32-Bit RISC Microcontroller

TX Family

TMPM471F10FG

Reference Manual

Exception

(EXCEPT-TMPM471F10FG)

Revision 1.0

2024-08

Toshiba Electronic Devices & Storage Corporation

© 2024
Toshiba Electronic Devices & Storage Corporation

Contents

Preface 5
Related document 5
Conventions 6
Terms and Abbreviations 8
1. Outlines 9
1.1. Exception Types 9
1.2. Exception Handling Flowchart 10
1.2.1. Exception Request and Detection 11
1.2.2. Exception Handling and Branch to Interrupt Service Routine (Pre-emption) 13
1.2.3. Executing ISR 14
1.2.4. Exception Exit 15
2. Reset Exception 16
3. SysTick 17
4. Interrupts 17
4.1. Non-Maskable Interrupt (NMI) 17
4.2. Maskable Interrupt 17
4.3. Interrupt Request 18
4.3.1. Interrupt Route 18
4.3.2. Interrupt Request Generation 21
4.3.3. Monitor of Interrupt Request 21
4.3.4. Transmission of Interrupt Request 21
4.3.5. Precautions When Using External Interrupt Pins 21
4.4. List of Interrupt Factors 22
4.4.1. Joint Interrupt 25
4.5. Interrupt Detection Level 28
4.5.1. Precautions When Releasing Low-power Consumption Mode 28
4.6. Interrupt Handling 29
4.6.1. Flowchart of Interrupt Handling 29
4.6.2. Preparation 30
4.6.3. Detection (INTIF) 32
4.6.4. Detection (CPU) 32
4.6.5. CPU Processing 32
4.6.6. Processing in Interrupt Service Routine (Clearing Interrupt Factor) 33
5. Exception/Interrupt-related Registers 34
5.1. Register List 34
5.2. Interrupt Control Register A 38
5.2.1. *[IANIC00]* (Non-Maskable Interrupt A Control Register 00) 38
5.3. Interrupt Control Registers B 38
5.3.1. *[IBNIC00]* (Non-Maskable Interrupt B Control Register 00) 38
5.3.2. *[IBIMC000]* to *[IBIMC095]* Interrupt B Mode Control Registers) 39
5.4. Reset Flag Registers 40

5.4.1. *[RLMRSTFLG0]* (Reset Flag Register 0) 40
5.4.2. *[RLMRSTFLG1]* (Reset Flag Register 1) 41
5.5. Interrupt Monitor Registers 42
5.5.1. *[IMNFLGNMI]* (Non-Maskable Interrupt Monitor Flag Register) 42
5.5.2. *[IMNFLG3]* (Interrupt Monitor Flag Register 3) 43
5.5.3. *[IMNFLG4]* (Interrupt Monitor Flag Register 4) 45
5.6. NVIC Registers 46
5.6.1. SysTick Control and Status Register 46
5.6.2. SysTick Reload Value Register 46
5.6.3. SysTick Current Value Register 46
5.6.4. SysTick Calibration Value Register 47
5.6.5. Interrupt Control Registers 48
5.6.6. Interrupt Priority Register 60
5.6.7. Vector Table Offset Register 61
5.6.8. Application Interrupt and Reset Control Register 62
5.6.9. System Handler Priority Register 63
5.6.10. System Handler Control and Status Register 64
6. Revision History 65
RESTRICTIONS ON PRODUCT USE 66

List of Figures

Figure 4.1 Interrupt Transfer Route Diagram 19

List of Tables

Table 1.1 Exception Types and Priority 11
Table 1.2 Priority Grouping Setting 12
Table 4.1 Explanation of Each Interrupt Transfer Route 20
Table 4.2 List of Interrupt Factors (Non-Maskable Interrupt) 22
Table 4.3 List of Interrupt Factors (Interrupt Control Register B) (1/3) 22
Table 4.4 List of Interrupt Factors (Interrupt Control Register B) (2/3) 23
Table 4.5 List of Interrupt Factors (Interrupt Control Register B) (3/3) 24
Table 4.6 Joint Interrupt List (1/3) 25
Table 4.7 Joint Interrupt List (2/3) 26
Table 4.8 Joint Interrupt List (3/3) 27
Table 6.1 Revision History 65

Preface

Related document

Document name
Oscillation Frequency Detector
Clock Selective Watchdog Timer
Voltage Detection Circuit
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.

****
Arm, Cortex and Thumb are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved.
****

All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

Terms and Abbreviations

Some of abbreviations used in this document are as follows:

ADC	Analog to Digital Converter
A-ENC32	Advanced Encoder Input Circuit (32-bit)
A-PMD	Advanced Programmable Motor Control Circuit
DMAC	Direct Memory Access Controller
EI2C	I^2C Interface Version A
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
INTIF	Interrupt Interface Logic
ISR	Interrupt Service Routine
I^2C	Inter-Integrated Circuit
LVD	Voltage Detection Circuit
NICxx	Non-Maskable Interrupt Control xx
OFD	Oscillation Frequency Detector
POR	Power On Reset Circuit
PORF	Power On Reset Circuit for FLASH and Debug
RLMRSTFLGx	RLM Reset Flag x
SIWDT	Clock Selective Watchdog Timer
TSPI	Serial Peripheral Interface
T32A	32-bit Timer Event Counter
UART	Universal Asynchronous Receiver Transmitter

Exceptions have close relation to the CPU core. Refer to "Arm Cortex-M4 Processor Technical Reference Manual" if needed.

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

The following types of exceptions exist in this product.
For detailed descriptions of 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	Refer to
Detection by INTIF/CPU	The INTIF/CPU detects the exception request.	Section 1.2.1
⇩		
Handling by CPU	The CPU handles the exception request.	Section 1.2.2
⇩		
Branch to ISR	The CPU branches to the corresponding interrupt service routine.	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) Exception occurrence

Exception factors 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 terminal 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 "4. Interrupts"

(2) Exception detection

If multiple exceptions occur simultaneously, the CPU takes the exception with the highest priority.
Table 1.1 shows the priority of exceptions. "Configurable" means that you can assign a priority level to that exception. Memory Management, Bus Fault and Usage Fault exceptions can be enabled or disabled. If a disabled exception occurs, it is handled as Hard Fault.

Table 1.1 Exception Types and Priority

Exception type	Priority	Description	Offset
Reset	-3 (highest)	Reset pin, POR , PORF, OFD, SIWDT, LVD, SYSRESETREQ, LOCKUP signal	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 interrupts have different factors and numbers in each product. For details, refer to "4.4. List of Interrupt Factors".

(3) Priority setting

- Priority Level

The external interrupt priority 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 varies from 3 bits to 8 bits depending on products. Thus, the range of priority values you can specify is different depending on products.
In the case of 8-bit configuration, the priority can be configured in the range from 0 to 255. The highest priority is "0". If multiple elements with the same priority exist, the smaller the number, the higher the priority becomes.

<PRI_n[7:0]> bit is defined as the upper 4-bit configuration with TMPM471F10FG. The priority can be configured in the range from 0 to 15.

