32-Bit RISC Microcontroller

TMPM4K Group(1)

Reference Manual
Exception
(EXCEPT-M4K(1))

Revision 1.1

2018-09

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Contents

List of Figures

List of Tables

Preface

Related document

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

Terms and Abbreviations

Some of abbreviations used in this document are as follows:

ADC	Analog to Digital Converter
A-ENC	Advanced Encoder input Circuit
A-PMD	Advanced Programmable Motor Control Circuit
A-VE+	Advanced Vector Engine Plus
DMAC	Direct Memory Access Controller
DNF	Digital Noise Filter
IA	Interrupt control register A
IB	Interrupt control register B
IMCxx	Interrupt Mode Control xx
IMNFLGNMI	Interrupt Monitor Flag NMI
IMNFLGx	Interrupt Monitor Flag x
INT	Interrupt
INTIF	Interrupt Interface Logic
ISR	Interrupt Service Routine
I^2C	Inter-Integrated Circuit
LVD	Voltage Detection Circuit
NICxx	Non-Maskable Interrupt Control xx
NVIC	Nested Vectored Interrupt Controller
OFD	Oscillation Frequency Detector
POR	Power On Reset Circuit
RLMRSTFLGx	RLM Reset Flag x
SIWDT	Clock Selective Watchdog Timer
TSPI	Toshiba 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 documentation set for the Arm Cortex-M4 processors" 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 on each exception, refer to "Arm documentation set for the Arm Cortex-M4 processors".

- 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	See
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 (ISR).	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 sources include instruction execution by the CPU, memory accesses, and interrupt requests from external interrupt pins or peripheral functions.

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

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

The request of the exception by the external interruption terminal or the peripheral function occurs by each functional factor. Regarding to interruption which connected via INTIF, the setup of the interrupt control register is needed. For details, refer to the chapter, "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 reset, OFD reset, SIWDT reset, LVD reset, SYSRESETREQ reset, LOCKUP reset	0x00
Non-Maskable Interrupt	-2	SIWDT, LVD	0x08
Hard Fault	-1	Fault that cannot activate because a higher-priority fault is being handled or it is disabled	0x0C
Memory Management	Configurable	Exception from the Memory Protection Unit (MPU) Instruction fetch from the Execute Never (XN) region	0x10
Bus Fault	Configurable	Access violation to the Hard Fault region of the memory map	0x14
Usage Fault	Configurable	Undefined instruction execution or other faults related to instruction execution	0x18
Reserved	-		0x1C to 0x28
SVCall	Configurable	System service call with SVC instruction	0x2C
Debug Monitor	Configurable	Debug monitor when the CPU is not faulting	0x30
Reserved	-		0x34
PendSV	Configurable	Pending system service request	0x38
SysTick	Configurable	Notification from system timer	0x3C
External Interrupt	Configurable	External interrupt pin or peripheral function (Note)	0x40

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

(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> bit is defined as the upper 4-bit configuration with TMPM4K Group(1) products. 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.
The 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	None	[7:0]	1	256

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

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

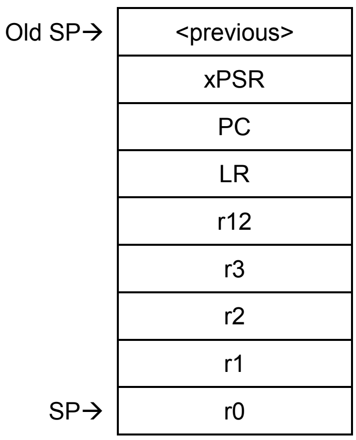
When an exception occurs, the CPU suspends the currently executing process and branches to the interrupt service routine. This is called "pre-emption".

(1) Stacking

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

1. Program Counter (PC)
2. Program Status Register (xPSR)
3. r0 to r3
4. r12
5. 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 an 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 may 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 an 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, see "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:

- 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".

- Returning to the 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.

- 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:

- Pop eight registers

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

- 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.

- Select SP

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

2. Reset Exception

Reset exceptions are generated from the following seven sources.
Use the ***[RLMRSTFLGn]*** of the Reset Flag Register to identify the source 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 reference manual "Power supply and Reset operation"

- Reset exception by OFD

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

- Reset exception by SIWDT

  The watchdog timer (SIWDT) has a reset generating feature. For details, refer to reference manual "Clock Selective Watchdog Timer".

- Reset exception by LVD

  The LVD has a reset generating feature. For details, refer to reference manual "Voltage Detector Circuit".

- Reset exception by <SYSRESETREQ>

  A reset can be generated by setting the <SYSRESETREQ> bit 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 CPU when the un-recoverable interrupt 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 you set a value in the SysTick Reload Value Register and enable the SysTick features in 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. You may be pending exceptions and use a flag to know when 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 two sources.

- Non-maskable interrupt by SIWDT
  The watchdog timer (SIWDT) has a non-maskable interrupt generating feature. For details, refer to reference manual "Clock Selective Watchdog Timer".

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

4.2. Maskable Interrupt

Please refer to interrupt control register A / interrupt control register B of the "4.4 List of Interrupt Sources" for the factor of mask interruption.

4.3. Interrupt Request

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

4.3.1. Interrupt Route

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

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

- The interrupt that is releasable from IDLE, STOP1 mode
  Interruption which can be released of IDLE and STOP1 mode is controlled by the interrupt control register A in INTIF via INTIF, and is notified to CPU. (Route A, B, C)

- The interrupt that is releasable from IDLE, STOP1 mode
  Interruption which can be released of IDLE and the STOP1 mode is controlled by the interrupt control register B in INTIF via INTIF, and is notified CPU. (Route D, E, F)

- The interrupt that is releasable from IDLE mode
  Although some factors of interruption which can be released of IDLE mode are controlled by the interrupt control register B via INTIF (Route G), other factors are notified to CPU directly (Route H) not passing through INTIF.

