A ch2 timer B capture1 interrupt)	***[IBIMC016]***	***[IMNFLG3]*** <INT112FLG>
		INTT32A02EVRYC (T32A ch2 timer C every count interrupt)	***[IBIMC017]***	***[IMNFLG3]*** <INT113FLG>
27	INTT32A03ACCAP	INTT32A03ACAP0 (T32A ch3 timer A capture0 interrupt)	***[IBIMC018]***	***[IMNFLG3]*** <INT114FLG>
		INTT32A03ACAP1 (T32A ch3 timer A capture1 interrupt)	***[IBIMC019]***	***[IMNFLG3]*** <INT115FLG>
		INTT32A03CCAP0 (T32A ch3 timer C capture0 interrupt)	***[IBIMC020]***	***[IMNFLG3]*** <INT116FLG>

Interrupt number	interrupt source	Interrupt request	Interrupt control register	Interrupt monitor register
28	INTT32A03BCAP	INTT32A03BCAP0 (T32A ch3 timer B capture0 interrupt)	***[IBIMC021]***	***[IMNFLG3]*** <INT117FLG>
		INTT32A03BCAP1 (T32A ch3 timer B capture1 interrupt)	***[IBIMC022]***	***[IMNFLG3]*** <INT118FLG>
		INTT32A03EVRYC (T32A ch3 timer C every count interrupt)	***[IBIMC023]***	***[IMNFLG3]*** <INT119FLG>
29	INTT32A04ACCAP	INTT32A04ACAP0 (T32A ch4 timer A capture0 interrupt)	***[IBIMC024]***	***[IMNFLG3]*** <INT120FLG>
		INTT32A04ACAP1 (T32A ch4 timer A capture1 interrupt)	***[IBIMC025]***	***[IMNFLG3]*** <INT121FLG>
		INTT32A04CCAP0 (T32A ch4 timer C capture0 interrupt)	***[IBIMC026]***	***[IMNFLG3]*** <INT122FLG>
30	INTT32A04BCAP	INTT32A04BCAP0 (T32A ch4 timer B capture0 interrupt)	***[IBIMC027]***	***[IMNFLG3]*** <INT123FLG>
		INTT32A04BCAP1 (T32A ch4 timer B capture1 interrupt)	***[IBIMC028]***	***[IMNFLG3]*** <INT124FLG>
		INTT32A04EVRYC (T32A ch4 timer C every count interrupt)	***[IBIMC029]***	***[IMNFLG3]*** <INT125FLG>

Table 4.6 List of Joint Interrupt (2/3)

Interrupt number	interrupt source	Interrupt request	Interrupt control register	Interrupt monitor register
93	INTT32A00AC	INTT32A00A (T32A ch0 timer A match, overflow, and underflow interrupt)		
		INTT32A00C (T32A ch0 timer C match, overflow, and underflow interrupt)		
94	INTT32A00B	INTT32A00B (T32A ch0 timer B match, overflow, and underflow interrupt)		
		INTT32A00CCAP1 (T32A ch0 timer C capture 1 interrupt)		
95	INTT32A01AC	INTT32A01A (T32A ch1 timer A match, overflow, and underflow interrupt)		
		INTT32A01C (T32A ch1 timer C match, overflow, and underflow interrupt)		
96	INTT32A01B	INTT32A01B (T32A ch1 timer B match, overflow, and underflow interrupt)		
		INTT32A01CCAP1 (T32A ch1 timer C capture 1 interrupt)		
97	INTT32A02AC	INTT32A02A (T32A ch2 timer A match, overflow, and underflow interrupt)		
		INTT32A02C (T32A ch2 timer C match, overflow, and underflow interrupt)		
98	INTT32A02B	INTT32A02B (T32A ch2 timer B match, overflow, and underflow interrupt)		
		INTT32A02CCAP1 (T32A ch2 timer C capture 1 interrupt)		
99	INTT32A03AC	INTT32A03A (T32A ch0 timer A match, overflow, and underflow interrupt)		
		INTT32A03C (T32A ch0 timer C match, overflow, and underflow interrupt)		
100	INTT32A03B	INTT32A03B (T32A ch0 timer B match, overflow, and underflow interrupt)		
		INTT32A03CCAP1 (T32A ch0 timer C capture 1 interrupt)		
101	INTT32A04AC	INTT32A04A (T32A ch4 timer A match, overflow, and underflow interrupt)		
		INTT32A04C (T32A ch4 timer C match, overflow, and underflow interrupt)		
102	INTT32A04B	INTT32A04B (T32A ch4 timer B match, overflow, and underflow interrupt)		
		INTT32A04CCAP1 (T32A ch4 timer C capture 1 interrupt)		

Table 4.7 List of Joint Interrupt (3/3)

Interrupt number	interrupt source	Interrupt request	Interrupt control register	Interrupt monitor register
31	INTDMAATC	DMAC unit A transfer end interrupt ch0	***[IBIMC030]***	***[IMNFLG3]*** <INT126FLG>
		DMAC unit A transfer end interrupt ch1	***[IBIMC031]***	***[IMNFLG3]*** <INT127FLG>
		DMAC unit A transfer end interrupt ch2	***[IBIMC032]***	***[IMNFLG4]*** <INT128FLG>
		DMAC unit A transfer end interrupt ch3	***[IBIMC033]***	***[IMNFLG4]*** <INT129FLG>
		DMAC unit A transfer end interrupt ch4	***[IBIMC034]***	***[IMNFLG4]*** <INT130FLG>
		DMAC unit A transfer end interrupt ch5	***[IBIMC035]***	***[IMNFLG4]*** <INT131FLG>
		DMAC unit A transfer end interrupt ch6	***[IBIMC036]***	***[IMNFLG4]*** <INT132FLG>
		DMAC unit A transfer end interrupt ch7	***[IBIMC037]***	***[IMNFLG4]*** <INT133FLG>
		DMAC unit A transfer end interrupt ch8	***[IBIMC038]***	***[IMNFLG4]*** <INT134FLG>
		DMAC unit A transfer end interrupt ch9	***[IBIMC039]***	***[IMNFLG4]*** <INT135FLG>
		DMAC unit A transfer end interrupt ch10	***[IBIMC040]***	***[IMNFLG4]*** <INT136FLG>
		DMAC unit A transfer end interrupt ch11	***[IBIMC041]***	***[IMNFLG4]*** <INT137FLG>
		DMAC unit A transfer end interrupt ch12	***[IBIMC042]***	***[IMNFLG4]*** <INT138FLG>
		DMAC unit A transfer end interrupt ch13	***[IBIMC043]***	***[IMNFLG4]*** <INT139FLG>
		DMAC unit A transfer end interrupt ch14	***[IBIMC044]***	***[IMNFLG4]*** <INT140FLG>
		DMAC unit A transfer end interrupt ch15	***[IBIMC045]***	***[IMNFLG4]*** <INT141FLG>
		DMAC unit A transfer end interrupt ch16	***[IBIMC046]***	***[IMNFLG4]*** <INT142FLG>
		DMAC unit A transfer end interrupt ch17	***[IBIMC047]***	***[IMNFLG4]*** <INT143FLG>
		DMAC unit A transfer end interrupt ch18	***[IBIMC048]***	***[IMNFLG4]*** <INT144FLG>
		DMAC unit A transfer end interrupt ch19	***[IBIMC049]***	***[IMNFLG4]*** <INT145FLG>
		DMAC unit A transfer end interrupt ch20	***[IBIMC050]***	***[IMNFLG4]*** <INT146FLG>
		DMAC unit A transfer end interrupt ch21	***[IBIMC051]***	***[IMNFLG4]*** <INT147FLG>
		DMAC unit A transfer end interrupt ch22	***[IBIMC052]***	***[IMNFLG4]*** <INT148FLG>
		DMAC unit A transfer end interrupt ch23	***[IBIMC053]***	***[IMNFLG4]*** <INT149FLG>
		DMAC unit A transfer end interrupt ch24	***[IBIMC054]***	***[IMNFLG4]*** <INT150FLG>
		DMAC unit A transfer end interrupt ch25	***[IBIMC055]***	***[IMNFLG4]*** <INT151FLG>
		DMAC unit A transfer end interrupt ch26	***[IBIMC056]***	***[IMNFLG4]*** <INT152FLG>
		DMAC unit A transfer end interrupt ch27	***[IBIMC057]***	***[IMNFLG4]*** <INT153FLG>
		DMAC unit A transfer end interrupt ch28	***[IBIMC058]***	***[IMNFLG4]*** <INT154FLG>

Interrupt number	interrupt source	Interrupt request	Interrupt control register	Interrupt monitor register
		DMAC unit A transfer end interrupt ch29	***[IBIMC059]***	***[IMNFLG4]*** <INT155FLG>
		DMAC unit A transfer end interrupt ch30	***[IBIMC060]***	***[IMNFLG4]*** <INT156FLG>
		DMAC unit A transfer end interrupt ch31	***[IBIMC061]***	***[IMNFLG4]*** <INT157FLG>

4.5. Interrupt Detection Level

When using interrupt via INTIF, interrupt detection level ("Low" level, "High" level, Rising edge, or Falling edge) can be selected by interrupt control register A or B. The detected interrupt is output to the CPU with a "High" level signal.

The interrupt signals which are directly input from the various peripheral functions to the CPU, a "High" pulse is output to the CPU as an interrupt request.

The CPU recognizes the “High” level of interrupt signal as an interrupt request.

4.5.1. Precautions when Releasing Low-power Consumption Mode

Following settings should be done when releasing STOP1/2 mode.

- The setup of the interrupt control register. (***[IAIMCxx]***, ***[IBIMCxxx]***)
  - Interrupt detection level
  - Interrupt detection enable/disable

- The setup of the NVIC interrupt enable set register (STOP1 only)
  - enable/disable setup