- Priority Grouping

The priority group can be split into groups. By setting the <PRIGROUP> of the Application Interrupt and Reset Control Register, <PRI_n> can be divided into the pre-emption priority and the sub priority.
A priority is compared with the pre-emption priority. If the priority is the same as the pre-emption priority, then it is compared with the sub priority. If the sub priority is the same as the priority, the smaller the exception number, the higher the priority.
Table 1.2 shows the priority group setting. The pre-emption priority and the sub priority in the table are the number 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	-	[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)

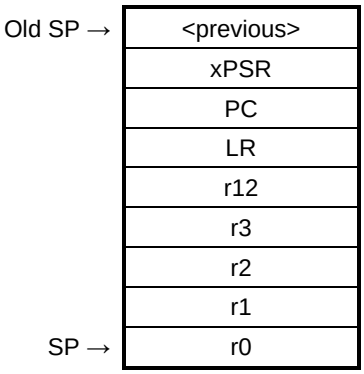
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

(1) 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 by the completion of the stack push. The following shows the state of the stack after the register contents have been pushed.



(2) Fetching ISR

The CPU performs the evacuation of the register. In addition, the CPU performs instruction fetch of the interrupt service routine at the same time.

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. By setting the Vector Table Offset Register, you can place the vector table at any address in the Code or SRAM space.
The vector table should also contain the initial value of the main stack.

(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 a new vector address for branching to the corresponding ISR, but the CPU does not newly push the register contents to the stack.

(4) Vector table

The vector table is configured as shown below.
You must always set the first four words (stack top address, reset ISR address, NMI ISR address, and Hard Fault ISR address). Set ISR addresses for other exceptions if necessary.

For other exceptions, you should prepare the ISR addresses 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. Executing ISR

An ISR performs necessary processing for the corresponding 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 abandons the currently executing ISR and services the newly detected exception.

1.2.4. Exception Exit

(1) Execution after returning from ISR

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

(a) 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".

(b) Returning to last stacked ISR

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

(c) Returning to the previous program

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

(2) Exception Exit Sequence

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

(a) Pop 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 stacked xPSR. The CPU uses this to track 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 is SP_main or SP_process.

2. Reset Exception

Reset exceptions are generated from the following factors.
Use reset flag register ***[RLMRSTFLGn]*** to identify the factor of a reset.

- Reset exception by external reset pin
  A reset exception occurs when an external reset pin changes from "Low" to "High".

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

- Reset exception by OFD
  The OFD has a reset generating feature. For details, refer to the reference manual "Oscillation Frequency Detector".

- Reset exception by SIWDT
  The SIWDT has a reset generating feature. For details, refer to the reference manual "Clock Selective Watchdog Timer".

- Reset exception by LVD
  The LVD has a reset generating feature. For details, refer to the reference manual "Voltage Detector Circuit".

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

- Reset exception by <SYSRESETREQ>
  A reset can be generated by setting <SYSRESETREQ> in the NVIC's Application Interrupt and Reset Control Register.

- Reset exception by LOCKUP signal
  A reset can be generated by the LOCKUP signal which can be output from the Cortex-M4 with FPU when the un-recoverable exception occurs. For details on the LOCKUP signal, please refer to "Arm Cortex-M4 Processor Technical Reference Manual".

3. SysTick

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

When setting a value in the SysTick Reload Value Register and enable the SysTick features by the SysTick Control and Status Register, the counter loads with the value set in the Reload Value Register and begins counting down. When the counter reaches "0", a SysTick exception occurs. An exception can be pended and it can be informed that the timer reaches "0".

4. Interrupts

This chapter explains the route from which a factor and an interrupt request are transmitted, and a required setup.

4.1. Non-Maskable Interrupt (NMI)

Non-Maskable interrupts are generated from the following factors.

- Non-Maskable interrupt by SIWDT
  The SIWDT has a non-maskable interrupt generating feature.
  For details of the SIWDT, refer to the reference manual "Clock Selective Watchdog Timer".

- Non-Maskable interrupt by LVD
  The LVD has a non-maskable interrupt generating feature.
  For details of the LVD, refer to the reference manual "Voltage Detector Circuit".

4.2. Maskable Interrupt

Refer to the interrupt control register A and B of the "4.4. List of Interrupt Factors" for the factor of the maskable interrupts.

4.3. Interrupt Request

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

4.3.1. Interrupt Route

The interrupt is available for the release from a low-power consumption mode, and a route varies according to a factor.

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

- The interrupts that can release IDLE, STOP1 mode

  They have two interrupt routes via INTIF that can release IDLE and STOP1 mode.

  (a) They are controlled by interrupt control register A in INTIF and notified to CPU. (Route A, B, and C)
  (b) They are controlled by interrupt control register B in INTIF and notified to CPU. (Route D, E, and F)

- The interrupt that can release IDLE mode

  Some factors of interrupt which can release IDLE mode are controlled by interrupt control register B in INTIF and notified to CPU (Route G). Other factors are notified to CPU directly not passing through INTIF (Route H).

When the interrupt factor via INTIF regardless of low-power consumption mode release is used, interrupt control register A or B must be set for it.