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

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

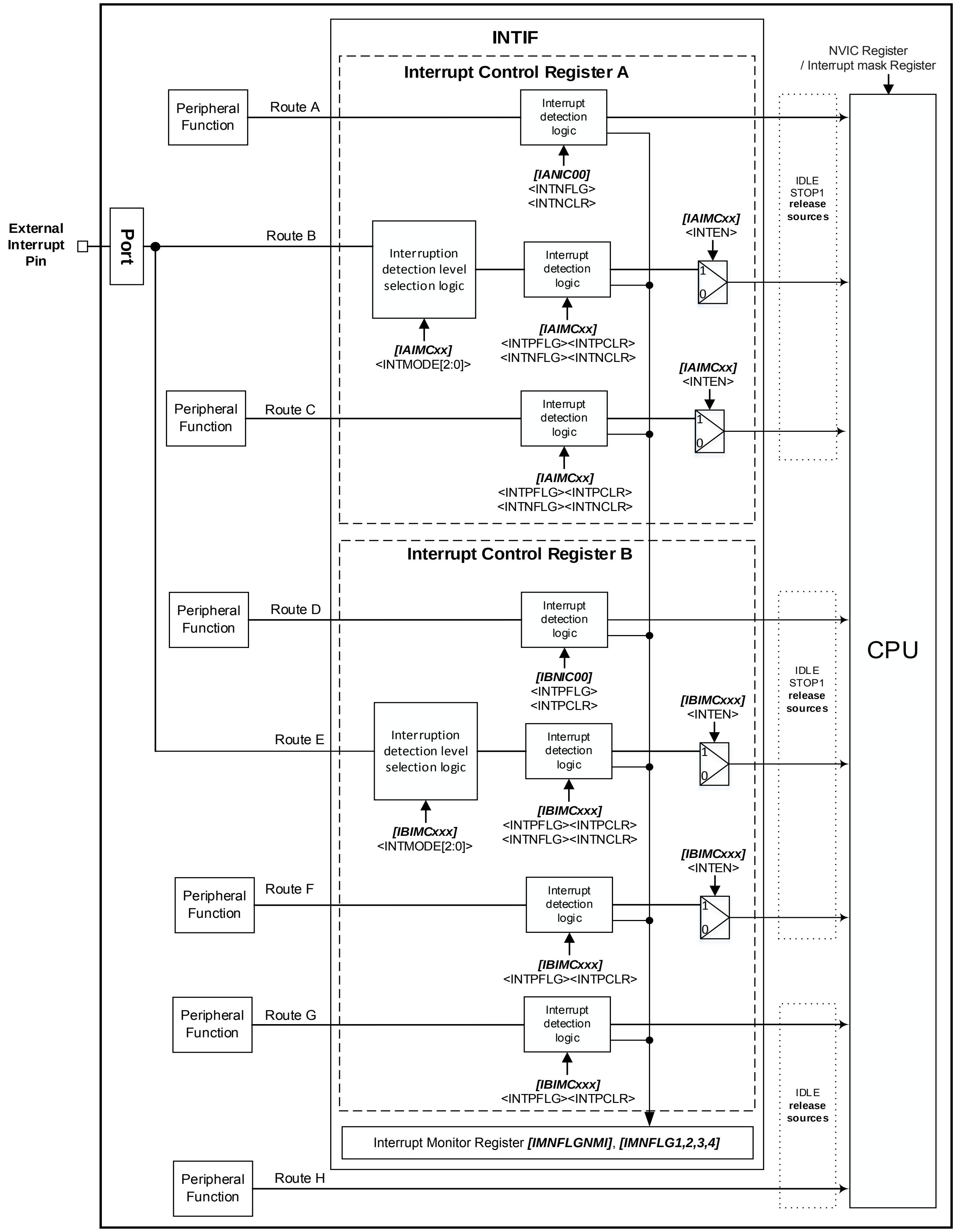


Figure 4.1 Interruption transfer route Diagram

Table 4.1 Explanation of each interruption transfer route

Route	Interrupt No.	Interrupt Request	Route Description
A	-	LVD interrupt	This route is NMI interrupt. It is a route inputted into CPU via INTIF. An interrupt release setup is carried out by the interrupt control register A (***[IANIC00]***).
B	0, 1, 2,3	External interrupts (00,01,02,03)	The interrupt request of a port is a route inputted into CPU via INTIF. Permission/prohibition of selection of an Interruption detection level, interrupt release, and an interrupt request are set up by the interrupt control register A (***[IAIMCxx]***) for every factor.
C	-	-	It is a route inputted into CPU via INTIF. Permission/prohibition of interrupt release and an interrupt request are set up by the interrupt control register A (***[IAIMCxx]***) for every factor.
D	-	WDT interrupt	It is mask impossible interruption. It is a route inputted into CPU via INTIF. An interrupt release setup is carried out by the interrupt control register B (***[IBNIC00]***).
E	4 to 10	External interrupts (04 to 10)	The interrupt request of a port is a route inputted into CPU via INTIF. Permission/prohibition of selection of an Interruption detection level, interrupt release, and an interrupt request are set up by the interrupt control register B (***[IBIMCxxx]***) for every factor.
F	-	-	It is a route inputted into CPU via INTIF. Permission/prohibition of interruption are set up by the interrupt control register B (***[IBIMCxxx]***) for every factor.
G	84 to 85	DMAC transmission end interrupt (ch0 to 31) DMAC transmission error interrupt (Note)	It is a route inputted into CPU via INTIF. An interrupt release setup is carried out by the interrupt control register B (***[IBIMCxxx]***) for every factor.
H	11 to 83, 87	Other interrupts	It is a route as which an interrupt request is directly inputted into CPU not passing through INTIF.

Note: Interruption of DMAC transfer end is interruption by which interruption of two or more channels was combined with one interruption number. Please refer to "4.4.1 About joint interruption" for details.

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 sources, or by setting the relevant bit of NVIC's Interrupt Set-Pending Register for interrupt request source.

- 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.
  See the chapter of each peripheral function for details.
- By setting Interrupt Set-Pending Register (forced pending)
  An interrupt request can be forced to be generated by setting the relevant bit of the Interrupt Set-Pending Register of NVIC.