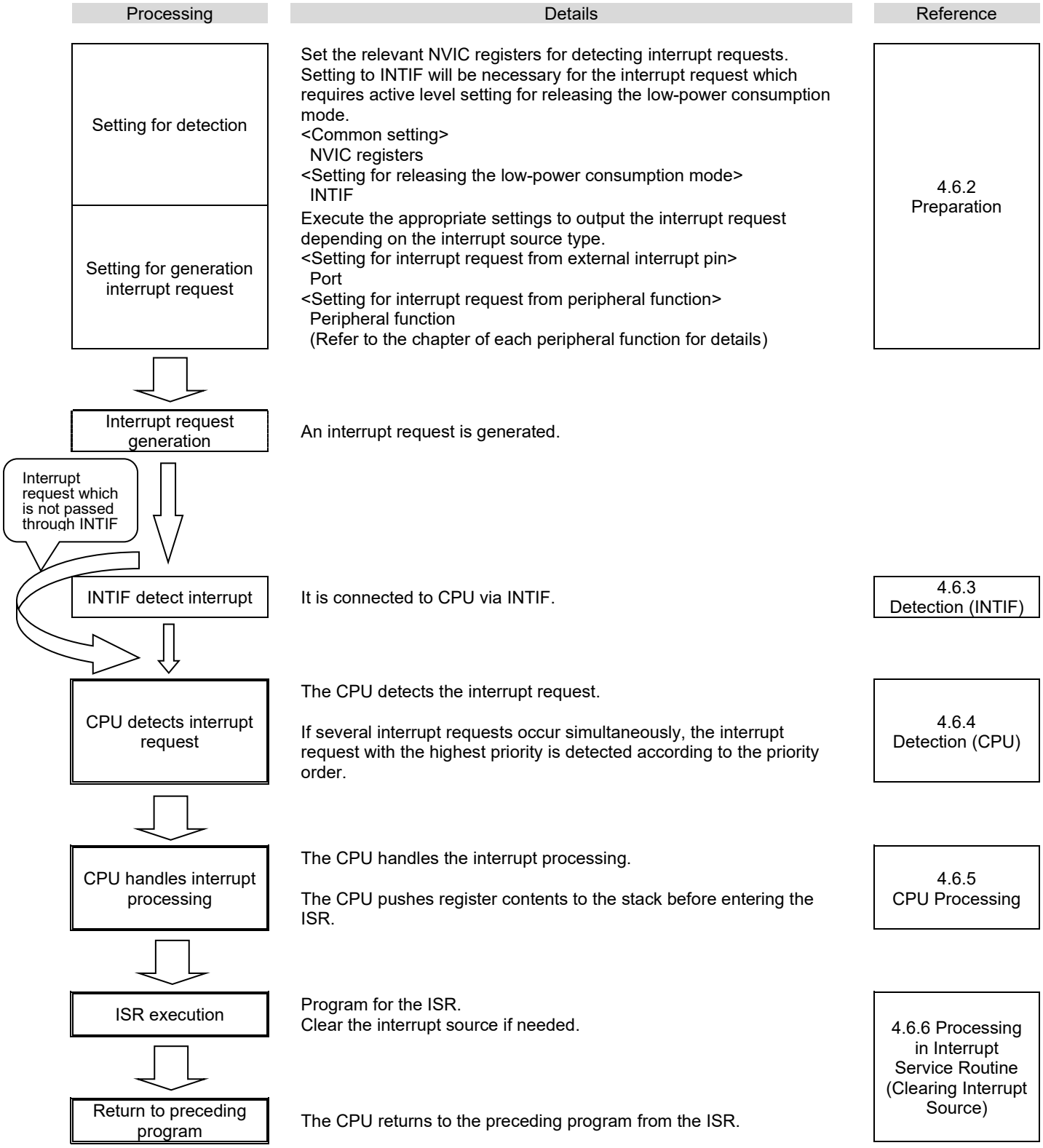
The operation to return from STOP1 mode to NORMAL mode is as follows.

By jumping to the interrupt service routine after the high-speed clock oscillation resumes, operation resumes from the instruction following the instruction to enter STOP1 mode. In addition, when returning from STOP2 mode to NORMAL mode, power supply is applied to the main power domain and the operation is restarted from the reset sequence.

4.6. Interrupt Handling

4.6.1. Flowchart of Handling Interrupt Processing

The following shows the flowchart of handling interrupt process.
The flowchart below explains the handling process by hardware and software.



4.6.2. Preparation

When preparing for an interrupt , need to pay attention to the order of configuration to avoid any unexpected interrupt on the way.

Starting an interrupt or changing its configuration must be implemented in the following order basically. First, disable the interrupt detection of the CPU. Then, configure from the farthest route from the CPU. Finally, enable the interrupt detection of the CPU.

To configure the INTIF, the configuration of INTIF must be implemented in the following order indicated here. First, configure the precondition. Secondly, clear the status related to the interrupt in the INTIF not to cause any unexpected interrupt. Finally, enable the interrupt detection of the INTIF.

The following sections are listed in the order of interrupt handling and describe how to configure them.

(d) Disabling interrupt acceptance of CPU
(e) CPU interrupt settings
(f) Preparation (1) (Interrupt from external interrupt pins)
(g) Preparation (2) (Interrupt from peripheral functions)
(h) Preparation (3) (Interrupt by set-pending register)
(i) INTIF settings
(j) Enabling interrupt acceptance of CPU

(a) Disabling interrupt acceptance of CPU

To make the CPU for not accepting any interrupt, write "1" to ***[PRIMASK]***. All interrupts and exceptions other than non-maskable interrupts and hard faults can be masked.

Use "MSR" instruction to set this register.

Interrupt Mask Register		
[PRIMASK]	←	"1" (Interrupt disabled)

Note 1: ***[PRIMASK]*** register cannot be modified in the user access level

Note 2: If a fault causes when "1" is set to the ***[PRIMASK]***, it is handled as a hard fault.

(b) CPU interrupt settings

A priority level is set by writing to <PRI_n> of an interrupt priority register in the NVIC.
Each interrupt source is provided with eight bits for assigning a priority level from 0 to 255, but the number of bits actually used varies with each product. Priority level 0 is the highest priority level. If several sources have the same priority, the smallest-numbered interrupt source has the highest priority.
A grouping priority by writing to the <PRIGROUP> of the application interrupt and reset control register.

NVIC Register		
<PRI_n>	←	"Priority"
<PRIGROUP>	←	"group priority" (This is configurable if required)

Note: "n" indicates the corresponding exception number and interrupt number.
This product uses four bits for assigning a priority level.

(c) Preparation (1) (Interrupt from external interrupt pins)

In order to use external interrupt pin, it is necessary to do proper setting to the port function register of the corresponding pin. Setting "1" to ***[PxIE]***<PxmIE> allows the pin to be used as the function pin and the input port.

Port Register		
[PxIE]<PxmIE>	←	"1"

Note: x: port number, m: corresponding bit
Be careful not to enable interrupts that are not used when performing interruptsetting. Also be aware of the description of "4.3.5 Precautions when Using External Interrupt Pins"

(d) Preparation (2) (Interrupt from peripheral functions)

The setting varies depending on the peripheral functions to be used. Refer to the reference manual of each peripheral function for details.

(e) Preparation (3) (Interrupt by set-pending register)

To generate an interrupt by using the interrupt set-pending register, set "1" to the corresponding bit of this register.

NVIC Register		
<SETPEND>	←	"1"

Note: <SETPEND>: corresponding bit

(f) INTIF Setting

Enabling the interrupt via INTIF is sets by interrupt control registers.

[IANIC00], ***[IBNIC00]***, ***[IAIMCxx]***, and ***[IBIMCxxx]*** registers are used for configuring each interrupt source. Before enabling an interrupt , clear the interrupt request in order to avoid unexpected interrupt in the interrupt detection circuit.
For details of the interrupt control register, refer to the followings.

Interrupt Control Register		
[IAIMCxx]<INTMODE> ***[IBIMCxxx]***<INTMODE>	←	Value corresponding to the interrupt to be used (Only for the interrupt having interrupt detection level)
[IANIC00]<INTNCLR> ***[IBNIC00]***<INTPCLR> ***[IAIMCxx]***<INTPCLR><INTNCLR> ***[IBIMCxxx]***<INTPCLR><INTNCLR>	←	Interrupt request clear to use
[IAIMCxx]<INTEN> ***[IBIMCxxx]***<INTEN>	←	"1" (Interrupt detection enabled)

Note: xx and xxx: number specific to the interrupt request

(g) Enabling interrupt acceptance of CPU

Enabling interrupt acceptance of CPU is shown below.
Clear the pending interrupt by the interrupt clear-pending register. Enable the interrupt acceptance with the interrupt set-enable register. Each bit of the register is assigned to a single interrupt request.
Writing "1" to the corresponding bit of the interrupt clear-pending register clears the pending interrupt request. Writing "1" to the corresponding bit of the interrupt set-enable register enables the interrupt acceptance.
To generate interrupt requests by the interrupt set-pending register setting, requests to trigger interrupts are lost if pending interrupts are cleared. Thus, this operation is not necessary.
At the end, ***[PRIMASK]*** register is cleared to “0”.

NVIC Register		
<CLRPEND>	←	"1"
<SETENA>	←	"1"
Interrupt Mask Register		
[PRIMASK]	←	"0"

Note 1: <CLRPEND>,<SETENA>: corresponding bit

Note 2: ***[PRIMASK]*** register cannot be modified by the user access level.

4.6.3. Detection (INTIF)

When an interrupt request is detected by the INTIF, it sends as the interrupt signal of "High" level from the INTF to the CPU.

The INTIF has the functions of the interrupt detection level selection logic, the functions of detection logic, and the function of the interrupt enable/disable. Each function of INTIF is set by the interrupt control register A or B.

It keeps sending the interrupt signal of High" level to the CPU until the <detection flag> is cleared by the interrupt control register after the INTF detects an interrupt. If the ISR is exited without clearing the interrupt request, the same interrupt will be detected again. Thus, be sure to clear <detection flag> of each interrupt request in the ISR. At the same time, the corresponding interrupt monitor flag register is also cleared.

4.6.4. Detection (CPU)

The CPU detects an interrupt request with the highest priority in accordance with priority.

4.6.5. CPU Processing

On detecting an interrupt request, the CPU pushes the contents of xPSR, PC, LR, r12, and r3 to r0 to the stack then enter the ISR.

4.6.6. Processing in Interrupt Service Routine (Clearing Interrupt Source)

An ISR requires specific programming according to the application to be used. This section describes what is recommended in the interrupt service routine and how the source is cleared.

(a) Process in interrupt service routine
An ISR normally pushes register contents to the stack and handles an interrupt as required.
The Cortex-M4 processor with FPU automatically pushes the contents of xPSR, PC, LR, r12, and r3to r0 to the stack. No extra programming is required for them.
Push the contents of other registers if needed.
Interrupt requests with higher priority and exceptions such as NMI are accepted even when an ISR is being executed. It is recommended that the contents of general-purpose registers that might be rewritten are pushed.

(b) Clearing interrupt source
Some interrupt requests have to be cleared by the interrupt control register.
If an interrupt detection level is set as level-sensitive, an interrupt request continues to exist unless the source itself is cleared. Therefore, the interrupt source must be cleared. If an interrupt source is cleared in level detection, the interrupt request signal from INTIF will be cleared automatically.
An interrupt source is cleared by clearing the <Interrupt flag> of the interrupt control register of INTIF in the case of edge detection. When an effective edge occurs again, it is anew recognized as an interrupt source.

Note: After clearing <Interrupt flag> of the interrupt control register, please be sure to read the flag which was cleared.

5. Registers

5.1. Register List

Control Registers and their addresses are as follows.

5.1.1. Interrupt Control Registers A

Peripheral function		Channel/unit	Base address
Interrupt control register A	IA	-	0x4003E000

Register name		Address (+BASE)
Non-maskable Interrupt A Control Register 00	***[IANIC00]***	0x0000
Interrupt A Mode Control Register 00	***[IAIMC00]***	0x0020
Interrupt A Mode Control Register 01	***[IAIMC01]***	0x0021
Interrupt A Mode Control Register 02	***[IAIMC02]***	0x0022
Interrupt A Mode Control Register 03	***[IAIMC03]***	0x0023
Interrupt A Mode Control Register 04	***[IAIMC04]***	0x0024
Interrupt A Mode Control Register 05	***[IAIMC05]***	0x0025
Interrupt A Mode Control Register 06	***[IAIMC06]***	0x0026
Interrupt A Mode Control Register 07	***[IAIMC07]***	0x0027
Interrupt A Mode Control Register 08	***[IAIMC08]***	0x0028
Interrupt A Mode Control Register 09	***[IAIMC09]***	0x0029
Interrupt A Mode Control Register 10	***[IAIMC10]***	0x002A
Interrupt A Mode Control Register 11	***[IAIMC11]***	0x002B
Interrupt A Mode Control Register 12	***[IAIMC12]***	0x002C
Interrupt A Mode Control Register 13	***[IAIMC13]***	0x002D
Interrupt A Mode Control Register 14	***[IAIMC14]***	0x002E
Interrupt A Mode Control Register 15	***[IAIMC15]***	0x002F
Interrupt A Mode Control Register 16	***[IAIMC16]***	0x0030
Interrupt A Mode Control Register 17	***[IAIMC17]***	0x0031
Interrupt A Mode Control Register 18	***[IAIMC18]***	0x0032
Interrupt A Mode Control Register 19	***[IAIMC19]***	0x0033
Interrupt A Mode Control Register 20	***[IAIMC20]***	0x0034

Note: Byte access is required when accessing to interrupt control register A.