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

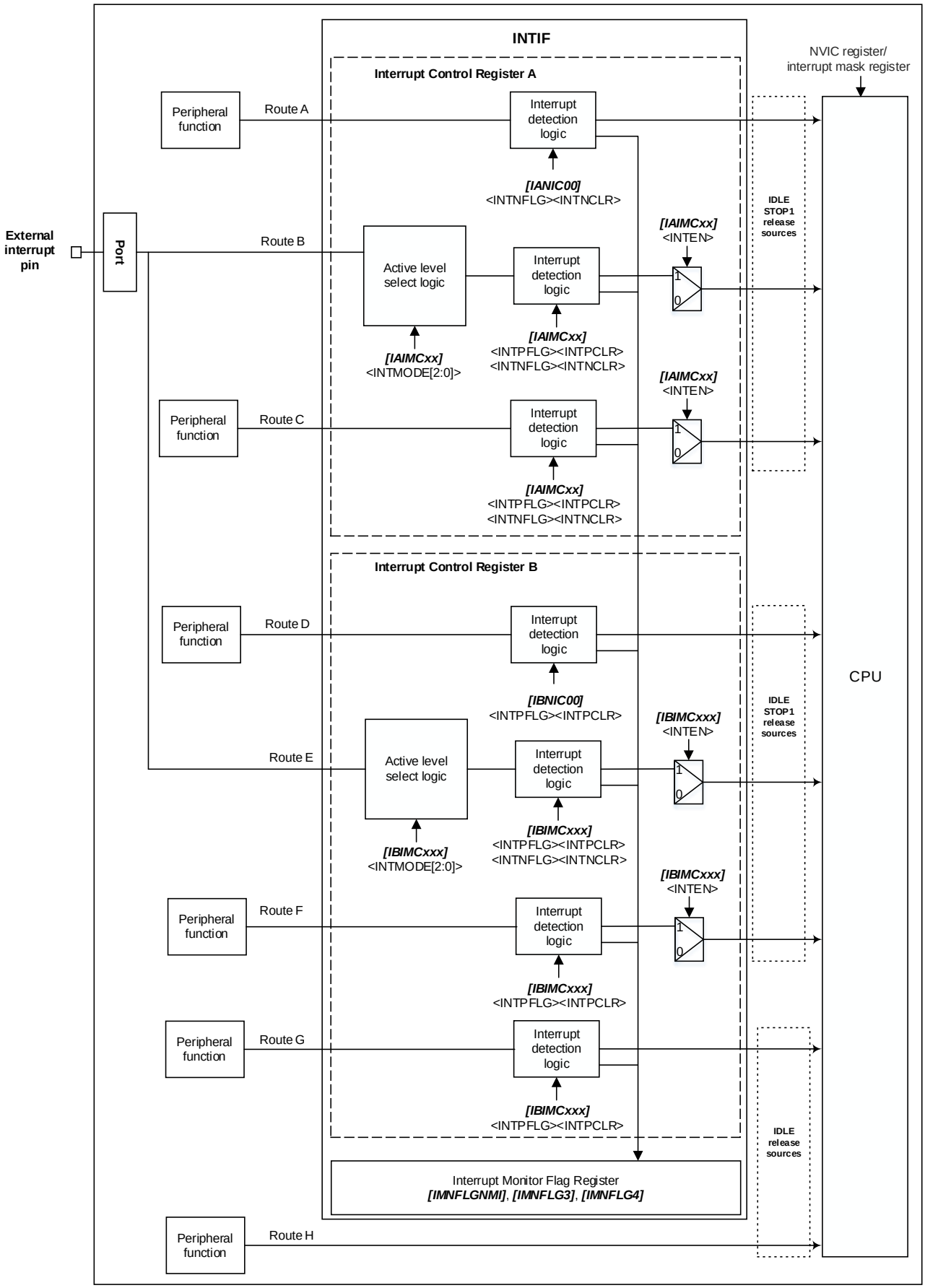


Figure 4.1 Interrupt Transfer Route Diagram

Table 4.1 Explanation of Each Interrupt Transfer Route

Route	Interrupt No.	Interrupt request	Description of route
A	-	LVD interrupt	This route is non-maskable interrupt. It is a route input into CPU via INTIF. An interrupt release setup is carried out by the interrupt control register A (***[IANIC00]***).
B	-	-	The interrupt request of a port is a route input into CPU via INTIF. Selection of an Interrupt detection level, interrupt release, and interrupt request enable/disable are set up by the interrupt control register A (***[IAIMCxx]***) for every interrupt request.
C	-	-	It is a route input into CPU via INTIF. Selection of an Interrupt detection level, interrupt release, and interrupt request enable/disable are set up by the interrupt control register A (***[IAIMCxx]***) for every interrupt request.
D	-	SIWDT interrupt	This route is non-maskable interrupt. It is a route input into CPU via INTIF. An interrupt release setup is carried out by the interrupt control register B (***[IBNIC00]***).
E	0 to 15	External interrupts (0 to F)	The interrupt request of a port is a route input into CPU via INTIF. Selection of an Interrupt detection level, interrupt release, and interrupt request enable/disable are set up by the interrupt control register B (***[IBIMCxxx]***) for every interrupt request.
F	-	-	It is a route input into CPU via INTIF. Interrupt request enable/disable is set up by the interrupt control register B (***[IBIMCxxx]***) for every interrupt request.
G	93, 94	DMAC transfer completion interrupt (ch0 to 31), DMAC transfer error interrupt (Note)	It is a route input into CPU via INTIF. Interrupt release is set up by the interrupt control register B (***[IBIMCxxx]***) for every interrupt request.
H	16 to 92, 95	Other interrupts	It is a route as which an interrupt request is directly input into CPU not passing through INTIF.

Note: The DMAC transfer completion interrupt is the interrupt into which interrupts of two or more channels are combined.

4.3.2. Interrupt Request Generation

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

- Interrupt from external interrupt pin
  Set the port control register so that the external pin can perform as an interrupt function pin.

- Interrupt from peripheral function
  Set the peripheral function to make it possible to output interrupt requests.
  Refer to the reference manual of each peripheral function for details.

- Generation of interrupt request forcibly
  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

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

For detail, refer to "4.4. List of Interrupt Factors".

4.3.4. Transmission of Interrupt Request

An interrupt request which is not passing through the interrupt control register is directly connected to the CPU. The corresponding interrupt control register of INTIF for the interrupts connected to the CPU through INTIF, which can be also used as interrupt requests for releasing the low-power consumption mode, is required the proper setting. A "High" level interrupt signal is transmitted to the CPU, when the interrupt is used to release the low-power consumption mode.

Set an interrupt detection level and interrupt detection enable/disable in INTIF.

Furthermore, please be cautious about external interrupt pin as in the next section.

4.3.5. Precautions When Using External Interrupt Pins

When using external interrupt, please care about the following points so that an unexpected interrupt does not occur.

If input for port is disabled (***[PxIE]***<PxmIE> = 0), input signal from an external interrupt pin is "Low". When the <INTMODE> of Interrupt Control Register B (***[IBIMCxxx]***) is "000" (Detection level is Low level.), input signal from an external interrupt pin is transmitted to the CPU as low-level signal. Therefore, the CPU recognizes it as an interrupt request from an external interrupt pin. If the corresponding interrupt detection is changed to enable in this state, an It's interrupt occurs.

First, the interrupt pin should be "High" and the input for port should be enabled. Then, the interrupt detection should be enabled by the CPU.

4.4. List of Interrupt Factors

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

Table 4.2 List of Interrupt Factors (Non-Maskable Interrupt)

Interrupt factor	Interrupt request	Interrupt control register	Interrupt monitor register
INTLVD	LVD interrupt	***[IANIC00]***	***[IMNFLGNMI]*** <INT000FLG>
INTWDT0	SIWDT interrupt	***[IBNIC00]***	***[IMNFLGNMI]*** <INT016FLG>

There are no interrupt factors of the interrupt control register A in TMPM471F10FG.

All maskable interrupt factors are in the interrupt control register B.

Some interrupts can be used as factors for releasing low-power consumption mode. The interrupt control register B performs setting for detecting the release of the low-power consumption mode, and interrupt detection enable/disable.

Table 4.3 List of Interrupt Factors (Interrupt Control Register B) (1/3)

Interrupt No.	Interrupt factor	Interrupt request	Interrupt control register	Interrupt monitor register
0	INT0	External interrupt 0	***[IBIMC000]***	***[IMNFLG3]*** <INT096FLG>
1	INT1	External interrupt 1	***[IBIMC001]***	***[IMNFLG3]*** <INT097FLG>
2	INT2	External interrupt 2	***[IBIMC002]***	***[IMNFLG3]*** <INT098FLG>
3	INT3	External interrupt 3	***[IBIMC003]***	***[IMNFLG3]*** <INT099FLG>
4	INT4	External interrupt 4	***[IBIMC004]***	***[IMNFLG3]*** <INT100FLG>
5	INT5	External interrupt 5	***[IBIMC005]***	***[IMNFLG3]*** <INT101FLG>
6	INT6	External interrupt 6	***[IBIMC006]***	***[IMNFLG3]*** <INT102FLG>
7	INT7	External interrupt 7	***[IBIMC007]***	***[IMNFLG3]*** <INT103FLG>
8	INT8	External interrupt 8	***[IBIMC008]***	***[IMNFLG3]*** <INT104FLG>
9	INT9	External interrupt 9	***[IBIMC009]***	***[IMNFLG3]*** <INT105FLG>
10	INTA	External interrupt A	***[IBIMC010]***	***[IMNFLG3]*** <INT106FLG>
11	INTB	External interrupt B	***[IBIMC011]***	***[IMNFLG3]*** <INT107FLG>
12	INTC	External interrupt C	***[IBIMC012]***	***[IMNFLG3]*** <INT108FLG>
13	INTD	External interrupt D	***[IBIMC013]***	***[IMNFLG3]*** <INT109FLG>
14	INTE	External interrupt E	***[IBIMC014]***	***[IMNFLG3]*** <INT110FLG>
15	INTF	External interrupt F	***[IBIMC015]***	***[IMNFLG3]*** <INT111FLG>