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

4.3.3. Monitor of the Interrupt Request

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

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

4.3.4. Transmission of Interrupt Request

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

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

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

4.3.5. Precautions When Using External Interrupt Pins

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

If input is disabled (***[PxIE]***<PxmIE>=0), inputs from external interrupt pins are "Low". When the <INTMODE> bit of Interrupt Control Register A (***[IAIMCxx]***) is "Low", then input signals from the external interrupt pins are sent to the CPU as is.　Since the CPU recognizes "Low" input as an interrupt, interrupts occur if corresponding interrupts are enabled by the CPU as inputs are being disabled.

To use the external interrupt without setting it as a trigger to release the low power consumption mode, set the interrupt pin input as "High" and enable it. Then, enable interrupts on the CPU.

4.4. List of Interrupt Sources

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

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

Interrupt Source	Interrupt Request	Interrupt Control Register	Interrupt Monitor Register
INTLVD	LVD interrupt	***[IANIC00]***	***[IMNFLGNMI]*** <INT000FLG>
INTWDT0	WDT interrupt	***[IBNIC00]***	***[IMNFLGNMI]*** <INT016FLG>

Table 4.3 shows the list of interrupt sources of Interrupt Control Register A. These interrupt sources can be the sources for releasing the low power consumption mode. The Interrupt Control Register A will perform several setting for detecting the release of the low power consumption mode, and interrupt enable/disable.

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

Interrupt No,	Interrupt Source	Interrupt Request	Interrupt Control Register	Interrupt Monitor Register
0	INT00	External interrupt 00a	***[IAIMC00]***	***[IMNFLG1]*** <INT032FLG>
		External interrupt 00b	***[IAIMC32]***	***[IMNFLG2]*** <INT064FLG>
1	INT01	External interrupt 01a	***[IAIMC01]***	***[IMNFLG1]*** <INT033FLG>
		External interrupt 01b	***[IAIMC33]***	***[IMNFLG2]*** <INT065FLG>
2	INT02	External interrupt 02a	***[IAIMC02]***	***[IMNFLG1]*** <INT034FLG>
		External interrupt 02b	***[IAIMC34]***	***[IMNFLG2]*** <INT066FLG>
3	INT03	External interrupt 03a	***[IAIMC03]***	***[IMNFLG1]*** <INT035FLG>
		External interrupt 03b	***[IAIMC35]***	***[IMNFLG2]*** <INT067FLG>

The factor list of the interrupt control registers B is shown in Table 4.4 to Table 4.7. A part of interruption sets up interruption permission / prohibition by the interrupt control register B.

Table 4.4 List of Interrupt Sources (Interrupt Control Register B) (1/4)

Interrupt No.	Interrupt Source	Interrupt Request	Interrupt Control Register	Interrupt Monitor Register
4	INT04	External interrupt 04	***[IBIMC033]***	***[IMNFLG4]*** <INT129FLG>
5	INT05	External interrupt 05	***[IBIMC034]***	***[IMNFLG4]*** <INT130FLG>
6	INT06	External interrupt 06	***[IBIMC035]***	***[IMNFLG4]*** <INT131FLG>
7	INT07	External interrupt 07a	***[IBIMC036]***	***[IMNFLG4]*** <INT132FLG>
		External interrupt 07b	***[IBIMC040]***	***[IMNFLG4]*** <INT136FLG>
8	INT08	External interrupt 08	***[IBIMC037]***	***[IMNFLG4]*** <INT133FLG>
9	INT09	External interrupt 09	***[IBIMC038]***	***[IMNFLG4]*** <INT134FLG>
10	INT10	External interrupt 10	***[IBIMC039]***	***[IMNFLG4]*** <INT135FLG>
11	INTVCN0	A-VE+ ch0 Schedule end interrupt		
12	INTVCT0	A-VE+ ch0 Task end interrupt		
13	INTEMG0	A-PMD ch0 EMG interrupt		
14	INTEMG1	A-PMD ch1 EMG interrupt		
15	INTOVV0	A-PMD ch0 OVV interrupt		
16	INTOVV1	A-PMD ch1 OVV interrupt		
17	INTPWM0	A-PMD ch0 PWM interrupt		
18	INTPWM1	A-PMD ch1 PWM interrupt		
19	INTENC00	A-ENC ch0 Encoder input interrupt 0		
20	INTENC01	A-ENC ch0 Encoder input interrupt 1		
21	INTADAPDA	ADC unit A PMD trigger interrupt A		
22	INTADAPDB	ADC unit A PMD trigger interrupt B		
23	INTADAPDC	ADC unit A PMD trigger interrupt C		
24	INTADAPDD	ADC unit A PMD trigger interrupt D		
25	INTADAPFLG	ADC unit A Priority interrupt		
26	INTADACP0	ADC unit A Monitor function 0 interrupt		
27	INTADACP1	ADC unit A Monitor function 1 interrupt		
28	INTADATRG	ADC unit A General purpose trigger interrupt		
29	INTADASGL	ADC unit A Single conversion interrupt		
30	INTADACNT	ADC unit A Continuous conversion interrupt		

Table 4.5 List of Interrupt Sources (Interrupt Control Register B) (2/4)