5.1.2. Interrupt Control Registers B

Peripheral function		Channel/unit	Base address
Interrupt control register B	IB	-	0x400F4E00

Register name		Address (+BASE)
Non-maskable Interrupt B Control Register 00	***[IBNIC00]***	0x0010
Interrupt B Mode Control Register 000	***[IBIMC000]***	0x0060
Interrupt B Mode Control Register 001	***[IBIMC001]***	0x0061
Interrupt B Mode Control Register 002	***[IBIMC002]***	0x0062
Interrupt B Mode Control Register 003	***[IBIMC003]***	0x0063
Interrupt B Mode Control Register 004	***[IBIMC004]***	0x0064
Interrupt B Mode Control Register 005	***[IBIMC005]***	0x0065
Interrupt B Mode Control Register 006	***[IBIMC006]***	0x0066
Interrupt B Mode Control Register 007	***[IBIMC007]***	0x0067
Interrupt B Mode Control Register 008	***[IBIMC008]***	0x0068
Interrupt B Mode Control Register 009	***[IBIMC009]***	0x0069
Interrupt B Mode Control Register 010	***[IBIMC010]***	0x006A
Interrupt B Mode Control Register 011	***[IBIMC011]***	0x006B
Interrupt B Mode Control Register 012	***[IBIMC012]***	0x006C
Interrupt B Mode Control Register 013	***[IBIMC013]***	0x006D
Interrupt B Mode Control Register 014	***[IBIMC014]***	0x006E
Interrupt B Mode Control Register 015	***[IBIMC015]***	0x006F
Interrupt B Mode Control Register 016	***[IBIMC016]***	0x0070
Interrupt B Mode Control Register 017	***[IBIMC017]***	0x0071
Interrupt B Mode Control Register 018	***[IBIMC018]***	0x0072
Interrupt B Mode Control Register 019	***[IBIMC019]***	0x0073
Interrupt B Mode Control Register 020	***[IBIMC020]***	0x0074
Interrupt B Mode Control Register 021	***[IBIMC021]***	0x0075
Interrupt B Mode Control Register 022	***[IBIMC022]***	0x0076
Interrupt B Mode Control Register 023	***[IBIMC023]***	0x0077
Interrupt B Mode Control Register 024	***[IBIMC024]***	0x0078
Interrupt B Mode Control Register 025	***[IBIMC025]***	0x0079
Interrupt B Mode Control Register 026	***[IBIMC026]***	0x007A
Interrupt B Mode Control Register 027	***[IBIMC027]***	0x007B
Interrupt B Mode Control Register 028	***[IBIMC028]***	0x007C
Interrupt B Mode Control Register 029	***[IBIMC029]***	0x007D
Interrupt B Mode Control Register 030	***[IBIMC030]***	0x007E
Interrupt B Mode Control Register 031	***[IBIMC031]***	0x007F
Interrupt B Mode Control Register 032	***[IBIMC032]***	0x0080
Interrupt B Mode Control Register 033	***[IBIMC033]***	0x0081
Interrupt B Mode Control Register 034	***[IBIMC034]***	0x0082
Interrupt B Mode Control Register 035	***[IBIMC035]***	0x0083
Interrupt B Mode Control Register 036	***[IBIMC036]***	0x0084
Interrupt B Mode Control Register 037	***[IBIMC037]***	0x0085
Interrupt B Mode Control Register 038	***[IBIMC038]***	0x0086
Interrupt B Mode Control Register 039	***[IBIMC039]***	0x0087

Register name		Address (+BASE)
Interrupt B Mode Control Register 040	***[IBIMC040]***	0x0088
Interrupt B Mode Control Register 041	***[IBIMC041]***	0x0089
Interrupt B Mode Control Register 042	***[IBIMC042]***	0x008A
Interrupt B Mode Control Register 043	***[IBIMC043]***	0x008B
Interrupt B Mode Control Register 044	***[IBIMC044]***	0x008C
Interrupt B Mode Control Register 045	***[IBIMC045]***	0x008D
Interrupt B Mode Control Register 046	***[IBIMC046]***	0x008E
Interrupt B Mode Control Register 047	***[IBIMC047]***	0x008F
Interrupt B Mode Control Register 048	***[IBIMC048]***	0x0090
Interrupt B Mode Control Register 049	***[IBIMC049]***	0x0091
Interrupt B Mode Control Register 050	***[IBIMC050]***	0x0092
Interrupt B Mode Control Register 051	***[IBIMC051]***	0x0093
Interrupt B Mode Control Register 052	***[IBIMC052]***	0x0094
Interrupt B Mode Control Register 053	***[IBIMC053]***	0x0095
Interrupt B Mode Control Register 054	***[IBIMC054]***	0x0096
Interrupt B Mode Control Register 055	***[IBIMC055]***	0x0097
Interrupt B Mode Control Register 056	***[IBIMC056]***	0x0098
Interrupt B Mode Control Register 057	***[IBIMC057]***	0x0099
Interrupt B Mode Control Register 058	***[IBIMC058]***	0x009A
Interrupt B Mode Control Register 059	***[IBIMC059]***	0x009B
Interrupt B Mode Control Register 060	***[IBIMC060]***	0x009C
Interrupt B Mode Control Register 061	***[IBIMC061]***	0x009D

Note: Byte access is required when accessing to interrupt control register B.

5.1.3. Reset Flag Registers

Peripheral function		Channel/unit	Base address
Power control/reset	RLM	-	0x4003E400

Register name		Address (+BASE)
Reset Flag Register 0	***[RLMRSTFLG0]***	0x0002
Reset Flag Register 1	***[RLMRSTFLG1]***	0x0003

Note: Byte access is required when accessing to reset flag registers.

5.1.4. Interrupt Monitor Flag Registers

Peripheral function		Channel/unit	Base address
Interrupt Monitor	IMN	-	0x400F4F00

Register name		Address (+BASE)
Non-maskable Interrupt Monitor Flag Register	***[IMNFLGNMI]***	0x0000
Interrupt Monitor Flag Register 1	***[IMNFLG1]***	0x0004
Interrupt Monitor Flag Register 3	***[IMNFLG3]***	0x000C
Interrupt Monitor Flag Register 4	***[IMNFLG4]***	0x0010

5.1.5. NVIC Registers

Peripheral function		Channel/unit	Base address
NVIC Register	-	-	0xE000E000

Register name	Address (+Base)
SysTick Control and Status Register	0x0010
SysTick Reload Value Register	0x0014
SysTick Current Value Register	0x0018
SysTick Calibration Value Register	0x001C
Interrupt Set-Enable Register 0	0x0100
Interrupt Set-Enable Register 1	0x0104
Interrupt Set-Enable Register 2	0x0108
Interrupt Set-Enable Register 3	0x010C
Interrupt Set-Enable Register 4	0x0110
Interrupt Clear-Enable Register 0	0x0180
Interrupt Clear-Enable Register 1	0x0184
Interrupt Clear-Enable Register 2	0x0188
Interrupt Clear-Enable Register 3	0x018C
Interrupt Clear-Enable Register 4	0x0190
Interrupt Set-Pending Register 0	0x0200
Interrupt Set-Pending Register 1	0x0204
Interrupt Set-Pending Register 2	0x0208
Interrupt Set-Pending Register 3	0x020C
Interrupt Set-Pending Register 4	0x0210
Interrupt Clear-Pending Register 0	0x0280
Interrupt Clear-Pending Register 1	0x0284
Interrupt Clear-Pending Register 2	0x0288
Interrupt Clear-Pending Register 3	0x28C
Interrupt Clear-Pending Register 4	0x290
Interrupt Priority Register	0x0400 to 0x04C9
Vector Table Offset Register	0x0D08
Application Interrupt and Reset Control Register	0x0D0C
System Handler Priority Register	0x0D18 to 0x0D20
System Handler Control and State Register	0x0D24

5.2. Register Details

5.2.1. Interrupt Control Registers A

5.2.1.1. *[IANIC00]* (Non-maskable Interrupt A Control Register 00)

Bit	Bit symbol	After reset	Type	Function
7	INTNCLR	0	W	Detection flag clear control 0: - 1: Clear Read as "0".
6	-	0	R	Read as "0".
5	INTNFLG	0	R	Detection flag 0: Not detected 1: Detected
4:0	-	00101	R	Read as "00101".

5.2.1.2. *[IAIMC00]* to *[IAIMC20]* (Interrupt A Mode Control Register 00 to 20)

Bit	Bit symbol	After reset	Type	Function
7	INTNCLR	0	W	Falling edge detection flag clear control 0: - 1: Clear Read as "0".
6	INTPCLR	0	W	Rising edge detection flag clear control 0: - 1: Clear Read as "0".
5	INTNFLG	0	R	Falling edge detection flag 0: Not detected 1: Detected
4	INTPFLG	0	R	Rising edge detection flag 0: Not detected 1: Detected
3:1	INTMODE[2:0]	000	R/W	Interrupt detection level selection 000: Low level 001: High level 010: Falling edge 011: Rising edge 100: Both edge 101: Reserved 110: Reserved 111: Reserved
0	INTEN	0	R/W	Interrupt control 0: Interrupt detection disabled 1: Interrupt detection enabled

5.2.2. Interrupt Control Registers B

5.2.2.1. *[IBNIC00]* (Non-maskable Interrupt B Control Register 00)

Bit	Bit symbol	After reset	Type	Function
7	-	0	R	Read as "0".
6	INTPCLR	0	W	Detection flag clear control 0: - 1: Clear Read as "0".
5	-	0	R	Read as "0".
4	INTPFLG	0	R	Detection flag 0: Not detected 1: Detected
3:0	-	0111	R	Read as "0111".

5.2.2.2. *[IBIMC000]* to *[IBIMC061]* (Interrupt B Mode Control Register 000 to 061)

Bit	Bit symbol	After reset	Type	Function
7	-	0	R	Read as "0".
6	INTPCLR	0	W	Detection flag clear control 0: - 1: Clear Read as "0”.
5	-	0	R	Read as "0".
4	INTPFLG	0	R	Detection flag 0: Not detected 1: Detected
3:1	-	011	R	Read as "011".
0	INTEN	0	R/W	Interrupt control 0: Interrupt detection disabled 1: Interrupt detection enabled

5.2.3. Reset Flag Registers

5.2.3.1. *[RLMRSTFLG0]* (Reset Flag Register 0)

Bit	Bit symbol	After power on reset	Type	Function
7:6	-	Undefined	R	Read as an undefined value.
5	LVDRSTF	Undefined	R	LVD/PORF reset flag 0: - 1: Reset by LVD/PORF
			W	LVD/PORF reset flag 0: Clear 1: don't care
4	STOP2RSTF	Undefined	R	STOP2 reset flag 0: - 1: Reset by STOP2 mode releasing
			W	STOP2 reset flag 0: Clear 1: don't care
3	PINRSTF	Undefined	R	Reset pin flag 0: - 1: Reset by reset pin
			W	Reset pin flag 0: Clear 1: don't care
2:1	-	Undefined	R	Read as an undefined value.
			W	Write as "00".
0	PORSTF	1	R	Power-on reset flag 0: - 1: Reset by Power-on reset
			W	Power-on reset flag 0: Clear 1: don't care

Note: Reset flags except <PORSTF> become undefined after power-on reset is released. When releasing power-on reset is detected, set all the reset flags to "0" for initializing reset flags.