Table 4.4 List of Interrupt Factors (Interrupt Control Register B) (2/3)

Interrupt No.	Interrupt factor	Interrupt request	Interrupt control register	Interrupt monitor register
16	INTEMG0	A-PMD ch0 EMG Interrupt		
17	INTEMG1	A-PMD ch1 EMG Interrupt		
18	INTOVV0	A-PMD ch0 OVV Interrupt		
19	INTOVV1	A-PMD ch1 OVV Interrupt		
20	INTPWM0	A-PMD ch0 PWM Interrupt		
21	INTPWM1	A-PMD ch1 PWM Interrupt		
22	INTENC00	A-ENC32 ch0 encoder input interrupt 0		
23	INTENC01	A-ENC32 ch0 encoder input interrupt 1		
24	INTENC10	A-ENC32 ch1 encoder input interrupt 0		
25	INTENC11	A-ENC32 ch1 encoder input interrupt 1		
26	INTADAPDA	ADC unit A PMD trigger program conversion complete A		
27	INTADAPDB	ADC unit A PMD trigger program conversion complete B		
28	INTADACP0	ADC unit A monitor function 0 interrupt		
29	INTADACP1	ADC unit A monitor function 1 interrupt		
30	INTADATRG	ADC unit A general trigger program conversion complete		
31	INTADASGL	ADC unit A single program conversion completion		
32	INTADACNT	ADC unit A continuous program conversion complete		
33	INTADBPDA	ADC unit B PMD trigger program conversion complete A		
34	INTADBPDB	ADC unit B PMD trigger program conversion complete B		
35	INTADBCP0	ADC unit B monitor function 0 interrupt		
36	INTADBCP1	ADC unit B monitor function 1 interrupt		
37	INTADBTRG	ADC unit B general trigger program conversion complete		
38	INTADBSGL	ADC unit B single program conversion completion		
39	INTADBCNT	ADC unit B continuous program conversion complete		
40	INTSC0RX (Note)	TSPI ch0 reception interrupt/UART ch0 reception interrupt		
41	INTSC0TX (Note)	TSPI ch0 transmission interrupt/UART ch0 transmission interrupt		
42	INTSC0ERR (Note)	TSPI ch0 error interrupt/UART ch0 error interrupt		
43	INTSC1RX (Note)	TSPI ch1 reception interrupt/UART ch1 reception interrupt		
44	INTSC1TX (Note)	TSPI ch1 transmission interrupt/UART ch1 transmission interrupt		
45	INTSC1ERR (Note)	TSPI ch1 error interrupt/UART ch1 error interrupt		
46	INTSC2RX (Note)	TSPI ch2 reception interrupt/UART ch2 reception interrupt		
47	INTSC2TX (Note)	TSPI ch2 transmission interrupt/UART ch2 transmission interrupt		
48	INTSC2ERR (Note)	TSPI ch2 error interrupt/UART ch2 error interrupt		
49	INTSC3RX (Note)	TSPI ch3 reception interrupt/UART ch3 reception interrupt		
50	INTSC3TX (Note)	TSPI ch3 transmission interrupt/UART ch3 transmission interrupt		
51	INTSC3ERR (Note)	TSPI ch3 error interrupt/UART ch3 error interrupt		
52	INTUART4RX	UART ch4 reception interrupt		
53	INTUART4TX	UART ch4 transmission interrupt		
54	INTUART4ERR	UART ch4 error interrupt		
55	INTI2C0ST	EI2C ch0 status interrupt		
56	INTI2C0TBE	EI2C ch0 transmit buffer empty interrupt		

Note: Refer to "4.4.1. Joint Interrupt".

Table 4.5 List of Interrupt Factors (Interrupt Control Register B) (3/3)

Interrupt No.	Interrupt factor	Interrupt request	Interrupt control register	Interrupt monitor register
57	INTI2C0RBF	EI2C ch0 receive buffer full interrupt		
58	INTI2C1ST	EI2C ch1 status interrupt		
59	INTI2C1TBE	EI2C ch1 transmit buffer empty interrupt		
60	INTI2C1RBF	EI2C ch1 receive buffer full interrupt		
61	INTT32A00AC (Note)	T32A ch0 timer A/C match, overflow, and underflow		
62	INTT32A00ACCAP0 (Note)	T32A ch0 timer A/C capture 0		
63	INTT32A00ACCAP1 (Note)	T32A ch0 timer A/C capture 1		
64	INTT32A00B	T32A ch0 timer B match, overflow, and underflow		
65	INTT32A00BCAP0	T32A ch0 timer B capture 0		
66	INTT32A00BCAP1	T32A ch0 timer B capture 1		
67	INTT32A01AC (Note)	T32A ch1 timer A/C match, overflow, and underflow		
68	INTT32A01ACCAP0 (Note)	T32A ch1 timer A/C capture 0		
69	INTT32A01ACCAP1 (Note)	T32A ch1 timer A/C capture 1		
70	INTT32A01B	T32A ch1 timer B match, overflow, and underflow		
71	INTT32A01BCAP0	T32A ch1 timer B capture 0		
72	INTT32A01BCAP1	T32A ch1 timer B capture 1		
73	INTT32A02AC (Note)	T32A ch2 timer A/C match, overflow, and underflow		
74	INTT32A02ACCAP0 (Note)	T32A ch2 timer A/C capture 0		
75	INTT32A02ACCAP1 (Note)	T32A ch2 timer A/C capture 1		
76	INTT32A02B	T32A ch2 timer B match, overflow, and underflow		
77	INTT32A02BCAP0	T32A ch2 timer B capture 0		
78	INTT32A02BCAP1	T32A ch2 timer B capture 1		
79	INTT32A03AC (Note)	T32A ch3 timer A/C match, overflow, and underflow		
80	INTT32A03ACCAP0 (Note)	T32A ch3 timer A/C capture 0		
81	INTT32A03ACCAP1 (Note)	T32A ch3 timer A/C capture 1		
82	INTT32A03B	T32A ch3 timer B match, overflow, and underflow		
83	INTT32A03BCAP0	T32A ch3 timer B capture 0		
84	INTT32A03BCAP1	T32A ch3 timer B capture 1		
85	INTT32A04AC (Note)	T32A ch4 timer A/C match, overflow, and underflow		
86	INTT32A04ACCAP0 (Note)	T32A ch4 timer A/C capture 0		
87	INTT32A04ACCAP1 (Note)	T32A ch4 timer A/C capture 1		
88	INTT32A04B	T32A ch4 timer B match, overflow, and underflow		
89	INTT32A04BCAP0	T32A ch4 timer B capture 0		
90	INTT32A04BCAP1	T32A ch4 timer B capture 1		
91	INTPARI0	RAM Parity interrupt 0		
92	INTPARI1	RAM Parity interrupt 1		
93	INTDMAATC	DMAC transfer completion interrupt (ch0 to 31)	***[IBIMC016]*** to ***[IBIMC047]***	***[IMNFLG3]*** <INT112FLG> to <INT127FLG> ***[IMNFLG4]*** <INT128FLG> to <INT143FLG>
94	INTDMAAERR	DMAC transfer error interrupt	***[IBIMC048]***	***[IMNFLG4]*** <144FLG>
95	INTFLCRDY	Code FLASH Ready interrupt		

Note: Refer to "4.4.1. Joint Interrupt".