Interrupt No.	Interrupt Source	Interrupt Request	Interrupt Control Register	Interrupt Monitor Register
31	INTSC0RX	TSPI ch0 Receive interrupt UART ch0 Reception interrupt		
32	INTSC0TX	TSPI ch0 Transmit interrupt UART ch0 Transmission interrupt		
33	INTSC0ERR	TSPI ch0 Error interrupt UART ch0 Error interrupt		
34	INTSC1RX	TSPI ch1 Receive interrupt UART ch1 Reception interrupt		
35	INTSC1TX	TSPI ch1 Transmit interrupt UART ch1 Transmission interrupt		
36	INTSC1ERR	TSPI ch1 Error interrupt UART ch1 Error interrupt		
37	INTSC2RX	TSPI ch2 Receive interrupt UART ch2 Reception interrupt		
38	INTSC2TX	TSPI ch2 Transmit interrupt UART ch2 Transmission interrupt		
39	INTSC2ERR	TSPI ch2 Error interrupt UART ch2 Error interrupt		
40	INTSC3RX	TSPI ch3 Receive interrupt UART ch3 Reception interrupt		
41	INTSC3TX	TSPI ch3 Transmit interrupt UART ch3 Transmission interrupt		
42	INTSC3ERR	TSPI ch3 Error interrupt UART ch3 Error interrupt		
43	INTI2C0	I^2C ch0 I^2C interrupt		
44	INTI2C0AL	I^2C ch0 I^2C arbitration lost detection interrupt		
45	INTI2C0BF	I^2C ch0 I^2C bus free detection interrupt		
46	INTI2C0NA	I^2C ch0 I^2C NACK detection interrupt		
47	INTT32A0AC	T32A ch0 timer A/C match, overflow, and underflow		
48	INTT32A0ACCAP0	T32A ch0 timer A/C capture 0		
49	INTT32A0ACCAP1	T32A ch0 timer A/C capture 1		
50	INTT32A0B	T32A ch0 timer B match, overflow, and underflow		
51	INTT32A0BCAP0	T32A ch0 timer B capture 0		
52	INTT32A0BCAP1	T32A ch0 timer B capture 1		
53	INTT32A1AC	T32A ch1 timer A/C match, Overflow, and underflow		
54	INTT32A1ACCAP0	T32A ch1 timer A/C capture 0		
55	INTT32A1ACCAP1	T32A ch1 timer A/C capture 1		
56	INTT32A1B	T32A ch1 timer B match, overflow, and underflow		
57	INTT32A1BCAP0	T32A ch1 timer B capture 0		
58	INTT32A1BCAP1	T32A ch1 timer B capture 1		

Table 4.6 List of Interrupt Sources (Interrupt Control Register B) (3/4)

Interrupt No.	Interrupt Source	Interrupt Request	Interrupt Control Register	Interrupt Monitor Register
59	INTT32A2AC	T32A ch2 timer A/C match, overflow, and underflow		
60	INTT32A2ACCAP0	T32A ch2 timer A/C capture 0		
61	INTT32A2ACCAP1	T32A ch2 timer A/C capture 1		
62	INTT32A2B	T32A ch2 timer B match, overflow, and underflow		
63	INTT32A2BCAP0	T32A ch2 timer B capture 0		
64	INTT32A2BCAP1	T32A ch2 timer B capture 1		
65	INTT32A3AC	T32A ch3 timer A/C match, overflow, and underflow		
66	INTT32A3ACCAP0	T32A ch3 timer A/C capture 0		
67	INTT32A3ACCAP1	T32A ch3 timer A/C capture 1		
68	INTT32A3B	T32A ch3 timer B match, overflow, and underflow		
69	INTT32A3BCAP0	T32A ch3 timer B capture 0		
70	INTT32A3BCAP1	T32A ch3 timer B capture 1		
71	INTT32A4AC	T32A ch4 timer A/C match, overflow, and underflow		
72	INTT32A4ACCAP0	T32A ch4 timer A/C capture 0		
73	INTT32A4ACCAP1	T32A ch4 timer A/C capture 1		
74	INTT32A4B	T32A ch4 timer B match, overflow, and underflow		
75	INTT32A4BCAP0	T32A ch4 timer B capture 0		
76	INTT32A4BCAP1	T32A ch4 timer B capture 1		
77	INTT32A5AC	T32A ch5 timer A/C match, overflow, and underflow		
78	INTT32A5ACCAP0	T32A ch5 timer A/C capture 0		
79	INTT32A5ACCAP1	T32A ch5 timer A/C capture 1		
80	INTT32A5B	T32A ch5 timer B match, overflow, and underflow		
81	INTT32A5BCAP0	T32A ch5 timer B capture 0		
82	INTT32A5BCAP1	T32A ch5 timer B capture 1		

Table 4.7 List of Interrupt Sources (Interrupt Control Register B) (4/4)

Interrupt No.	Interrupt Source	Interrupt Request	Interrupt Control Register	Interrupt Monitor Register
83	INTPARI	RAMP RAM Parity interrupt		
84	INTDMAATC	DMAC unit A transmission end interrupt (ch0 to 31)	***[IBIMC000]*** to ***[IBIMC031]*** (Note)	***[IMNFLG3]*** <INT096FLG> to <INT127FLG> (Note)
85	INTDMAAERR	DMAC unit A transmission error interrupt	***[IBIMC032]***	***[IMNFLG4]*** <INT128FLG>
87	INTFLCRDY	Code FLASH Ready interrupt		

Note: Please refer to "4.4.1 About joint interruption".

4.4.1. About joint interruption

The details of interruption united in TMPM4K Group(1) are as follows.

Table 4.8 Joint interruption list (1)

Interrupt No,	Interrupt Source	Interrupt Request	Interrupt Control Register	Interrupt Monitor Register
0	INT00	External interrupt 00a	***[IAIMC00]***	***[IMNFLG1]***<INT032FLG>
		External interrupt 00b	***[IAIMC32]***	***[IMNFLG2]***<INT064FLG>
1	INT01	External interrupt 01a	***[IAIMC01]***	***[IMNFLG1]***<INT033FLG>
		External interrupt 01b	***[IAIMC33]***	***[IMNFLG2]***<INT065FLG>
2	INT02	External interrupt 02a	***[IAIMC02]***	***[IMNFLG1]***<INT034FLG>
		External interrupt 02b	***[IAIMC34]***	***[IMNFLG2]***<INT066FLG>
3	INT03	External interrupt 03a	***[IAIMC03]***	***[IMNFLG1]***<INT035FLG>
		External interrupt 03b	***[IAIMC35]***	***[IMNFLG2]***<INT067FLG>
7	INT07	External interrupt 07a	***[IBIMC036]***	***[IMNFLG4]***<INT132FLG>
		External interrupt 07b	***[IBIMC040]***	***[IMNFLG4]***<INT136FLG>

Table 4.9 Joint interruption list (2)