5.2.3.2. *[RLMRSTFLG1]* (Reset Flag Register 1)

Bit	Bit symbol	After power on reset	Type	Function
7:5	-	0	R	Read as "0".
4	SAMRSTF	Undefined	R	SAM reset flag 0: - 1: Reset by SAM
			W	SAM reset flag 0: Clear 1: don't care
3	OFDRSTF	0	R	OFD reset flag 0: - 1: Reset by OFD
			W	OFD reset flag 0: Clear 1: don't care
2	WDTRSTF	0	R	SIWDT reset flag 0: - 1: Reset by SIWDT
			W	SIWDT reset flag 0: Clear 1: don't care
1	LOCKRSTF	0	R	LOCKUP reset flag 0: - 1: Reset by LOCKUP
			W	LOCKUP reset flag 0: Clear 1: don't care
0	SYSRSTF	0	R	<SYSRESETREQ> reset flag 0: - 1: Reset by <SYSRESETREQ>
			W	<SYSRESETREQ> reset flag 0: Clear 1: don't care

5.2.4. Interrupt Monitor Flag Registers

5.2.4.1. *[IMNFLGNMI]* (Non-maskable Interrupt Monitor Flag Register)

Bit	Bit symbol	After reset	Type	Function
31:17	-	0	R	Read as "0".
16	INT016FLG	0	R	INTWDT0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
15:1	-	0	R	Read as "0".
0	INT000FLG	0	R	INTLVD interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

5.2.4.2. *[IMNFLG1]* (Interrupt Monitor Flag Register 1)

Bit	Bit symbol	After reset	Type	Function
31:16	-	0	R	Read as "0".
20	INT052FLG	0	R	INT20 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
19	INT051FLG	0	R	INT19 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
18	INT050FLG	0	R	INT18 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
17	INT049FLG	0	R	INT17 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
16	INT048FLG	0	R	INT16 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
15	INT047FLG	0	R	INT15 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
14	INT046FLG	0	R	INT14 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
13	INT045FLG	0	R	INT13 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
12	INT044FLG	0	R	INT12 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
11	INT043FLG	0	R	INT11 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
10	INT042FLG	0	R	INT10 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
9	INT041FLG	0	R	INT09 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
8	INT040FLG	0	R	INT08 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

Bit	Bit symbol	After reset	Type	Function
7	INT039FLG	0	R	INT07 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
6	INT038FLG	0	R	INT06 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
5	INT037FLG	0	R	INT05 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
4	INT036FLG	0	R	INT04 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
3	INT035FLG	0	R	INT03 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
2	INT034FLG	0	R	INT02 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
1	INT033FLG	0	R	INT01 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
0	INT032FLG	0	R	INT00 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

5.2.4.3. *[IMNFLG3]* (Interrupt Monitor Flag Register 3)

Bit	Bit symbol	After reset	Type	Function
31	INT127FLG	0	R	INTDMAATC (ch1) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
30	INT126FLG	0	R	INTDMAATC (ch0) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
29	INT125FLG	0	R	INTT32A04EVRYC interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
28	INT124FLG	0	R	INTT32A04BCAP1 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
27	INT123FLG	0	R	INTT32A04BCAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
26	INT122FLG	0	R	INTT32A04CCAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
25	INT121FLG	0	R	INTT32A04ACAP1 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
24	INT120FLG	0	R	INTT32A04ACAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
23	INT119FLG	0	R	INTT32A03EVRYC interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
22	INT118FLG	0	R	INTT32A03BCAP1 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
21	INT117FLG	0	R	INTT32A03BCAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
20	INT116FLG	0	R	INTT32A03CCAP0I interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
19	INT115FLG	0	R	INTT32A03ACAP1Iinterrupt detection flag 0: Interrupt not detected 1: Interrupt detected
18	INT114FLG	0	R	INTT32A03ACAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
17	INT113FLG	0	R	INTT32A02EVRYC interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
16	INT112FLG	0	R	INTT32A02BCAP1 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
15	INT111FLG	0	R	INTT32A02BCAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
14	INT110FLG	0	R	INTT32A02CCAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
13	INT109FLG	0	R	INTT32A02ACAP1 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
12	INT108FLG	0	R	INTT32A02ACAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

Bit	Bit symbol	After reset	Type	Function
11	INT107FLG	0	R	INTT32A01EVRYC interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
10	INT106FLG	0	R	INTT32A01BCAP1 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
9	INT105FLG	0	R	INTT32A01BCAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
8	INT104FLG	0	R	INTT32A01CCAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
7	INT103FLG	0	R	INTT32A01ACAP1 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
6	INT102FLG	0	R	I INTT32A01ACAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
5	INT101FLG	0	R	INTT32A00EVRYC interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
4	INT100FLG	0	R	INTT32A00BCAP1 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
3	INT099FLG	0	R	INTT32A00BCAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
2	INT098FLG	0	R	INTT32A00CCAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
1	INT097FLG	0	R	INTT32A00ACAP1 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
0	INT096FLG	0	R	INTT32A00ACAP0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

5.2.4.4. *[IMNFLG4]* (Interrupt Monitor Flag Register 4)

Bit	Bit symbol	After reset	Type	Function
31:9	-	0	R	Read as "0".
29	INT157FLG	0	R	INTDMAATC (ch31) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
28	INT156FLG	0	R	INTDMAATC (ch30) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
27	INT155FLG	0	R	I INTDMAATC (ch29) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
26	INT154FLG	0	R	INTDMAATC (ch28) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
25	INT153FLG	0	R	INTDMAATC (ch27) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
24	INT152FLG	0	R	INTDMAATC (ch26) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
23	INT151FLG	0	R	INTDMAATC (ch25) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
22	INT150FLG	0	R	INTDMAATC (ch24) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
21	INT149FLG	0	R	INTDMAATC (ch23) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
20	INT148FLG	0	R	I INTDMAATC (ch22) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
19	INT147FLG	0	R	INTDMAATC (ch21) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
18	INT146FLG	0	R	INTDMAATC (ch20) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
17	INT145FLG	0	R	INTDMAATC (ch19) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
16	INT144FLG	0	R	INTDMAATC (ch18) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
15	INT143FLG	0	R	INTDMAATC (ch17) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
14	INT142FLG	0	R	INTDMAATC (ch16) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
13	INT141FLG	0	R	INTDMAATC (ch15) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
12	INT140FLG	0	R	INTDMAATC (ch14) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
11	INT139FLG	0	R	INTDMAATC (ch13) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

Bit	Bit symbol	After reset	Type	Function
10	INT138FLG	0	R	INTDMAATC (ch12) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
9	INT137FLG	0	R	INTDMAATC (ch11) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
8	INT136FLG	0	R	INTDMAATC (ch10) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
7	INT135FLG	0	R	INTDMAATC (ch9) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
6	INT134FLG	0	R	INTDMAATC (ch8) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
5	INT133FLG	0	R	INTDMAATC (ch7) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
4	INT132FLG	0	R	INTDMAATC (ch6) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
3	INT131FLG	0	R	INTDMAATC (ch5) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
2	INT130FLG	0	R	INTDMAATC (ch4) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
1	INT129FLG	0	R	INTDMAATC (ch3) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
0	INT128FLG	0	R	INTDMAATC (ch2) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

5.2.5. NVIC Registers

5.2.5.1. SysTick Control and Status Register

Bit	Bit symbol	After reset	Type	Function
31:17	-	0	R	Read as "0".
16	COUNTFLAG	0	R/W	0: Timer not counted to “0” 1: Timer counted to “0” Returns "1" if timer counted to "0" since last time this was read. Clears on read of any part of the SysTick Control and Status Register.
15:3	-	0	R	Read as "0".
2	CLKSOURCE	0	R/W	0: External reference clock (fosc/64) 1: CPU clock(fsys)
1	TICKINT	0	R/W	0: Do not pend SysTick 1: Pend SysTick
0	ENABLE	0	R/W	0: Disable 1: Enable If "1" is set, it re-load to a counter with the value of the Reload Value Register and starts operation.

5.2.5.2. SysTick Reload Value Register

Bit	Bit symbol	After reset	Type	Function
31:24	-	0	R	Read as "0".
23:0	RELOAD[23:0]	Undefined	R/W	Reload value Set the value to load into the SysTick Current Value Register when the timer reaches "0".

5.2.5.3. SysTick Current Value Register

Bit	Bit symbol	After reset	Type	Function
31:24	-	0	R	Read as "0".
23:0	CURRENT[23:0]	Undefined	R	Current SysTick timer value
			W	Clear Writing to this bit symbol with any value clears timer counter to “0”. Clearing this bit symbol also clears the <COUNTFLAG> of the SysTick Control and Status Register.

5.2.5.4. SysTick Calibration Value Register

Bit	Bit symbol	After reset	Type	Function
31	NOREF	0	R	0: Reference clock provided 1: No reference clock
30	SKEW	1	R	0: Calibration value is 10 ms. 1: Calibration value is not 10ms.
29:24	-	0	R	Read as "0".
23:0	TENMS	0x000000	R	Calibration value (Note)

Note: This product does not prepare the calibration value.

5.2.5.5. Interrupt Control Registers

Following registers will be used to control each interrupt source; interrupt set-enable register, interrupt clear-enable register, interrupt set-pending register, and interrupt clear-pending register.
Each bit corresponds to the specified number of interrupts.

(1) Interrupt Set-Enable Register

This register can be used to enable interrupts and check the enable/disable condition.
Writing "1" to a bit in this register enables the corresponding interrupt.
Writing "0" has no effect.
Reading the bits can be checked the enable/disable condition of the corresponding interrupts.
Writing "1" to a corresponding bit in the interrupt clear-enable register clears the bit in this register.

(a) Interrupt Set-Enable Register 0

Bit	Bit symbol	After reset	Type	Function
31	SETENA (Interrupt 31)	0	R/W	[Write] 1: Enable interrupt. [Read] 0: Interrupt is disabled. 1: Interrupt is enabled.
30	SETENA (Interrupt 30)	0		
29	SETENA (Interrupt 29)	0		
28	SETENA (Interrupt 28)	0		
27	SETENA (Interrupt 27)	0		
26	SETENA (Interrupt 26)	0		
25	SETENA (Interrupt 25)	0		
24	SETENA (Interrupt 24)	0		
23	SETENA (Interrupt 23)	0		
22	SETENA (Interrupt 22)	0		
21	SETENA (Interrupt 21)	0		
20	SETENA (Interrupt 20)	0		
19	SETENA (Interrupt 19)	0		
18	SETENA (Interrupt 18)	0		
17	SETENA (Interrupt 17)	0		
16	SETENA (Interrupt 16)	0		
15	SETENA (Interrupt 15)	0		
14	SETENA (Interrupt 14)	0		
13	SETENA (Interrupt 13)	0		
12	SETENA (Interrupt 12)	0		
11	SETENA (Interrupt 11)	0		
10	SETENA (Interrupt 10)	0		
9	SETENA (Interrupt 9)	0		
8	SETENA (Interrupt 8)	0		
7	SETENA (Interrupt 7)	0		
6	SETENA (Interrupt 6)	0		
5	SETENA (Interrupt 5)	0		
4	SETENA (Interrupt 4)	0		
3	SETENA (Interrupt 3)	0		
2	SETENA (Interrupt 2)	0		
1	SETENA (Interrupt 1)	0		
0	SETENA (Interrupt 0)	0		

(b) Interrupt Set-Enable Register 1

Bit	Bit symbol	After reset	Type	Function
31	SETENA (Interrupt 63)	0	R/W	[Write] 1: Enable interrupt. [Read] 0: Interrupt is disabled. 1: Interrupt is enabled.
30	SETENA (Interrupt 62)	0		
29	SETENA (Interrupt 61)	0		
28	SETENA (Interrupt 60)	0		
27	SETENA (Interrupt 59)	0		
26	SETENA (Interrupt 58)	0		
25	SETENA (Interrupt 57)	0		
24	SETENA (Interrupt 56)	0		
23	SETENA (Interrupt 55)	0		
22	SETENA (Interrupt 54)	0		
21	SETENA (Interrupt 53)	0		
20	SETENA (Interrupt 52)	0		
19	SETENA (Interrupt 51)	0		
18	SETENA (Interrupt 50)	0		
17	SETENA (Interrupt 49)	0		
16	SETENA (Interrupt 48)	0		
15	SETENA (Interrupt 47)	0		
14	SETENA (Interrupt 46)	0		
13	SETENA (Interrupt 45)	0		
12	SETENA (Interrupt 44)	0		
11	SETENA (Interrupt 43)	0		
10	SETENA (Interrupt 42)	0		
9	SETENA (Interrupt 41)	0		
8	SETENA (Interrupt 40)	0		
7	SETENA (Interrupt 39)	0		
6	SETENA (Interrupt 38)	0		
5	SETENA (Interrupt 37)	0		
4	SETENA (Interrupt 36)	0		
3	SETENA (Interrupt 35)	0		
2	SETENA (Interrupt 34)	0		
1	SETENA (Interrupt 33)	0	R/W	Write as "0".
0	SETENA (Interrupt 32)	0	R/W	[Write] 1: Enable interrupt. [Read] 0: Interrupt is disabled. 1: Interrupt is enabled.