4.4.1. Joint Interrupt

Details of the joint interrupts in TMPM471F10FG are as follows.

Table 4.6 Joint Interrupt List (1/3)

Interrupt No.	Joint interrupt factor	Interrupt factor	Interrupt request	Interrupt control register	Interrupt monitor register
40	INTSC0RX	INTT0RX	TSPI ch0 reception interrupt		
		INTUART0RX	UART ch0 reception interrupt		
41	INTSC0TX	INTT0TX	TSPI ch0 transmission interrupt		
		INTUART0TX	UART ch0 transmission interrupt		
42	INTSC0ERR	INTT0ERR	TSPI ch0 error interrupt		
		INTUART0ERR	UART ch0 error interrupt		
43	INTSC1RX	INTT1RX	TSPI ch1 reception interrupt		
		INTUART1RX	UART ch1 reception interrupt		
44	INTSC1TX	INTT1TX	TSPI ch1 transmission interrupt		
		INTUART1TX	UART ch1 transmission interrupt		
45	INTSC1ERR	INTT1ERR	TSPI ch1 error interrupt		
		INTUART1ERR	UART ch1 error interrupt		
46	INTSC2RX	INTT2RX	TSPI ch2 reception interrupt		
		INTUART2RX	UART ch2 reception interrupt		
47	INTSC2TX	INTT2TX	TSPI ch2 transmission interrupt		
		INTUART2TX	UART ch2 transmission interrupt		
48	INTSC2ERR	INTT2ERR	TSPI ch2 error interrupt		
		INTUART2ERR	UART ch2 error interrupt		
49	INTSC3RX	INTT3RX	TSPI ch3 reception interrupt		
		INTUART3RX	UART ch3 reception interrupt		
50	INTSC3TX	INTT3TX	TSPI ch3 transmission interrupt		
		INTUART3TX	UART ch3 transmission interrupt		
51	INTSC3ERR	INTT3ERR	TSPI ch3 error interrupt		
		INTUART3ERR	UART ch3 error interrupt		

Note: Set the IPs so that only one interrupt of them is generated.

Table 4.7 Joint Interrupt List (2/3)

Interrupt No.	Joint interrupt factor	Interrupt factor	Interrupt request	Interrupt control register	Interrupt monitor register
61	INTT32A00AC	INTT32A00A	T32A ch0 timer A match, overflow, and underflow		
		INTT32A00C	T32A ch0 timer C match, overflow, and underflow		
62	INTT32A00ACCAP0	INTT32A00ACAP0	T32A ch0 timer A capture 0		
		INTT32A00CCAP0	T32A ch0 timer C capture 0		
63	INTT32A00ACCAP1	INTT32A00ACAP1	T32A ch0 timer A capture 1		
		INTT32A00CCAP1	T32A ch0 timer C capture 1		
67	INTT32A01AC	INTT32A01A	T32A ch1 timer A match, overflow, and underflow		
		INTT32A01C	T32A ch1 timer C match, overflow, and underflow		
68	INTT32A01ACCAP0	INTT32A01ACAP0	T32A ch1 timer A capture 0		
		INTT32A01CCAP0	T32A ch1 timer C capture 0		
69	INTT32A01ACCAP1	INTT32A01ACAP1	T32A ch1 timer A capture 1		
		INTT32A01CCAP1	T32A ch1 timer C capture 1		
73	INTT32A02AC	INTT32A02A	T32A ch2 timer A match, overflow, and underflow		
		INTT32A02C	T32A ch2 timer C match, overflow, and underflow		
74	INTT32A02ACCAP0	INTT32A02ACAP0	T32A ch2 timer A capture 0		
		INTT32A02CCAP0	T32A ch2 timer C capture 0		
75	INTT32A02ACCAP1	INTT32A02ACAP1	T32A ch2 timer A capture 1		
		INTT32A02CCAP1	T32A ch2 timer C capture 1		
79	INTT32A03AC	INTT32A03A	T32A ch3 timer A match, overflow, and underflow		
		INTT32A03C	T32A ch3 timer C match, overflow, and underflow		
80	INTT32A03ACCAP0	INTT32A03ACAP0	T32A ch3 timer A capture 0		
		INTT32A03CCAP0	T32A ch3 timer C capture 0		
81	INTT32A03ACCAP1	INTT32A03ACAP1	T32A ch3 timer A capture 1		
		INTT32A03CCAP1	T32A ch3 timer C capture 1		
85	INTT32A04AC	INTT32A04A	T32A ch4 timer A match, overflow, and underflow		
		INTT32A04C	T32A ch4 timer C match, overflow, and underflow		
86	INTT32A04ACCAP0	INTT32A04ACAP0	T32A ch4 timer A capture 0		
		INTT32A04CCAP0	T32A ch4 timer C capture 0		
87	INTT32A04ACCAP1	INTT32A04ACAP1	T32A ch4 timer A capture 1		
		INTT32A04CCAP1	T32A ch4 timer C capture 1		

Note: Set the IPs so that only one interrupt of them is generated.

Table 4.8 Joint Interrupt List (3/3)

Interrupt No.	Interrupt request		Interrupt control register	Interrupt monitor register
93	DMAC transfer completion interrupt (INTDMAATC)	ch0	***[IBIMC016]***	***[IMNFLG3]***<INT112FLG>
		ch1	***[IBIMC017]***	***[IMNFLG3]***<INT113FLG>
		ch2	***[IBIMC018]***	***[IMNFLG3]***<INT114FLG>
		ch3	***[IBIMC019]***	***[IMNFLG3]***<INT115FLG>
		ch4	***[IBIMC020]***	***[IMNFLG3]***<INT116FLG>
		ch5	***[IBIMC021]***	***[IMNFLG3]***<INT117FLG>
		ch6	***[IBIMC022]***	***[IMNFLG3]***<INT118FLG>
		ch7	***[IBIMC023]***	***[IMNFLG3]***<INT119FLG>
		ch8	***[IBIMC024]***	***[IMNFLG3]***<INT120FLG>
		ch9	***[IBIMC025]***	***[IMNFLG3]***<INT121FLG>
		ch10	***[IBIMC026]***	***[IMNFLG3]***<INT122FLG>
		ch11	***[IBIMC027]***	***[IMNFLG3]***<INT123FLG>
		ch12	***[IBIMC028]***	***[IMNFLG3]***<INT124FLG>
		ch13	***[IBIMC029]***	***[IMNFLG3]***<INT125FLG>
		ch14	***[IBIMC030]***	***[IMNFLG3]***<INT126FLG>
		ch15	***[IBIMC031]***	***[IMNFLG3]***<INT127FLG>
		ch16	***[IBIMC032]***	***[IMNFLG4]***<INT128FLG>
		ch17	***[IBIMC033]***	***[IMNFLG4]***<INT129FLG>
		ch18	***[IBIMC034]***	***[IMNFLG4]***<INT130FLG>
		ch19	***[IBIMC035]***	***[IMNFLG4]***<INT131FLG>
		ch20	***[IBIMC036]***	***[IMNFLG4]***<INT132FLG>
		ch21	***[IBIMC037]***	***[IMNFLG4]***<INT133FLG>
		ch22	***[IBIMC038]***	***[IMNFLG4]***<INT134FLG>
		ch23	***[IBIMC039]***	***[IMNFLG4]***<INT135FLG>
		ch24	***[IBIMC040]***	***[IMNFLG4]***<INT136FLG>
		ch25	***[IBIMC041]***	***[IMNFLG4]***<INT137FLG>
		ch26	***[IBIMC042]***	***[IMNFLG4]***<INT138FLG>
		ch27	***[IBIMC043]***	***[IMNFLG4]***<INT139FLG>
		ch28	***[IBIMC044]***	***[IMNFLG4]***<INT140FLG>
		ch29	***[IBIMC045]***	***[IMNFLG4]***<INT141FLG>
		ch30	***[IBIMC046]***	***[IMNFLG4]***<INT142FLG>
		ch31	***[IBIMC047]***	***[IMNFLG4]***<INT143FLG>