Interrupt No.	Interrupt Source		Interrupt Control Register	Interrupt Monitor Register
110	DMAC unit A transmission end interrupt (INTDMAATC)	ch0	***[IBIMC000]***	***[IMNFLG3]***<INT096FLG>
		ch1	***[IBIMC001]***	***[IMNFLG3]***<INT097FLG>
		ch2	***[IBIMC002]***	***[IMNFLG3]***<INT098FLG>
		ch3	***[IBIMC003]***	***[IMNFLG3]***<INT099FLG>
		ch4	***[IBIMC004]***	***[IMNFLG3]***<INT100FLG>
		ch5	***[IBIMC005]***	***[IMNFLG3]***<INT101FLG>
		ch6	***[IBIMC006]***	***[IMNFLG3]***<INT102FLG>
		ch7	***[IBIMC007]***	***[IMNFLG3]***<INT103FLG>
		ch8	***[IBIMC008]***	***[IMNFLG3]***<INT104FLG>
		ch9	***[IBIMC009]***	***[IMNFLG3]***<INT105FLG>
		ch10	***[IBIMC010]***	***[IMNFLG3]***<INT106FLG>
		ch11	***[IBIMC011]***	***[IMNFLG3]***<INT107FLG>
		ch12	***[IBIMC012]***	***[IMNFLG3]***]<INT108FLG>
		ch13	***[IBIMC013]***	***[IMNFLG3***]<INT109FLG>
		ch14	***[IBIMC014]***	***[IMNFLG3]***<INT110FLG>
		ch15	***[IBIMC015]***	***[IMNFLG3]***<INT111FLG>
		ch16	***[IBIMC016]***	***[IMNFLG3]***<INT112FLG>
		ch17	***[IBIMC017]***	***[IMNFLG3]***<INT113FLG>
		ch18	***[IBIMC018]***	***[IMNFLG3]***<INT114FLG>
		ch19	***[IBIMC019]***	***[IMNFLG3]***<INT115FLG>
		ch20	***[IBIMC020]***	***[IMNFLG3]***<INT116FLG>
		ch21	***[IBIMC021]***	***[IMNFLG3]***<INT117FLG>
		ch22	***[IBIMC022]***	***[IMNFLG3]***<INT118FLG>
		ch23	***[IBIMC023]***	***[IMNFLG3]***<INT119FLG>
		ch24	***[IBIMC024]***	***[IMNFLG3]***<INT120FLG>
		ch25	***[IBIMC025]***	***[IMNFLG3]***<INT121FLG>
		ch26	***[IBIMC026]***	***[IMNFLG3]***<INT122FLG>
		ch27	***[IBIMC027]***	***[IMNFLG3]***<INT123FLG>
		ch28	***[IBIMC028]***	***[IMNFLG3]***<INT124FLG>
		ch29	***[IBIMC029]***	***[IMNFLG3]***<INT125FLG>
		ch30	***[IBIMC030]***	***[IMNFLG3]***<INT126FLG>
		ch31	***[IBIMC031]***	***[IMNFLG3]***<INT127FLG>

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 the Low Power Consumption Mode

Following setting should be done when releasing STOP1 mode.

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

- The setup of the NVIC interrupt enable set register
  - enable/disable setup

In order to return to NORMAL mode from STOP1 mode, resume suspended instruction by jumping into interrupt after high speed clock oscillation.

4.6. Interrupt Handling

4.6.1. Flowchart

The following shows how an interrupt is handled.
The flowchart below explains the interrupt handling process by hardware and software.

Processing	Details	See
Setting for detection	Set the relevant NVIC registers for detecting interrupts. Setting to INTIF will be necessary when the interrupt require active level setting for releasing the low power consumption mode. <Common setting> NVIC registers <Setting for releasing the Interrupt Control Registers> INTIF	4.6.2 Preparation
Setting for sending interrupt request	Execute an appropriate setting to send the interrupt request depending on the interrupt type. <Setting for interrupt from external pin> Port <Setting for interrupt from peripheral function> Peripheral function (see the chapter of each peripheral function for details)	4.6.2 Preparation
⇩		
Interrupt request generation	An interrupt request is generated	
⇩ (Interruption which does not connect via INTIF: bypasses INTIF detect interrupt)		
INTIF detect interrupt	It is connected to CPU via INTIF.	4.6.3 Detection (INTIF)
⇩		
CPU detects interrupt	The CPU detects the interrupt. If multiple interrupt requests occur simultaneously, the interrupt request with the highest priority is detected according to the priority order.	4.6.4 Detection (CPU)
⇩		
CPU handles interrupt	The CPU handles the interrupt The CPU pushes register contents to the stack before entering the ISR.	4.6.5 CPU Processing
⇩		
ISR execution	Program for the ISR Clear the interrupt source if needed	4.6.6 Interrupt Service Routine (ISR) (Clearing an interrupt Source)
⇩		
Return to preceding program	Configure to return to the preceding program of the ISR	4.6.6 Interrupt Service Routine (ISR) (Clearing an interrupt Source)

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 from the farthest route from the CPU. Finally, enable the interrupt by the CPU.

To configure the INTIF, you must follow the order indicated here not to cause any unexpected interrupt. First, configure the precondition. Secondly, clear the data 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.

1. Disabling interrupt by CPU
2. CPU registers setting
3. Preconfiguration (1) (Interrupt from external pin)
4. Preconfiguration (2) (Interrupt from peripheral function)
5. Preconfiguration (3) (Interrupt Set-Pending Register)
6. Configuring the INTIF
7. 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 modified in the user access level

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

(2) CPU Registers Setting

You can assign a priority level by writing to <PRI_n> field in an Interrupt Priority Register in the NVIC.

Each interrupt source is provided with eight bits for assigning a priority level from 0 to 255, but the number of bits actually used varies with each product. Priority level 0 is the highest priority level. If multiple sources have the same priority, the smallest-numbered interrupt source 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) Preconfiguration (1) (Interrupt from external pin)

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

Port Register		
[PxIE]<PxmIE>	←	"1"

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

(4) Preconfiguration (2) (Interrupt from peripheral function)

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

(5) Preconfiguration (3) (Interrupt from Set-Pending Register)

To generate an interrupt by using the Interrupt Set-Pending Register, set "1" to the corresponding bit of this register.