(c) Interrupt Set-Enable Register 2

Bit	Bit symbol	After reset	Type	Function
31	SETENA (Interrupt 95)	0	R/W	[Write] 1: Enable interrupt. [Read] 0: Interrupt is disabled. 1: Interrupt is enabled.
30	SETENA (Interrupt 94)	0		
29	SETENA (Interrupt 93)	0		
28	SETENA (Interrupt 92)	0		
27	SETENA (Interrupt 91)	0		
26	SETENA (Interrupt 90)	0		
25	SETENA (Interrupt 89)	0		
24	SETENA (Interrupt 88)-	0		
23	SETENA (Interrupt 87)	0		
22	SETENA (Interrupt 86)	0		
21	SETENA (Interrupt 85)	0		
20	SETENA (Interrupt 84)	0		
19	SETENA (Interrupt 83)	0		
18	SETENA (Interrupt 82)	0		
17	SETENA (Interrupt 81)	0		
16	SETENA (Interrupt 80)	0		
15	SETENA (Interrupt 79)	0		
14	SETENA (Interrupt 78)	0		
13	SETENA (Interrupt 77)	0		
12	SETENA (Interrupt 76)	0		
11	SETENA (Interrupt 75)	0		
10	SETENA (Interrupt 74)	0		
9	SETENA (Interrupt 73)	0		
8	SETENA (Interrupt 72)	0		
7	SETENA (Interrupt 71)	0		
6	SETENA (Interrupt 70)	0		
5	SETENA (Interrupt 69)	0		
4	SETENA (Interrupt 68)	0		
3	SETENA (Interrupt 67)	0		
2	SETENA (Interrupt 66)	0		
1	SETENA (Interrupt 65)	0		
0	SETENA (Interrupt 64)	0		

(d) Interrupt Set-Enable Register 3

Bit	Bit symbol	After reset	Type	Function
31	SETENA (Interrupt 127)	0	R/W	[Write] 1: Enable interrupt. [Read] 0: Interrupt is disabled. 1: Interrupt is enabled.
30	SETENA (Interrupt 126)	0		
29	SETENA (Interrupt 125)	0		
28	SETENA (Interrupt 124)	0		
27	SETENA (Interrup123)	0		
26	SETENA (Interrupt 122)	0		
25	SETENA (Interrupt 121)	0		
24	SETENA (Interrupt 120)	0		
23	SETENA (Interrupt 119)	0		
22	SETENA (Interrupt 118)	0		
21	SETENA (Interrupt 117)	0		
20	SETENA (Interrupt 116)	0		
19	SETENA (Interrupt 115)	0		
18	SETENA (Interrupt 114)	0		
17	SETENA (Interrupt 113)	0		
16	SETENA (Interrupt 112)	0		
15	SETENA (Interrupt 111)	0		
14	SETENA (Interrupt 110)	0		
13	SETENA (Interrupt 109)	0		
12	SETENA (Interrupt 108)	0		
11	SETENA (Interrupt 107)	0		
10	SETENA (Interrupt 106)	0		
9	SETENA (Interrupt 105)	0		
8	SETENA (Interrupt 104)	0		
7	SETENA (Interrupt 103)	0		
6	SETENA (Interrupt 102)	0		
5	SETENA (Interrupt 101)	0		
4	SETENA (Interrupt 100)	0		
3	SETENA (Interrupt 99)	0		
2	SETENA (Interrupt 98)	0		
1	SETENA (Interrupt 97)	0		
0	SETENA (Interrupt 96)	0		

(e) Interrupt Set-Enable Register 4

Bit	Bit symbol	After reset	Type	Function
31:30	-	0	R	Read as “0”.
29	SETENA (Interrupt 157)	0	R/W	[Write] 1: Enable interrupt. [Read] 0: Interrupt is disabled. 1: Interrupt is enabled.
28	SETENA (Interrupt 156)	0		
27	SETENA (Interrupt 155)	0		
26	SETENA (Interrupt 154)	0		
25	SETENA (Interrupt 153)	0		
24	SETENA (Interrupt 152)	0		
23	SETENA (Interrupt 151)	0		
22	SETENA (Interrupt 150)	0		
21	SETENA (Interrupt 149)	0		
20	SETENA (Interrupt 148)	0		
19	SETENA (Interrupt 147)	0		
18	SETENA (Interrupt 146)	0		
17	SETENA (Interrupt 145)	0		
16	SETENA (Interrupt 144)	0		
15	SETENA (Interrupt 143)	0		
14	SETENA (Interrupt 142)	0		
13	SETENA (Interrupt 141)	0		
12	SETENA (Interrupt 140)	0		
11	SETENA (Interrupt 139)	0		
10	SETENA (Interrupt 138)	0		
9	SETENA (Interrupt 137)	0		
8	SETENA (Interrupt 136)	0		
7	SETENA (Interrupt 135)	0		
6	SETENA (Interrupt 134)	0		
5	SETENA (Interrupt 133)	0		
4	SETENA (Interrupt 132)	0		
3	SETENA (Interrupt 131)	0		
2	SETENA (Interrupt 130)	0		
1	SETENA (Interrupt 129)	0		
0	SETENA (Interrupt 128)	0		

(2) Interrupt Clear-Enable Register

This register can be used to disable interrupts and check the enable/disable condition.
Writing "1" to a bit in this register disables the corresponding interrupt.
Writing "0" has no effect.
Reading the bits can see the enable/disable condition of the corresponding interrupts.

(a) Interrupt Clear-Enable Register 0

Bit	Bit symbol	After reset	Type	Function
31	CLRENA (Interrupt 31)	0	R/W	[Write] 1: Disable interrupt. [Read] 0: Interrupt is disabled. 1: Interrupt is enabled.
30	CLRENA (Interrupt 30)	0		
29	CLRENA (Interrupt 29)	0		
28	CLRENA (Interrupt 28)	0		
27	CLRENA (Interrupt 27)	0		
26	CLRENA (Interrupt 26)	0		
25	CLRENA (Interrupt 25)	0		
24	CLRENA (Interrupt 24)	0		
23	CLRENA (Interrupt 23)	0		
22	CLRENA (Interrupt 22)	0		
21	CLRENA (Interrupt 21)	0		
20	CLRENA (Interrupt 20)	0		
19	CLRENA (Interrupt 19)	0		
18	CLRENA (Interrupt 18)	0		
17	CLRENA (Interrupt 17)	0		
16	CLRENA (Interrupt 16)	0		
15	CLRENA (Interrupt 15)	0		
14	CLRENA (Interrupt 14)	0		
13	CLRENA (Interrupt 13)	0		
12	CLRENA (Interrupt 12)	0		
11	CLRENA (Interrupt 11)	0		
10	CLRENA (Interrupt 10)	0		
9	CLRENA (Interrupt 9)	0		
8	CLRENA (Interrupt 8)	0		
7	CLRENA (Interrupt 7)	0		
6	CLRENA (Interrupt 6)	0		
5	CLRENA (Interrupt 5)	0		
4	CLRENA (Interrupt 4)	0		
3	CLRENA (Interrupt 3)	0		
2	CLRENA (Interrupt 2)	0		
1	CLRENA (Interrupt 1)	0		
0	CLRENA (Interrupt 0)	0		

(b) Interrupt Clear-Enable Register 1

Bit	Bit symbol	After reset	Type	Function
31	CLRENA (Interrupt 63)	0	R/W	[Write] 1: Disable interrupt. [Read] 0: Interrupt is disabled. 1: Interrupt is enabled.
30	CLRENA (Interrupt 62)	0		
29	CLRENA (Interrupt 61)	0		
28	CLRENA (Interrupt 60)	0		
27	CLRENA (Interrupt 59)	0		
26	CLRENA (Interrupt 58)	0		
25	CLRENA (Interrupt 57)	0		
24	CLRENA (Interrupt 56)	0		
23	CLRENA (Interrupt 55)	0		
22	CLRENA (Interrupt 54)	0		
21	CLRENA (Interrupt 53)	0		
20	CLRENA (Interrupt 52)	0		
19	CLRENA (Interrupt 51)	0		
18	CLRENA (Interrupt 50)	0		
17	CLRENA (Interrupt 49)	0		
16	CLRENA (Interrupt 48)	0		
15	CLRENA (Interrupt 47)	0		
14	CLRENA (Interrupt 46)	0		
13	CLRENA (Interrupt 45)	0		
12	CLRENA (Interrupt 44)	0		
11	CLRENA (Interrupt 43)	0		
10	CLRENA (Interrupt 42)	0		
9	CLRENA (Interrupt 41)	0		
8	CLRENA (Interrupt 40)	0		
7	CLRENA (Interrupt 39)	0		
6	CLRENA (Interrupt 38)	0		
5	CLRENA (Interrupt 37)	0		
4	CLRENA (Interrupt 36)	0		
3	CLRENA (Interrupt 35)	0		
2	CLRENA (Interrupt 34)	0		
1	CLRENA (Interrupt 33)	0	R/W	Write as "1".
0	CLRENA (Interrupt 32)	0	R/W	[Write] 1: Disable interrupt. [Read] 0: Interrupt is disabled. 1: Interrupt is enabled.