4.5. Interrupt Detection Level

When using interrupt via INTIF, interrupt detection level ("Low" level/"High" level/Rising edge/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 transmitted from the various peripheral functions to the CPU, a "High" pulse is output to the CPU as an interrupt request.

The CPU detects the interrupt signal "High" to be an interrupt factor.

4.5.1. Precautions When Releasing Low-power Consumption Mode

The following setting should be done when releasing STOP1 mode.

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

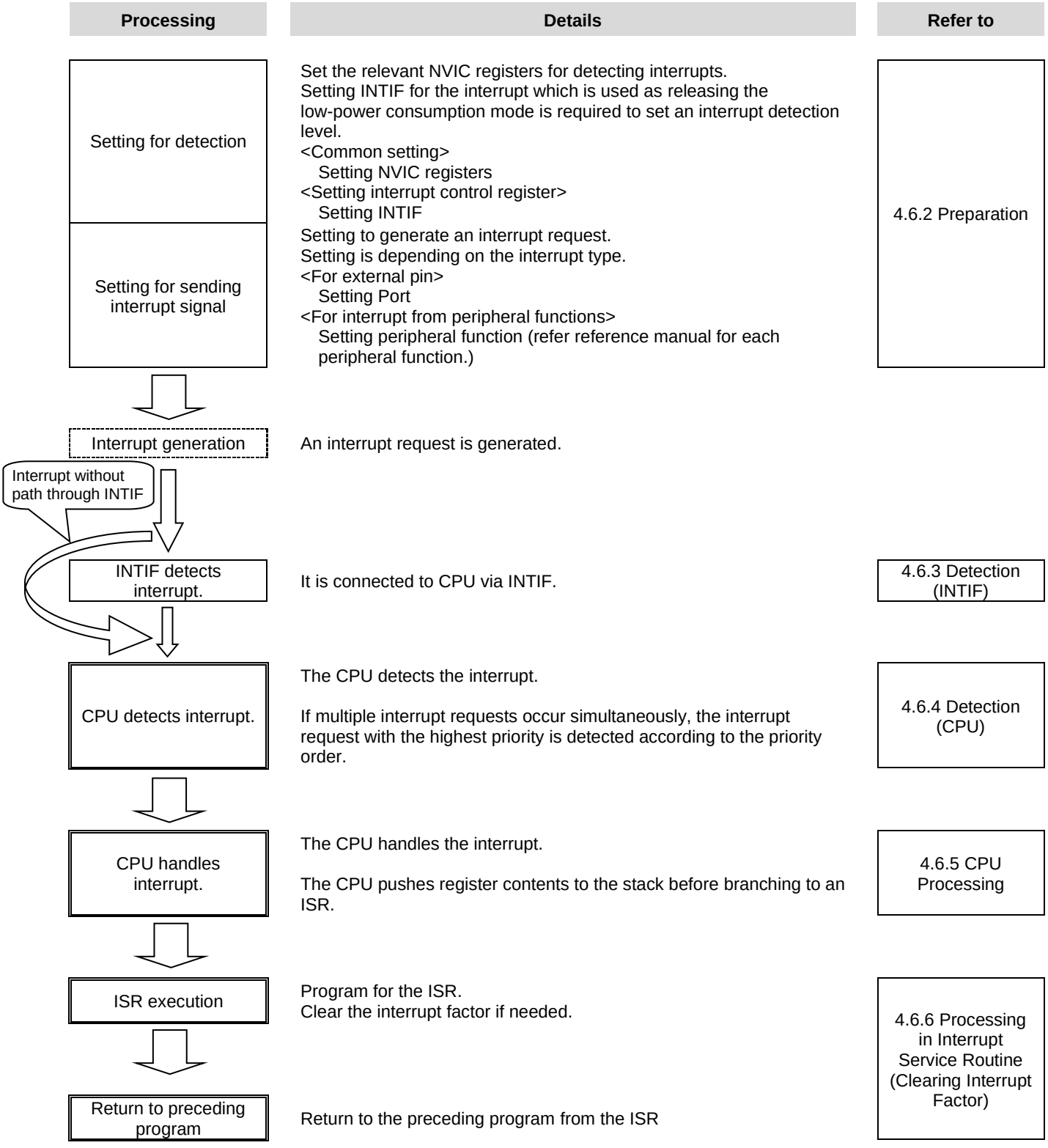
- The setup of the NVIC's interrupt enable set register
  Setting to Enable

When returning to NORMAL mode from STOP1 mode, resume suspended instruction by jumping into the interrupt after high-speed clock oscillation.

4.6. Interrupt Handling

4.6.1. Flowchart of Interrupt Handling

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



4.6.2. Preparation

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

Initiating an interrupt or changing its configuration must be implemented in the following order basically. First, disable the interrupt by the CPU. Then, configure setting from a circuit distant from the CPU. Finally, enable the interrupt by the CPU.

To configure the INTIF, you need to pay attention to the order of configuration to avoid any unexpected interrupt on the way. First, configure the precondition. Secondly, clear the information related to the interrupt in the INTIF and then enable the interrupt.

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

(a) Disabling interrupt by CPU
(b) Setting from a circuit distant from the CPU
(c) Preparation interrupt request (1) (Interrupt from external pin)
(d) Preparation interrupt request (2) (Interrupt from peripheral function)
(e) Preparation interrupt request (3) (Interrupt set-pending register)
(f) Setting the INTIF
(g) Enabling interrupt by CPU

(1) Disabling Interrupt by CPU

To make the CPU for not accepting any interrupt, write "1" to the corresponding bit of the ***[PRIMASK]*** register. 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 set in the user access level.

Note 2: If a fault occurs when ***[PRIMASK]*** register is set to "1", it is treated as a hard fault.