NVIC Register		
<SETPEND>	←	"1"

Note: <SETPEND>: corresponding bit

(6) Configuring the INTIF

The interrupt by way of INTIF sets the permission of the interrupt in interrupt control registers.

The ***[IANIC00]***/***[IBNIC00]***/***[IAIMCxx]***/***[IBIMCxxx]*** registers are capable of configuring each interrupt source. Before enabling an interrupt, clear the interrupt request having active level in order to avoid unexpected interrupt.
For details of the interrupt control register, refer to the following.

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

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

(7) Enabling Interrupt by CPU

Enable the interrupt by the CPU as shown below.

Clear the suspended interrupt in the Interrupt Clear-Pending Register. Enable the intended interrupt with the Interrupt Set-Enable Register. Each bit of the register is assigned to a single interrupt source.

Writing "1" to the corresponding bit of the Interrupt Clear-Pending Register clears the suspended interrupt. Writing "1" to the corresponding bit of the Interrupt Set-Enable Register enables the intended interrupt.

To generate interrupts in the Interrupt Set-Pending Register setting, factors to trigger interrupts are lost if pending interrupts are cleared. Thus, this operation is not necessary.

At the end, ***[PRIMASK]*** register is zero cleared.

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

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

Note 2: ***[PRIMASK]*** Register cannot be modified by the user access level;

4.6.3. Detection (INTIF)

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

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

It keeps sending the interrupt signal in "High" level to the CPU until the interrupt request is cleared by <detection flag> in the Interrupt Control Register. If the ISR is exited without clearing the interrupt request, the same interrupt will be detected again when normal operation is resumed. Thus, be sure to clear <detection flag> each interrupt request in the ISR. At the same time, the corresponding interrupt monitor register is also cleared.

4.6.4. Detection (CPU)

The CPU detects an interrupt request 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-r0 to the stack then enter the ISR.

4.6.6. Interrupt Service Routine (ISR) (Clearing an interrupt Source)

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

(1) Process in the Interrupt Service Routine

An ISR normally pushes register contents to the stack and handles an interrupt as required.

The Cortex-M4 processor with FPU automatically pushes the contents of xPSR, PC, LR, r12, and r3-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 you to push the contents of general-purpose registers that might be rewritten.

(2) Clearing an Interrupt Source

Some interrupt requests have to be cleared with the Interrupt Control Register.

If an interruption detection level is set as level-sensitive, an interrupt request continues to exist until it is cleared at its source. Therefore, the interrupt source must be cleared. If a factor is withdrawn in level detection, the interrupt request signal from INTIF will be withdrawn automatically.

A factor is withdrawn by clearing the interruption flag of the Interrupt Control Register of INTIF in the case of edge detection. When an effective edge occurs again, it is anew recognized as a factor.

Note: After clearing the interrupt flag of the Interrupt Control Register, please be sure to read the flag which was cleared.

5. Exception/ Interrupt-Related Registers

5.1. Register List

Control Registers and their addresses are as follows;

Interrupt Control Registers A

Peripheral function	Function name	Channel/Unit	Base address
Interrupt control register A	IA	-	0x4003E000

Register name		Address (+BASE)
Non-Maskable Interrupt A Control Register 00	***[IANIC00]***	0x0000
Interrupt A Mode Control Register 00	***[IAIMC00]***	0x0020
Interrupt A Mode Control Register 01	***[IAIMC01]***	0x0021
Interrupt A Mode Control Register 02	***[IAIMC02]***	0x0022
Interrupt A Mode Control Register 03	***[IAIMC03]***	0x0023
Interrupt A Mode Control Register 32	***[IAIMC32]***	0x0040
Interrupt A Mode Control Register 33	***[IAIMC33]***	0x0041
Interrupt A Mode Control Register 34	***[IAIMC34]***	0x0042
Interrupt A Mode Control Register 35	***[IAIMC35]***	0x0043

Note: Byte access is needed for ***[IANIC00]*** and ***[IAIMCxx]***

Interrupt Control Registers B

Peripheral function	Function name	Channel/Unit	Base address
Interrupt control register B	IB	-	0x400F4E00