(c) Interrupt Clear-Enable Register 2

Bit	Bit symbol	After reset	Type	Function
31	CLRENA (Interrupt 95)	0	R/W	[Write] 1: Disable interrupt. [Read] 0: Interrupt is disabled. 1: Interrupt is enabled.
30	CLRENA (Interrupt 94)	0		
29	CLRENA (Interrupt 93)	0		
28	CLRENA (Interrupt 92)	0		
27	CLRENA (Interrupt 91)	0		
26	CLRENA (Interrupt 90)	0		
25	CLRENA (Interrupt 89)	0		
24	CLRENA (Interrupt 89)-	0		
23	CLRENA (Interrupt 87)	0		
22	CLRENA (Interrupt 86)	0		
21	CLRENA (Interrupt 85)	0		
20	CLRENA (Interrupt 84)	0		
19	CLRENA (Interrupt 83)	0		
18	CLRENA (Interrupt 82)	0		
17	CLRENA (Interrupt 81)	0		
16	CLRENA (Interrupt 80)	0		
15	CLRENA (Interrupt 79)	0		
14	CLRENA (Interrupt 78)	0		
13	CLRENA (Interrupt 77)	0		
12	CLRENA (Interrupt 76)	0		
11	CLRENA (Interrupt 75)	0		
10	CLRENA (Interrupt 74)	0		
9	CLRENA (Interrupt 73)	0		
8	CLRENA (Interrupt 72)	0		
7	CLRENA (Interrupt 71)	0		
6	CLRENA (Interrupt 70)	0		
5	CLRENA (Interrupt 69)	0		
4	CLRENA (Interrupt 68)	0		
3	CLRENA (Interrupt 67)	0		
2	CLRENA (Interrupt 66)	0		
1	CLRENA (Interrupt 65)	0		
0	CLRENA (Interrupt 64)	0		

(d) Interrupt Clear-Enable Register 3

Bit	Bit symbol	After reset	Type	Function
31	CLRENA (Interrupt 127)	0	R/W	[Write] 1: Disable interrupt. [Read] 0: Interrupt is disabled. 1: Interrupt is enabled.
30	CLRENA (Interrupt 126)	0		
29	CLRENA (Interrupt 125)	0		
28	CLRENA (Interrupt 124)	0		
27	CLRENA (Interrupt 123	0		
26	CLRENA (Interrupt 122)	0		
25	CLRENA (Interrupt 121)	0		
24	CLRENA (Interrupt 120)	0		
23	CLRENA (Interrupt 119)	0		
22	CLRENA (Interrupt 118)	0		
21	CLRENA (Interrupt 117)	0		
20	CLRENA (Interrupt 116)	0		
19	CLRENA (Interrupt 115)	0		
18	CLRENA (Interrupt 114)	0		
17	CLRENA (Interrupt 113)	0		
16	CLRENA (Interrupt 112)	0		
15	CLRENA (Interrupt 111)	0		
14	CLRENA (Interrupt 110)	0		
13	CLRENA (Interrupt 109)	0		
12	CLRENA (Interrupt 108)	0		
11	CLRENA (Interrupt 107)	0		
10	CLRENA (Interrupt 106)	0		
9	CLRENA (Interrupt 105)	0		
8	CLRENA (Interrupt 104)	0		
7	CLRENA (Interrupt 103)	0		
6	CLRENA (Interrupt 102)	0		
5	CLRENA (Interrupt 101)	0		
4	CLRENA (Interrupt 100)	0		
3	CLRENA (Interrupt 99)	0		
2	CLRENA (Interrupt 98)	0		
1	CLRENA (Interrupt 97)	0		
0	CLRENA (Interrupt 96)	0		

(e) Interrupt Clear-Enable Register 4

Bit	Bit symbol	After reset	Type	Function
31:30	-	0	R/W	[Write] 1: Disable interrupt. [Read] 0: Interrupt is disabled. 1: Interrupt is enabled.
29	CLRENA (Interrupt 157)	0		
28	CLRENA (Interrupt 156)	0		
27	CLRENA (Interrupt 155)	0		
26	CLRENA (Interrupt 154)	0		
25	CLRENA (Interrupt 153)	0		
24	CLRENA (Interrupt 152)	0		
23	CLRENA (Interrupt 151)	0		
22	CLRENA (Interrupt 150)	0		
21	CLRENA (Interrupt 149)	0		
20	CLRENA (Interrupt 148)	0		
19	CLRENA (Interrupt 147)	0		
18	CLRENA (Interrupt 146)	0		
17	CLRENA (Interrupt 145)	0		
16	CLRENA (Interrupt 144)	0		
15	CLRENA (Interrupt 143)	0		
14	CLRENA (Interrupt 142)	0		
13	CLRENA (Interrupt 141)	0		
12	CLRENA (Interrupt 140)	0		
11	CLRENA (Interrupt 139)	0		
10	CLRENA (Interrupt 138)	0		
9	CLRENA (Interrupt 137)	0		
8	CLRENA (Interrupt 136)	0		
7	CLRENA (Interrupt 135)	0		
6	CLRENA (Interrupt 134)	0		
5	CLRENA (Interrupt 133)	0		
4	CLRENA (Interrupt 132)	0		
3	CLRENA (Interrupt 131)	0		
2	CLRENA (Interrupt 130)	0		
1	CLRENA (Interrupt 129)	0		
0	CLRENA (Interrupt 128)	0		

(3) Interrupt Set-Pending Register

This register can be used to pended interrupts and check the pending condition.
Writing "1" to a bit in this register pends the corresponding interrupt. However, writing "1" has no effect on an Interrupt that is already pended or is disabled.
Writing "0" has no effect.
Reading the bit can be checked the pending condition of the corresponding interrupts.
Writing "1" to a corresponding bit in the interrupt clear-pending register clears the bit in this register.

(a) Interrupt Set-Pending Register 0

Bit	Bit symbol	After reset	Type	Function
31	SETPEND (Interrupt 31)	Undefined	R/W	[Write] 1: Pend interrupt. [Read] 0: Not pending 1: Pending
30	SETPEND (Interrupt 30)	Undefined		
29	SETPEND (Interrupt 29)	Undefined		
28	SETPEND (Interrupt 28)	Undefined		
27	SETPEND (Interrupt 27)	Undefined		
26	SETPEND (Interrupt 26)	Undefined		
25	SETPEND (Interrupt 25)	Undefined		
24	SETPEND (Interrupt 24)	Undefined		
23	SETPEND (Interrupt 23)	Undefined		
22	SETPEND (Interrupt 22)	Undefined		
21	SETPEND (Interrupt 21)	Undefined		
20	SETPEND (Interrupt 20)	Undefined		
19	SETPEND (Interrupt 19)	Undefined		
18	SETPEND (Interrupt 18)	Undefined		
17	SETPEND (Interrupt 17)	Undefined		
16	SETPEND (Interrupt 16)	Undefined		
15	SETPEND (Interrupt 15)	Undefined		
14	SETPEND (Interrupt 14)	Undefined		
13	SETPEND (Interrupt 13)	Undefined		
12	SETPEND (Interrupt 12)	Undefined		
11	SETPEND (Interrupt 11)	Undefined		
10	SETPEND (Interrupt 10)	Undefined		
9	SETPEND (Interrupt 9)	Undefined		
8	SETPEND (Interrupt 8)	Undefined		
7	SETPEND (Interrupt 7)	Undefined		
6	SETPEND (Interrupt 6)	Undefined		
5	SETPEND (Interrupt 5)	Undefined		
4	SETPEND (Interrupt 4)	Undefined		
3	SETPEND (Interrupt 3)	Undefined		
2	SETPEND (Interrupt 2)	Undefined		
1	SETPEND (Interrupt 1)	Undefined		
0	SETPEND (Interrupt 0)	Undefined		

(b) Interrupt Set-Pending Register 1

Bit	Bit symbol	After reset	Type	Function
31	SETPEND (Interrupt 63)	Undefined	R/W	[Write] 1: Pend interrupt. [Read] 0: Not pending 1: Pending
30	SETPEND (Interrupt 62)	Undefined		
29	SETPEND (Interrupt 61)	Undefined		
28	SETPEND (Interrupt 60)	Undefined		
27	SETPEND (Interrupt 59)	Undefined		
26	SETPEND (Interrupt 58)	Undefined		
25	SETPEND (Interrupt 57)	Undefined		
24	SETPEND (Interrupt 56)	Undefined		
23	SETPEND (Interrupt 55)	Undefined		
22	SETPEND (Interrupt 54)	Undefined		
21	SETPEND (Interrupt 53)	Undefined		
20	SETPEND (Interrupt 52)	Undefined		
19	SETPEND (Interrupt 51)	Undefined		
18	SETPEND (Interrupt 50)	Undefined		
17	SETPEND (Interrupt 49)	Undefined		
16	SETPEND (Interrupt 48)	Undefined		
15	SETPEND (Interrupt 47)	Undefined		
14	SETPEND (Interrupt 46)	Undefined		
13	SETPEND (Interrupt 45)	Undefined		
12	SETPEND (Interrupt 44)	Undefined		
11	SETPEND (Interrupt 43)	Undefined		
10	SETPEND (Interrupt 42)	Undefined		
9	SETPEND (Interrupt 41)	Undefined		
8	SETPEND (Interrupt 40)	Undefined		
7	SETPEND (Interrupt 39)	Undefined		
6	SETPEND (Interrupt 38)	Undefined		
5	SETPEND (Interrupt 37)	Undefined		
4	SETPEND (Interrupt 36)	Undefined		
3	SETPEND (Interrupt 35)	Undefined		
2	SETPEND (Interrupt 34)	Undefined		
1	SETPEND (Interrupt 33)	Undefined	R/W	Write as "0".
0	SETPEND (Interrupt 32)	Undefined	R/W	[Write] 1: Pend interrupt. [Read] 0: Not pending 1: Pending

(c) Interrupt Set-Pending Register 2

Bit	Bit symbol	After reset	Type	Function
31	SETPEND (Interrupt 95)	Undefined	R/W	[Write] 1: Pend interrupt. [Read] 0: Not pending 1: Pending
30	SETPEND (Interrupt 94)	Undefined		
29	SETPEND (Interrupt 93)	Undefined		
28	SETPEND (Interrupt 92)	Undefined		
27	SETPEND (Interrupt 91)	Undefined		
26	SETPEND (Interrupt 90)	Undefined		
25	SETPEND (Interrupt 89)	Undefined		
24	-SETPEND (Interrupt 88)	Undefined		
23	SETPEND (Interrupt 87)	Undefined		
22	SETPEND (Interrupt 86)	Undefined		
21	SETPEND (Interrupt 85)	Undefined		
20	SETPEND (Interrupt 84)	Undefined		
19	SETPEND (Interrupt 83)	Undefined		
18	SETPEND (Interrupt 82)	Undefined		
17	SETPEND (Interrupt 81)	Undefined		
16	SETPEND (Interrupt 80)	Undefined		
15	SETPEND (Interrupt 79)	Undefined		
14	SETPEND (Interrupt 78)	Undefined		
13	SETPEND (Interrupt 77)	Undefined		
12	SETPEND (Interrupt 76)	Undefined		
11	SETPEND (Interrupt 75)	Undefined		
10	SETPEND (Interrupt 74)	Undefined		
9	SETPEND (Interrupt 73)	Undefined		
8	SETPEND (Interrupt 72)	Undefined		
7	SETPEND (Interrupt 71)	Undefined		
6	SETPEND (Interrupt 70)	Undefined		
5	SETPEND (Interrupt 69)	Undefined		
4	SETPEND (Interrupt 68)	Undefined		
3	SETPEND (Interrupt 67)	Undefined		
2	SETPEND (Interrupt 66)	Undefined		
1	SETPEND (Interrupt 65)	Undefined		
0	SETPEND (Interrupt 64)	Undefined		