(2) Setting from a circuit distant from the CPU

You can assign a priority level by writing to <PRI_n> in the interrupt priority register in the NVIC. This register is assigned each 8 bits per interrupt factor. The number of bits is depending on each product. If they are 8 bits, a priority level can be set from 0 to 255. Priority level 0 is the highest priority level. If multiple factors have the same priority, the smallest-numbered interrupt factor has the highest priority. You can assign grouping priority by using the <PRIGROUP> in the application interrupt and reset control register.

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

Note: "n" indicates the number of the corresponding exceptions/interrupts. This product uses four bits for assigning a priority level.

(3) Preparation interrupt request (1) (Interrupt from external pin)

In order to use external interrupt pin, port setting for the corresponding pin is required. Setting ***[PxIE]***<PxmIE> to "1" allows the pin to be used as the input port.

Port register		
[PxIE]<PxmIE>	←	"1"

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

(4) Preparation interrupt request (2) (Interrupt from peripheral function)

The setting for interrupt request is depending on the peripheral function to be used. Refer to the reference manual of each peripheral function.

(5) Preparation interrupt request (3) (Interrupt by set-pending register)
To generate an interrupt by using the interrupt set-pending register, set the corresponding bit of this register to "1".

NVIC register		
<SETPEND>	←	"1"

Note: <SETPEND>: corresponding bit of interrupt set-pending register

(6) Setting INTIF

Setting the interrupt control register enables the interrupt detection of the interrupt via INTIF.

[IANIC00], ***[IBNIC00]***, and ***[IBIMCxxx]*** registers are setting register for each interrupt request. Before enabling an interrupt detection, clear the interrupt request in order to avoid unexpected interrupt.

For details of the interrupt control register, refer to the following.

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

Note: xxx: unique number assigned to each interrupt

(7) Enabling Interrupt by CPU

Enable the interrupt by the CPU as shown below.
Clear the pended interrupt by the interrupt clear-pending register. Enable an interrupt by the interrupt set-enable register. A bit of these registers is assigned to each interrupt factor.
Writing "1" to the corresponding bit of the interrupt clear-pending register clears the pended interrupt.
Writing "1" to the corresponding bit of the interrupt set-enable register enables an interrupt.
To generate interrupts by the interrupt set-pending register setting, interrupt factors are lost if pended interrupts are cleared. Thus, this operation is not required.
At the end, ***[PRIMASK]*** register is set to "0".

NVIC register		
<CLRPEND>	←	"1"
<SETENA>	←	"1"
Interrupt mask register		
[PRIMASK]	←	"0"

Note 1: <CLRPEND> and <SETENA>: corresponding bit of interrupt clear-pending register and interrupt set-enable register.

Note 2: ***[PRIMASK]*** register cannot be set in the user access level.

4.6.3. Detection (INTIF)

When the INTIF detects an interrupt, it sends the interrupt signal in "High" level to the CPU.

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

After it detected the interrupt, it keeps sending the interrupt signal in "High" level to the CPU until the detection flag is cleared by the interrupt control register. If exiting from the ISR without clearing the detection flag, the same interrupt is detected again. Thus, be sure to clear detection flag in the ISR.

At the same time, the corresponding bit of the interrupt monitor register is also cleared.

4.6.4. Detection (CPU)

The CPU detects an interrupt factor with the highest priority.

4.6.5. CPU Processing

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

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

An ISR requires specific programming according to the application to be used. This section describes about recommend process and clearing an interrupt factor.

(1) Process in ISR

An ISR normally pushes register contents to the stack and handles an interrupt processing.
The Cortex-M4 processor with FPU automatically pushes the contents of xPSR, PC, LR, r12, and r3 to 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. We recommend that you should push the contents of general-purpose registers that might be rewritten.

(2) Clearing interrupt factor
Some interrupts require clearing interrupt request by the interrupt control register.
If an interrupt detection level is set as level detection, an interrupt request continues to exist until its interrupt factor is cleared. Therefore, the interrupt factor must be cleared. If an interrupt factor is cleared in level detection, the interrupt request signal from INTIF will be deasserted automatically.
A interrupt factor is deasserted by clearing the detection flag of the interrupt control register of INTIF in edge detection. When an effective edge occurs again, it is anew recognized as a factor.

Note: After clearing the detection flag of the interrupt control register, be sure to read it which was cleared.

5. Exception/Interrupt-related Registers

5.1. Register List

Control registers and their addresses are as follows.

Interrupt Control Register 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

Note: Byte access is needed for the interrupt control register A.

Interrupt Control Registers B

Peripheral function		Channel/Unit	Base Address
Interrupt control register B	IB	-	0x40083200

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

Note: Byte access is needed for the interrupt control registers B.

Reset Flag Registers

Peripheral function		Channel/Unit	Base Address
Low-speed oscillation/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 needed for the reset flag registers.

Interrupt Monitor Registers

Peripheral function		Channel/Unit	Base Address
Interrupt Monitor	IMN	-	0x40083300

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

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 Clear-Enable Register 0	0x0180
Interrupt Clear-Enable Register 1	0x0184
Interrupt Clear-Enable Register 2	0x0188
Interrupt Set-Pending Register 0	0x0200
Interrupt Set-Pending Register 1	0x0204
Interrupt Set-Pending Register 2	0x0208
Interrupt Clear-Pending Register 0	0x0280
Interrupt Clear-Pending Register 1	0x0284
Interrupt Clear-Pending Register 2	0x0288
Interrupt Priority Register	0x0400 to 0x0457
Vector Table Offset Register	0x0D08
Application Interrupt and Reset Control Register	0x0D0C
System Handler Priority Register	0x0D18, 0x0D1C, 0x0D20
System Handler Control and Status Register	0x0D24

5.2. Interrupt Control Register A

5.2.1. *[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".

5.3. Interrupt Control Registers B

5.3.1. *[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".

5.3.2. *[IBIMC000]* to *[IBIMC095]* Interrupt B Mode Control Registers)

(1) ***[IBIMC000]*** to ***[IBIMC015]*** registers

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

(2) ***[IBIMC016]*** to ***[IBIMC095]*** registers

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".

5.4. Reset Flag Registers

5.4.1. *[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	-	Undefined	R	Read as an undefined value.
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.
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 release of power-on reset. When the release of power-on reset is detected, set "0" to all reset flags for initializing.