Register name		Address (+BASE)
Non-Maskable Interrupt B Control Register 00	***[IBNIC00]***	0x0010
Interrupt B Mode Control Register 000	***[IBIMC000]***	0x0060
Interrupt B Mode Control Register 001	***[IBIMC001]***	0x0061
Interrupt B Mode Control Register 002	***[IBIMC002]***	0x0062
Interrupt B Mode Control Register 003	***[IBIMC003]***	0x0063
Interrupt B Mode Control Register 004	***[IBIMC004]***	0x0064
Interrupt B Mode Control Register 005	***[IBIMC005]***	0x0065
Interrupt B Mode Control Register 006	***[IBIMC006]***	0x0066
Interrupt B Mode Control Register 007	***[IBIMC007]***	0x0067
Interrupt B Mode Control Register 008	***[IBIMC008]***	0x0068
Interrupt B Mode Control Register 009	***[IBIMC009]***	0x0069
Interrupt B Mode Control Register 010	***[IBIMC010]***	0x006A
Interrupt B Mode Control Register 011	***[IBIMC011]***	0x006B
Interrupt B Mode Control Register 012	***[IBIMC012]***	0x006C
Interrupt B Mode Control Register 013	***[IBIMC013]***	0x006D
Interrupt B Mode Control Register 014	***[IBIMC014]***	0x006E
Interrupt B Mode Control Register 015	***[IBIMC015]***	0x006F
Interrupt B Mode Control Register 016	***[IBIMC016]***	0x0070
Interrupt B Mode Control Register 017	***[IBIMC017]***	0x0071
Interrupt B Mode Control Register 018	***[IBIMC018]***	0x0072
Interrupt B Mode Control Register 019	***[IBIMC019]***	0x0073
Interrupt B Mode Control Register 020	***[IBIMC020]***	0x0074
Interrupt B Mode Control Register 021	***[IBIMC021]***	0x0075
Interrupt B Mode Control Register 022	***[IBIMC022]***	0x0076
Interrupt B Mode Control Register 023	***[IBIMC023]***	0x0077
Interrupt B Mode Control Register 024	***[IBIMC024]***	0x0078
Interrupt B Mode Control Register 025	***[IBIMC025]***	0x0079
Interrupt B Mode Control Register 026	***[IBIMC026]***	0x007A
Interrupt B Mode Control Register 027	***[IBIMC027]***	0x007B
Interrupt B Mode Control Register 028	***[IBIMC028]***	0x007C
Interrupt B Mode Control Register 029	***[IBIMC029]***	0x007D
Interrupt B Mode Control Register 030	***[IBIMC030]***	0x007E
Interrupt B Mode Control Register 031	***[IBIMC031]***	0x007F
Interrupt B Mode Control Register 032	***[IBIMC032]***	0x0080
Interrupt B Mode Control Register 033	***[IBIMC033]***	0x0081
Interrupt B Mode Control Register 034	***[IBIMC034]***	0x0082
Interrupt B Mode Control Register 035	***[IBIMC035]***	0x0083
Interrupt B Mode Control Register 036	***[IBIMC036]***	0x0084
Interrupt B Mode Control Register 037	***[IBIMC037]***	0x0085
Interrupt B Mode Control Register 038	***[IBIMC038]***	0x0086
Interrupt B Mode Control Register 039	***[IBIMC039]***	0x0087
Interrupt B Mode Control Register 040	***[IBIMC040]***	0x0088

Note: Byte access is needed for ***[IBNIC00]*** and ***[IBIMCxxx]*** Registers

Reset Flag Registers

Peripheral function	Function name	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 Reset Flag Register

Interrupt Monitor Registers

Peripheral function	Function name	Channel/Unit	Base address
Interrupt Monitor	IMN	-	0x400F4F00

Register name		Address (+BASE)
Non-Maskable Interrupt Monitor Flag Register	*[IMNFLGNMI]*	0x0000
Interrupt Monitor Flag Register 1	*[IMNFLG1]*	0x0004
Interrupt Monitor Flag Register 2	*[IMNFLG2]*	0x0008
Interrupt Monitor Flag Register 3	*[IMNFLG3]*	0x000C
Interrupt Monitor Flag Register 4	*[IMNFLG4]*	0x0010

NVIC Registers

Peripheral function	Function name	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 State Register	0x0D24

5.2. Interrupt Control Registers 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 Reading the bit returns "0"
6	-	0	R	Read as "0"
5	INTNFLG	0	R	Detection flag 0: Not detected 1: Detected
4:0	-	00101	R	Read as "00101"

5.2.2. *[IAIMC00 to 03, 32 to 35]* (Interrupt A Mode Control Registers)

(1) ***[IAIMC00 to 03, 32 to 35]*** Registers

Bit	Bit Symbol	After Reset	Type	Function
7	INTNCLR	0	W	Falling edge detection flag clear control 0: - 1: Clear Reading the bit returns "0"
6	INTPCLR	0	W	Rising edge detection flag clear control 0: - 1: Clear Reading the bit returns "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	Interruption detection level selection 000: Low level 001: High level 010: Falling edge 011: Rising edge 100: Both edge 101: Reserved 110: Reserved 111: Reserved
0	INTEN	0	R/W	Interrupt control 0: Interrupt detection disabled 1: Interrupt detection enabled

5.3. 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 Reading the bit returns "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 040]* (Interrupt B Mode Control Registers)

(1) ***[IBIMC000 to 032]*** 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 Reading the bit returns "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"

(2) ***[IBIMC033 to 040]*** Registers

Bit	Bit Symbol	After Reset	Type	Function
7	INTNCLR	0	W	Falling edge detection flag clear control 0: - 1: Clear Reading the bit returns "0"
6	INTPCLR	0	W	Rising edge detection flag clear control 0: - 1: Clear Reading the bit returns "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	Interruption detection level selection 000: Low level 001: High level 010: Falling edge 011: Rising edge 100: Both edge 101: Reserved 110: Reserved 111: Reserved
0	INTEN	0	R/W	Interrupt control 0: Interrupt detection disabled 1: Interrupt detection enabled

5.4. 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 valule.
5	LVDRSTF	Undefined	R	LVD reset flag 0: - 1: Reset from LVD
			W	LVD reset flag 0: Clear 1: don't care
4	-	Undefined	R	Read as an undefined value.
			W	Write "0"
3	PINRSTF	Undefined	R	Reset pin flag 0: - 1: Reset from reset pin
			W	Reset pin flag 0: Clear 1: don't care
2:1	-	Undefined	R	Read as an undefined value.
			W	Write "00"
0	PORSTF	1	R	Power on reset flag 0: - 1: Reset from by power on reset
			W	Power on reset flag 0: Clear 1: don't care

Note: Reset flags except <PORSTF> become undefined after power on reset release. When release of power on reset is detected, please write in to "0" to all the reset flags for initialize.