(d) Interrupt Set-Pending Register 3

Bit	Bit symbol	After reset	Type	Function
31	SETPEND (Interrupt 127)	Undefined	R/W	[Write] 1: Pend interrupt. [Read] 0: Not pending 1: Pending
30	SETPEND (Interrupt 126)	Undefined		
29	SETPEND (Interrupt 125)	Undefined		
28	SETPEND (Interrupt 124)	Undefined		
27	SETPEND (Interrupt 123)	Undefined		
26	SETPEND (Interrupt 122)	Undefined		
25	SETPEND (Interrupt 121)	Undefined		
24	SETPEND (Interrupt 120)	Undefined		
23	SETPEND (Interrupt 119)	Undefined		
22	SETPEND (Interrupt 118)	Undefined		
21	SETPEND (Interrupt 117)	Undefined		
20	SETPEND (Interrupt 116)	Undefined		
19	SETPEND (Interrupt 115)	Undefined		
18	SETPEND (Interrupt 114)	Undefined		
17	SETPEND (Interrupt 113)	Undefined		
16	SETPEND (Interrupt 112)	Undefined		
15	SETPEND (Interrupt 111)	Undefined		
14	SETPEND (Interrupt 110)	Undefined		
13	SETPEND (Interrupt 109)	Undefined		
12	SETPEND (Interrupt 108)	Undefined		
11	SETPEND (Interrupt 107)	Undefined		
10	SETPEND (Interrupt 106)	Undefined		
9	SETPEND (Interrupt 105)	Undefined		
8	SETPEND (Interrupt 104)	Undefined		
7	SETPEND (Interrupt 103)	Undefined		
6	SETPEND (Interrupt 102)	Undefined		
5	SETPEND (Interrupt 101)	Undefined		
4	SETPEND (Interrupt 100)	Undefined		
3	SETPEND (Interrupt 99)	Undefined		
2	SETPEND (Interrupt 98)	Undefined		
1	SETPEND (Interrupt 97)	Undefined		
0	SETPEND (Interrupt 96)	Undefined		

(e) Interrupt Set-Pending Register 4

Bit	Bit symbol	After reset	Type	Function
31:30	-	0	R	Read as “0”
29	SETPEND (Interrupt 157)	Undefined	R/W	[Write] 1: Pend interrupt. [Read] 0: Not pending 1: Pending
28	SETPEND (Interrupt 156)	Undefined		
27	SETPEND (Interrupt 155)	Undefined		
26	SETPEND (Interrupt 154)	Undefined		
25	SETPEND (Interrupt 153)	Undefined		
24	SETPEND (Interrupt 152)	Undefined		
23	SETPEND (Interrupt 151)	Undefined		
22	SETPEND (Interrupt 150)	Undefined		
21	SETPEND (Interrupt 149)	Undefined		
20	SETPEND (Interrupt 148)	Undefined		
19	SETPEND (Interrupt 147)	Undefined		
18	SETPEND (Interrupt 146)	Undefined		
17	SETPEND (Interrupt 145)	Undefined		
16	SETPEND (Interrupt 144)	Undefined		
15	SETPEND (Interrupt 143)	Undefined		
14	SETPEND (Interrupt 142)	Undefined		
13	SETPEND (Interrupt 141)	Undefined		
12	SETPEND (Interrupt 140)	Undefined		
11	SETPEND (Interrupt 139)	Undefined		
10	SETPEND (Interrupt 138)	Undefined		
9	SETPEND (Interrupt 137)	Undefined		
8	SETPEND (Interrupt 136)	Undefined		
7	SETPEND (Interrupt 135)	Undefined		
6	SETPEND (Interrupt 134)	Undefined		
5	SETPEND (Interrupt 133)	Undefined		
4	SETPEND (Interrupt 132)	Undefined		
3	SETPEND (Interrupt 131)	Undefined		
2	SETPEND (Interrupt 130)	Undefined		
1	SETPEND (Interrupt 129)	Undefined		
0	SETPEND (Interrupt 128)	Undefined		

(4) Interrupt Clear-Pending Register

This register can be used to clear pending interrupts and check the pending condition.
Writing "1" to a bit in this register clears the pending condition of the corresponding interrupt. However, writing "1" has no effect on an Interrupt that is already being serviced.
Writing "0" has no effect.
Reading the bit can be checked the pending condition of the corresponding interrupts.

(a) Interrupt Clear-Pending Register 0

Bit	Bit symbol	After reset	Type	function
31	CLRPEND (Interrupt 31)	Undefined	R/W	[Write] 1: Clear pending interrupt. [Read] 0: Not pending 1: Pending
30	CLRPEND (Interrupt 30)	Undefined		
29	CLRPEND (Interrupt 29)	Undefined		
28	CLRPEND (Interrupt 28)	Undefined		
27	CLRPEND (Interrupt 27)	Undefined		
26	CLRPEND (Interrupt 26)	Undefined		
25	CLRPEND (Interrupt 25)	Undefined		
24	CLRPEND (Interrupt 24)	Undefined		
23	CLRPEND (Interrupt 23)	Undefined		
22	CLRPEND (Interrupt 22)	Undefined		
21	CLRPEND (Interrupt 21)	Undefined		
20	CLRPEND (Interrupt 20)	Undefined		
19	CLRPEND (Interrupt 19)	Undefined		
18	CLRPEND (Interrupt 18)	Undefined		
17	CLRPEND (Interrupt 17)	Undefined		
16	CLRPEND (Interrupt 16)	Undefined		
15	CLRPEND (Interrupt 15)	Undefined		
14	CLRPEND (Interrupt 14)	Undefined		
13	CLRPEND (Interrupt 13)	Undefined		
12	CLRPEND (Interrupt 26)	Undefined		
11	CLRPEND (Interrupt 26)	Undefined		
10	CLRPEND (Interrupt 10)	Undefined		
9	CLRPEND (Interrupt 9)	Undefined		
8	CLRPEND (Interrupt 8)	Undefined		
7	CLRPEND (Interrupt 7)	Undefined		
6	CLRPEND (Interrupt 6)	Undefined		
5	CLRPEND (Interrupt 5)	Undefined		
4	CLRPEND (Interrupt 4)	Undefined		
3	CLRPEND (Interrupt 3)	Undefined		
2	CLRPEND (Interrupt 2)	Undefined		
1	CLRPEND (Interrupt 1)	Undefined		
0	CLRPEND (Interrupt 0)	Undefined		

(b) Interrupt Clear-Pending Register 1

Bit	Bit symbol	After reset	Type	Function
31	CLRPEND (Interrupt 63)	Undefined	R/W	[Write] 1: Clear pending interrupt. [Read] 0: Not pending 1: Pending
30	CLRPEND (Interrupt 62)	Undefined		
29	CLRPEND (Interrupt 61)	Undefined		
28	CLRPEND (Interrupt 60)	Undefined		
27	CLRPEND (Interrupt 59)	Undefined		
26	CLRPEND (Interrupt 58)	Undefined		
25	CLRPEND (Interrupt 57)	Undefined		
24	CLRPEND (Interrupt 56)	Undefined		
23	CLRPEND (Interrupt 55)	Undefined		
22	CLRPEND (Interrupt 54)	Undefined		
21	CLRPEND (Interrupt 53)	Undefined		
20	CLRPEND (Interrupt 52)	Undefined		
19	CLRPEND (Interrupt 51)	Undefined		
18	CLRPEND (Interrupt 50)	Undefined		
17	CLRPEND (Interrupt 49)	Undefined		
16	CLRPEND (Interrupt 48)	Undefined		
15	CLRPEND (Interrupt 47)	Undefined		
14	CLRPEND (Interrupt 46)	Undefined		
13	CLRPEND (Interrupt 45)	Undefined		
12	CLRPEND (Interrupt 44)	Undefined		
11	CLRPEND (Interrupt 43)	Undefined		
10	CLRPEND (Interrupt 42)	Undefined		
9	CLRPEND (Interrupt 41)	Undefined		
8	CLRPEND (Interrupt 40)	Undefined		
7	CLRPEND (Interrupt 39)	Undefined		
6	CLRPEND (Interrupt 38)	Undefined		
5	CLRPEND (Interrupt 37)	Undefined		
4	CLRPEND (Interrupt 36)	Undefined		
3	CLRPEND (Interrupt 35)	Undefined		
2	CLRPEND (Interrupt 34)	Undefined		
1	CLRPEND (Interrupt 33)	Undefined	R/W	Write as “1”.
0	CLRPEND (Interrupt 32)	Undefined	R/W	[Write] 1: Clear pending interrupt. [Read] 0: Not pending 1: Pending

(c) Interrupt Clear-Pending Register 2

Bit	Bit symbol	After reset	Type	Function
31	CLRPEND (Interrupt 95)	Undefined	R/W	[Write] 1: Clear pending interrupt. [Read] 0: Not pending 1: Pending
30	CLRPEND (Interrupt 94)	Undefined		
29	CLRPEND (Interrupt 93)	Undefined		
28	CLRPEND (Interrupt 92)	Undefined		
27	CLRPEND (Interrupt 91)	Undefined		
26	CLRPEND (Interrupt 90)	Undefined		
25	CLRPEND (Interrupt 89)	Undefined		
24	CLRPEND (Interrupt 88)	Undefined		
23	CLRPEND (Interrupt 87)	Undefined		
22	CLRPEND (Interrupt 86)	Undefined		
21	CLRPEND (Interrupt 85)	Undefined		
20	CLRPEND (Interrupt 84)	Undefined		
19	CLRPEND (Interrupt 83)	Undefined		
18	CLRPEND (Interrupt 82)	Undefined		
17	CLRPEND (Interrupt 81)	Undefined		
16	CLRPEND (Interrupt 80)	Undefined		
15	CLRPEND (Interrupt 79)	Undefined		
14	CLRPEND (Interrupt 78)	Undefined		
13	CLRPEND (Interrupt 77)	Undefined		
12	CLRPEND (Interrupt 76)	Undefined		
11	CLRPEND (Interrupt 75)	Undefined		
10	CLRPEND (Interrupt 74)	Undefined		
9	CLRPEND (Interrupt 73)	Undefined		
8	CLRPEND (Interrupt 72)	Undefined		
7	CLRPEND (Interrupt 71)	Undefined		
6	CLRPEND (Interrupt 70)	Undefined		
5	CLRPEND (Interrupt 69)	Undefined		
4	CLRPEND (Interrupt 68)	Undefined		
3	CLRPEND (Interrupt 67)	Undefined		
2	CLRPEND (Interrupt 66)	Undefined		
1	CLRPEND (Interrupt 65)	Undefined		
0	CLRPEND (Interrupt 64)	Undefined		

(d) Interrupt Clear-Pending Register 3

Bit	Bit symbol	After reset	Type	Function
31	CLRPEND (Interrupt 127)	Undefined	R/W	[Write] 1: Clear pending interrupt. [Read] 0: Not pending 1: Pending
30	CLRPEND (Interrupt 126)	Undefined		
29	CLRPEND (Interrupt 125)	Undefined		
28	CLRPEND (Interrupt 124)	Undefined		
27	CLRPEND (Interrupt 123)	Undefined		
26	CLRPEND (Interrupt 122)	Undefined		
25	CLRPEND (Interrupt 121)	Undefined		
24	CLRPEND (Interrupt 120)	Undefined		
23	CLRPEND (Interrupt 119)	Undefined		
22	CLRPEND (Interrupt 118)	Undefined		
21	CLRPEND (Interrupt 117)	Undefined		
20	CLRPEND (Interrupt 116)	Undefined		
19	CLRPEND (Interrupt 115)	Undefined		
18	CLRPEND (Interrupt 114)	Undefined		
17	CLRPEND (Interrupt 113)	Undefined		
16	CLRPEND (Interrupt 112)	Undefined		
15	CLRPEND (Interrupt 111)	Undefined		
14	CLRPEND (Interrupt 110)	Undefined		
13	CLRPEND (Interrupt 109)	Undefined		
12	CLRPEND (Interrupt 108)	Undefined		
11	CLRPEND (Interrupt 107)	Undefined		
10	CLRPEND (Interrupt 106)	Undefined		
9	CLRPEND (Interrupt 105)	Undefined		
8	CLRPEND (Interrupt 104)	Undefined		
7	CLRPEND (Interrupt 103)	Undefined		
6	CLRPEND (Interrupt 102)	Undefined		
5	CLRPEND (Interrupt 101)	Undefined		
4	CLRPEND (Interrupt 100)	Undefined		
3	CLRPEND (Interrupt 99)	Undefined		
2	CLRPEND (Interrupt 98)	Undefined		
1	CLRPEND (Interrupt 97)	Undefined		
0	CLRPEND (Interrupt 96)	Undefined		