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

Bit	Bit symbol	After power-on reset	Type	Function
7:4	-	0	R	Read as "0".
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

5.5. Interrupt Monitor Registers

5.5.1. *[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

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

Bit	Bit symbol	After reset	Type	Function
31	INT127FLG	0	R	INTDMAATC (ch15) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
30	INT126FLG	0	R	INTDMAATC (ch14) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
29	INT125FLG	0	R	INTDMAATC (ch13) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
28	INT124FLG	0	R	INTDMAATC (ch12) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
27	INT123FLG	0	R	INTDMAATC (ch11) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
26	INT122FLG	0	R	INTDMAATC (ch10) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
25	INT121FLG	0	R	INTDMAATC (ch9) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
24	INT120FLG	0	R	INTDMAATC (ch8) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
23	INT119FLG	0	R	INTDMAATC (ch7) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
22	INT118FLG	0	R	INTDMAATC (ch6) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
21	INT117FLG	0	R	INTDMAATC (ch5) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
20	INT116FLG	0	R	INTDMAATC (ch4) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
19	INT115FLG	0	R	INTDMAATC (ch3) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
18	INT114FLG	0	R	INTDMAATC (ch2) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
17	INT113FLG	0	R	INTDMAATC (ch1) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
16	INT112FLG	0	R	INTDMAATC (ch0) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
15	INT111FLG	0	R	INTF interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
14	INT110FLG	0	R	INTE interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
13	INT109FLG	0	R	INTD interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
12	INT108FLG	0	R	INTC interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

Bit	Bit symbol	After reset	Type	Function
11	INT107FLG	0	R	INTB interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
10	INT106FLG	0	R	INTA interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
9	INT105FLG	0	R	INT9 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
8	INT104FLG	0	R	INT8 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
7	INT103FLG	0	R	INT7 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
6	INT102FLG	0	R	INT6 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
5	INT101FLG	0	R	INT5 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
4	INT100FLG	0	R	INT4 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
3	INT099FLG	0	R	INT3 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
2	INT098FLG	0	R	INT2 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
1	INT097FLG	0	R	INT1 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
0	INT096FLG	0	R	INT0 interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

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

Bit	Bit symbol	After reset	Type	Function
31:17	-	-	R	Read as "0".
16	INT144FLG	0	R	INTDMAAERR interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
15	INT143FLG	0	R	INTDMAATC (ch31) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
14	INT142FLG	0	R	INTDMAATC (ch30) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
13	INT141FLG	0	R	INTDMAATC (ch29) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
12	INT140FLG	0	R	INTDMAATC (ch28) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
11	INT139FLG	0	R	INTDMAATC (ch27) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
10	INT138FLG	0	R	INTDMAATC (ch26) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
9	INT137FLG	0	R	INTDMAATC (ch25) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
8	INT136FLG	0	R	INTDMAATC (ch24) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
7	INT135FLG	0	R	INTDMAATC (ch23) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
6	INT134FLG	0	R	INTDMAATC (ch22) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
5	INT133FLG	0	R	INTDMAATC (ch21) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
4	INT132FLG	0	R	INTDMAATC (ch20) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
3	INT131FLG	0	R	INTDMAATC (ch19) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
2	INT130FLG	0	R	INTDMAATC (ch18) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
1	INT129FLG	0	R	INTDMAATC (ch17) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
0	INT128FLG	0	R	INTDMAATC (ch16) interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

5.6. NVIC Registers

5.6.1. 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 this bit is set to "1", the value of the Reload Value Register is loaded to counter and count starts.

5.6.2. 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".

5.6.3. 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 register with any value clears it to "0". Clearing this register also clears the <COUNTFLAG> bit of the SysTick Control and Status Register.

5.6.4. 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.

5.6.5. Interrupt Control Registers

Following registers are used to control each interrupt factor; interrupt set-enable register, interrupt clear-enable register, interrupt set-pending register, and interrupt clear-pending register.

5.6.5.1. Interrupt Set-Enable Registers

These registers can enable interrupts and check enable/disable condition of interrupts.
Writing "1" to a bit in these registers enables the corresponding interrupt.
Writing "0" has no effect.
Reading the bits can check the enable/disable state of the corresponding interrupts.
Writing "1" to a corresponding bit in the interrupt clear-enable register clears the bit in these registers.

(1) 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		

(2) 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		
0	SETENA (Interrupt 32)	0		

(3) 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		

5.6.5.2. Interrupt Clear-Enable Registers

These registers can disable interrupts and check enable/disable condition of interrupts.
Writing "1" to a bit in these registers disables the corresponding interrupt.
Writing "0" has no effect.
Reading the bits can check the enable/disable state of the corresponding interrupts.

(1) 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		

(2) 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		
0	CLRENA (Interrupt 32)	0		

(3) 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 88)	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		

5.6.5.3. Interrupt Set-Pending Registers

These registers can force interrupts into the pending state and check that interrupts are currently pending.

Writing "1" to a bit in these registers pends the corresponding interrupt. However, writing "1" has no effect on an interrupt that is already pending or is disabled.

Writing "0" has no effect.

Reading the bits can check the pended/not pended state of the corresponding interrupt.

Writing "1" to a corresponding bit in the Interrupt clear-pending register clears the bit in these registers.

(1) 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 pended 1: Pended
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		

(2) 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 pended 1: Pended
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		
0	SETPEND (Interrupt 32)	Undefined		

(3) 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 pended 1: Pended
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		

5.6.5.4. Interrupt Clear-Pending Registers

These registers can clear pending interrupts and check that interrupts are currently pending.

Writing "1" to a bit in these registers clears the corresponding pending interrupt. However, writing "1" has no effect on an interrupt that is already being serviced.

Writing "0" has no effect.

Reading the bits can check the pended/not pended state of the corresponding interrupt.

(1) 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 pended 1: Pended
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 12)	Undefined		
11	CLRPEND (Interrupt 11)	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		

(2) 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 pended 1: Pended
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		
0	CLRPEND (Interrupt 32)	Undefined		

(3) 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 pended 1: Pended
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		

5.6.6. Interrupt Priority Register

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

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_33	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

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 configuration 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".

5.6.7. 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 you can use for up to 16 interrupts. For more interrupts, you must adjust the alignment by rounding up to the next power of two.
6:0	-	0	R	Read as "0".

5.6.8. 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>.
			R	Register key Read as "0xFA05".
15	ENDIANESS	0	R/W	Endianness bit (Note 1) 0: Little endian 1: Big endian
14:11	-	0	R	Read as "0".
10:8	PRIGROUP[2:0]	000	R/W	Interrupt priority grouping 000: 7-bit of pre-emption priority, 1-bit of sub priority 001: 6-bit of pre-emption priority, 2-bit of sub priority 010: 5-bit of pre-emption priority, 3-bit of sub priority 011: 4-bit of pre-emption priority, 4-bit of sub priority 100: 3-bit of pre-emption priority, 5-bit of sub priority 101: 2-bit of pre-emption priority, 6-bit of sub priority 110: 1-bit of pre-emption priority, 7-bit of sub priority 111: no pre-emption priority, 8-bit of sub priority This field configures 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 CPU outputs a SYSRESETREQ signal by writing to "1". (Note 2)
1	VECTCLRACTIVE	0	R/W	Clear active vector bit 0: Do not clear the state information. This bit is cleared by writing this bit to "1" automatically. 1: Clear all state information for active NMI, fault, and interrupts. It is the responsibility of the application to reinitialize the stack.
0	VECTRESET	0	R/W	System reset bit 0: Do not reset system. 1: Reset system. Reset the system, with the exception of debug components (FPB, DWT, and ITM) by setting this bit to "1" and this bit is also cleared

Note 1: Little endian is the default memory format for this product.

Note 2: When SYSRESETREQ is output, warm reset is performed on this product. <SYSRESETREQ> is cleared by warm reset.

5.6.9. System Handler Priority Register

Each exception is provided with eight bits of the 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 configuration of the system handler priority registers for interrupt numbers 4 to 7.
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".

5.6.10. System Handler Control and Status 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: You must clear or set the active bits with extreme caution because clearing or setting these bits does not repair stack contents.

6. Revision History

Table 6.1 Revision History

Revision	Date	Description
1.0	2024-08-30	- 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 INTERRUPTION 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.**