(e) Interrupt Clear-Pending Register 4

Bit	Bit symbol	After reset	Type	Function
31:30	-	0	R	Read as “0”.
29	CLRPEND (Interrupt 157)	Undefined	R/W	[Write] 1: Clear pending interrupt. [Read] 0: Not pending 1: Pending
28	CLRPEND (Interrupt 156)	Undefined		
27	CLRPEND (Interrupt 155)	Undefined		
26	CLRPEND (Interrupt 154)	Undefined		
25	CLRPEND (Interrupt 153)	Undefined		
24	CLRPEND (Interrupt 152)	Undefined		
23	CLRPEND (Interrupt 151)	Undefined		
22	CLRPEND (Interrupt 150)	Undefined		
21	CLRPEND (Interrupt 149)	Undefined		
20	CLRPEND (Interrupt 148)	Undefined		
19	CLRPEND (Interrupt 147)	Undefined		
18	CLRPEND (Interrupt 146)	Undefined		
17	CLRPEND (Interrupt 145)	Undefined		
16	CLRPEND (Interrupt 144)	Undefined		
15	CLRPEND (Interrupt 143)	Undefined		
14	CLRPEND (Interrupt 142)	Undefined		
13	CLRPEND (Interrupt 141)	Undefined		
12	CLRPEND (Interrupt 140)	Undefined		
11	CLRPEND (Interrupt 139)	Undefined		
10	CLRPEND (Interrupt 138)	Undefined		
9	CLRPEND (Interrupt 137)	Undefined		
8	CLRPEND (Interrupt 136)	Undefined		
7	CLRPEND (Interrupt 135)	Undefined		
6	CLRPEND (Interrupt 134)	Undefined		
5	CLRPEND (Interrupt 133)	Undefined		
4	CLRPEND (Interrupt 132)	Undefined		
3	CLRPEND (Interrupt 131)	Undefined		
2	CLRPEND (Interrupt 130)	Undefined		
1	CLRPEND (Interrupt 129)	Undefined		
0	CLRPEND (Interrupt 128)	Undefined		

5.2.5.6. Interrupt Priority Register

Each interrupt is provided with eight bits of an interrupt priority register.
The following shows the addresses of the interrupt priority registers corresponding to interrupt numbers.

Table 5.1 Interrupt Priority Register

Address	31 24	23 16	15 8	7 0
0xE000E400	PRI_3	PRI_2	PRI_1	PRI_0
0xE000E404	PRI_7	PRI_6	PRI_5	PRI_4
0xE000E408	PRI_11	PRI_10	PRI_9	PRI_8
0xE000E40C	PRI_15	PRI_14	PRI_13	PRI_12
0xE000E410	PRI_19	PRI_18	PRI_17	PRI_16
0xE000E414	PRI_23	PRI_22	PRI_21	PRI_20
0xE000E418	PRI_27	PRI_26	PRI_25	PRI_24
0xE000E41C	PRI_31	PRI_30	PRI_29	PRI_28
0xE000E420	PRI_35	PRI_34	-	PRI_32
0xE000E424	PRI_39	PRI_38	PRI_37	PRI_36
0xE000E428	PRI_43	PRI_42	PRI_41	PRI_40
0xE000E42C	PRI_47	PRI_46	PRI_45	PRI_44
0xE000E430	PRI_51	PRI_50	PRI_49	PRI_48
0xE000E434	PRI_55	PRI_54	PRI_53	PRI_52
0xE000E438	PRI_59	PRI_58	PRI_57	PRI_56
0xE000E43C	PRI_63	PRI_62	PRI_61	PRI_60
0xE000E440	PRI_67	PRI_66	PRI_65	PRI_64
0xE000E444	PRI_71	PRI_70	PRI_69	PRI_68
0xE000E448	PRI_75	PRI_74	PRI_73	PRI_72
0xE000E44C	PRI_79	PRI_78	PRI_77	PRI_76
0xE000E450	PRI_83	PRI_82	PRI_81	PRI_80
0xE000E454	PRI_87	PRI_86	PRI_85	PRI_84
0xE000E458	PRI_91	PRI_90	PRI_89	PRI_88
0xE000E45C	PRI_95	PRI_94	PRI_93	PRI_92
0xE000E460	PRI_99	PRI_98	PRI_97	PRI_96
0xE000E464	PRI_103	PRI_102	PRI_101	PRI_100
0xE000E468	PRI_107	PRI_106	PRI_105	PRI_104
0xE000E46C	PRI_111	PRI_110	PRI_109	PRI_108
0xE000E470	PRI_115	PRI_114	PRI_113	PRI_112
0xE000E474	PRI_119	PRI_118	PRI_117	PRI_116
0xE000E478	PRI_123	PRI_122	PRI_121	PRI_120
0xE000E47C	PRI_127	PRI_126	PRI_125	PRI_124
0xE000E480	PRI_131	PRI_130	PRI_129	PRI_128
0xE000E484	PRI_135	PRI_134	PRI_133	PRI_132
0xE000E488	PRI_139	PRI_138	PRI_137	PRI_136
0xE000E48C	PRI_143	PRI_142	PRI_141	PRI_140
0xE000E490	PRI_147	PRI_146	PRI_145	PRI_144
0xE000E494	PRI_151	PRI_150	PRI_149	PRI_148

Address	31 24	23 16	15 8	7 0
0xE000E498	PRI_155	PRI_154	PRI_153	PRI_152
0xE000E49C	-	-	PRI_157	PRI_156

The number of bits to be used for assigning a priority varies with each product. This product uses four bits for assigning a priority.

The following shows the fields of the interrupt priority registers for interrupt numbers 0 to 3.

Unused bits return "0" when read, and writing to unused bits has no effect.

Bit	Bit symbol	After reset	Type	Function
31:28	PRI_3[3:0]	0000	R/W	Priority of interrupt number 3
27:24	-	0	R	Read as "0".
23:20	PRI_2[3:0]	0000	R/W	Priority of interrupt number 2
19:16	-	0	R	Read as "0".
15:12	PRI_1[3:0]	0000	R/W	Priority of interrupt number 1
11:8	-	0	R	Read as "0".
7:4	PRI_0[3:0]	0000	R/W	Priority of interrupt number 0
3:0	-	0	R	Read as "0".

5.2.5.7. Vector Table Offset Register

Bit	Bit symbol	After reset	Type	Function
31:7	TBLOFF[24:0]	0x0000000	R/W	Offset value Set the offset value from the address of "0x00000000". The offset must be aligned based on the number of exceptions in the table. This means that the minimum alignment is 32 words that up to 16 interrupts are used. For more interrupts, the alignment must be adjusted by rounding up to the next power of two.
6:0	-	0	R	Read as "0".

5.2.5.8. Application Interrupt and Reset Control Register

Bit	Bit symbol	After reset	Type	Function
31:16	VECTKEY/ VECTKEYSTAT[15:0]	Undefined	W	Register key Writing to this register requires 0x05FA in the <VECTKEY> field.
			R	Register key Read as 0xFA05.
15	ENDIANESS	0	R/W	Endianness bit: (Note 1) 1: Big endian 0: Little endian
14:11	-	0	R	Read as "0".
10:8	PRIGROUP[2:0]	000	R/W	Interrupt priority grouping 000: seven bits of pre-emption priority, one bit of sub priority 001: six bits of pre-emption priority, two bits of sub priority 010: five bits of pre-emption priority, three bits of sub priority 011: four bits of pre-emption priority, four bits of sub priority 100: three bits of pre-emption priority, five bits of sub priority 101: two bits of pre-emption priority, six bits of sub priority 110: one bit of pre-emption priority, seven bits of sub priority 111: no pre-emption priority, eight bits of sub priority The bit configuration to split the interrupt priority register <PRI_n> into pre-emption priority and sub priority.
7:3	-	0	R	Read as "0".
2	SYSRESETREQ	0	R/W	System Reset Request 1: CPU outputs a SYSRESETREQ signal. (Note 2)
1	VECTCLRACTIVE	0	R/W	Clear active vector bit 1: clear all state information for active NMI, fault, and interrupts. 0: do not clear. This bit self-clears. It is the responsibility of the application to reinitialize the stack.
0	VECTRESET	0	R/W	System Reset 1: reset system. 0: do not reset system. Resets the internal circuits in CPU except debug components (FPB, DWT and ITM) by setting "1" and this bit is also zero cleared

Note1: Little-endian is selected as the default for this product.

Note2: When SYSRESETREQ is output, warm reset occurs. <SYSRESETREQ> is cleared by warm reset.

5.2.5.9. System Handler Priority Register

Each exception is provided with eight bits of a System Handler Priority Register.
The following shows the addresses of the System Handler Priority Registers corresponding to each exception.

Address	31 24	23 16	15 8	7 0
0xE000ED18	PRI_7	PRI_6 (Usage Fault)	PRI_5 (Bus Fault)	PRI_4 (Memory Management)
0xE000ED1C	PRI_11 (SVCall)	PRI_10	PRI_9	PRI_8
0xE000ED20	PRI_15 (SysTick)	PRI_14 (PendSV)	PRI_13	PRI_12 (Debug Monitor)

The number of bits to be used for assigning a priority varies with each product. This product uses four bits for assigning a priority.

The following shows the fields of the System Handler Priority Registers for Usage Fault, Bus Fault, and Memory Management. Unused bits return "0" when read, and writing to unused bits has no effect.

Bit	Bit symbol	After reset	Type	Function
31:28	PRI_7[3:0]	0000	R/W	Reserved
27:24	-	0	R	Read as "0".
23:20	PRI_6[3:0]	0000	R/W	Priority of Usage Fault
19:16	-	0	R	Read as "0".
15:12	PRI_5[3:0]	0000	R/W	Priority of Bus Fault
11:8	-	0	R	Read as "0".
7:4	PRI_4[3:0]	0000	R/W	Priority of Memory Management
3:0	-	0	R	Read as "0".

5.2.5.10. System Handler Control and State Register

Bit	Bit symbol	After reset	Type	Function
31:19	-	0	R	Read as "0".
18	USGFAULTENA	0	R/W	Usage Fault 0: Disabled 1: Enabled
17	BUSFAULTENA	0	R/W	Bus Fault 0: Disabled 1: Enabled
16	MEMFAULTENA	0	R/W	Memory Management 0: Disabled 1: Enabled
15	SVCALLPENDED	0	R/W	SVCall 0: Not pended 1: Pended
14	BUSFAULTPENDED	0	R/W	Bus Fault 0: Not pended 1: Pended
13	MEMFAULTPENDED	0	R/W	Memory Management 0: Not pended 1: Pended
12	USGFAULTPENDED	0	R/W	Usage Fault 0: Not pended 1: Pended
11	SYSTICKACT	0	R/W	SysTick 0: Inactive 1: Active
10	PENDSVACT	0	R/W	PendSV 0: Inactive 1: Active
9	-	0	R	Read as "0".
8	MONITORACT	0	R/W	Debug Monitor 0: Inactive 1: Active
7	SVCALLACT	0	R/W	SVCall 0: Inactive 1: Active
6:4	-	0	R	Read as "0".
3	USGFAULTACT	0	R/W	Usage Fault 0: Inactive 1: Active
2	-	0	R	Read as "0".
1	BUSFAULTACT	0	R/W	Bus Fault 0: Inactive 1: Active
0	MEMFAULTACT	0	R/W	Memory Management 0: Inactive 1: Active

Note: Clearing or setting the active bits is required extreme caution because clearing and setting these bits does not modify stack contents.

6. Revision History

Table 6.1 Revision History

Revision	Date	Description
1.0	2026-01-30	- First release

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPT ION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**