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 from OFD
			W	OFD reset flag 0: Clear 1: don’t care
2	WDTRSTF	0	R	SIWDT reset flag 0: - 1: Reset from SIWDT
			W	SIWDT reset flag 0: Clear 1: don’t care
1	LOCKRSTF	0	R	LOCKUP reset flag 0: - 1: Reset from LOCKUP
			W	LOCKUP reset flag 0: Clear 1: don’t care
0	SYSRSTF	0	R	<SYSRESETREQ> reset flag 0: - 1: Reset from <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. *[IMNFLG1]* (Interrupt Monitor Flag Register 1)

Bit	Bit Symbol	After Reset	Type	Function
31:4	-	0	R	Read as "0"
3	INT035FLG	0	R	INT03a Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
2	INT034FLG	0	R	INT02a Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
1	INT033FLG	0	R	INT01a Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
0	INT032FLG	0	R	INT00a Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

5.5.3. *[IMNFLG2]* (Interrupt Monitor Flag Register 2)

Bit	Bit Symbol	After Reset	Type	Function
31:4	-	0	R	Read as "0"
3	INT067FLG	0	R	INT03b Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
2	INT066FLG	0	R	INT02b Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
1	INT065FLG	0	R	INT01b Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
0	INT064FLG	0	R	INT00b Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

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

Bit	Bit Symbol	After Reset	Type	Function
31	INT127FLG	0	R	INTDMAATC(ch31) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
30	INT126FLG	0	R	INTDMAATC(ch30) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
29	INT125FLG	0	R	INTDMAATC(ch29) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
28	INT124FLG	0	R	INTDMAATC(ch28) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
27	INT123FLG	0	R	INTDMAATC(ch27) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
26	INT122FLG	0	R	INTDMAATC(ch26) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
25	INT121FLG	0	R	INTDMAATC(ch25) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
24	INT120FLG	0	R	INTDMAATC(ch24) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
23	INT119FLG	0	R	INTDMAATC(ch23) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
22	INT118FLG	0	R	INTDMAATC(ch22) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
21	INT117FLG	0	R	INTDMAATC(ch21) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
20	INT116FLG	0	R	INTDMAATC(ch20) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
19	INT115FLG	0	R	INTDMAATC(ch19) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
18	INT114FLG	0	R	INTDMAATC(ch18) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
17	INT113FLG	0	R	INTDMAATC(ch17) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
16	INT112FLG	0	R	INTDMAATC(ch16) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
15	INT111FLG	0	R	INTDMAATC(ch15) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

Bit	Bit Symbol	After Reset	Type	Function
14	INT110FLG	0	R	INTDMAATC(ch14) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
13	INT109FLG	0	R	INTDMAATC(ch13) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
12	INT108FLG	0	R	INTDMAATC(ch12) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
11	INT107FLG	0	R	INTDMAATC(ch11) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
10	INT106FLG	0	R	INTDMAATC(ch10) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
9	INT105FLG	0	R	INTDMAATC(ch9) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
8	INT104FLG	0	R	INTDMAATC(ch8) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
7	INT103FLG	0	R	INTDMAATC(ch7) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
6	INT102FLG	0	R	INTDMAATC(ch6) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
5	INT101FLG	0	R	INTDMAATC(ch5) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
4	INT100FLG	0	R	INTDMAATC(ch4) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
3	INT099FLG	0	R	INTDMAATC(ch3) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
2	INT098FLG	0	R	INTDMAATC(ch2) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
1	INT097FLG	0	R	INTDMAATC(ch1) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
0	INT096FLG	0	R	INTDMAATC(ch0) Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected

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

Bit	Bit Symbol	After Reset	Type	Function
31:9	-	0	R	Read as "0"
8	INT136FLG	0	R	INT07b Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
7	INT135FLG	0	R	INT10 Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
6	INT134FLG	0	R	INT09 Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
5	INT133FLG	0	R	INT08 Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
4	INT132FLG	0	R	INT07a Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
3	INT131FLG	0	R	INT06 Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
2	INT130FLG	0	R	INT05 Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
1	INT129FLG	0	R	INT04 Interrupt detection flag 0: Interrupt not detected 1: Interrupt detected
0	INT128FLG	0	R	INTDMAAERR 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 "1" is set, it re-load with the value of the Reload Value register and starts operation.

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 four registers will be used to control each interrupt source; Interrupt Set-Enable Register, Interrupt Clear-Enable Register, Interrupt Set-Pending Register, and Interrupt Clear-Pending Register.

5.6.5.1. Interrupt Set-Enable Register

Each bit corresponds to the specified number of interrupts. It can enable interrupts and check if interrupts are enabled.

Writing "1" to a bit in this register enables the corresponding interrupt.

Writing "0" has no effect.

Reading the bits can see the enable/disable condition of the corresponding interrupts. Writing "1" to a corresponding bit in the Interrupt Clear-Enable Register clears the bit in this register.

(a) Interrupt Set-Enable Register 0

Bit	Bit Symbol	After Reset	Type	Function
31	SETENA (Interrupt31)	0	R/W	[Write] 1: Enable interrupt [Read] 0: Interrupt is disabled 1: Interrupt is enabled
30	SETENA (Interrupt30)	0		
29	SETENA (Interrupt29)	0		
28	SETENA (Interrupt28)	0		
27	SETENA (Interrupt27)	0		
26	SETENA (Interrupt26)	0		
25	SETENA (Interrupt25)	0		
24	SETENA (Interrupt24)	0		
23	SETENA (Interrupt23)	0		
22	SETENA (Interrupt22)	0		
21	SETENA (Interrupt21)	0		
20	SETENA (Interrupt20)	0		
19	SETENA (Interrupt19)	0		
18	SETENA (Interrupt18)	0		
17	SETENA (Interrupt17)	0		
16	SETENA (Interrupt16)	0		
15	SETENA (Interrupt15)	0		
14	SETENA (Interrupt14)	0		
13	SETENA (Interrupt13)	0		
12	SETENA (Interrupt12)	0		
11	SETENA (Interrupt11)	0		
10	SETENA (Interrupt10)	0		
9	SETENA (Interrupt9)	0		
8	SETENA (Interrupt8)	0		
7	SETENA (Interrupt7)	0		
6	SETENA (Interrupt6)	0		
5	SETENA (Interrupt5)	0		
4	SETENA (Interrupt4)	0		
3	SETENA (Interrupt3)	0		
2	SETENA (Interrupt2)	0		
1	SETENA (Interrupt1)	0		
0	SETENA (Interrupt0)	0		

(b) Interrupt Set-Enable Register 1

Bit	Bit Symbol	After Reset	Type	Function
31	SETENA (Interrupt63)	0	R/W	[Write] 1: Enable interrupt [Read] 0: Interrupt is disabled 1: Interrupt is enabled
30	SETENA (Interrupt62)	0		
29	SETENA (Interrupt61)	0		
28	SETENA (Interrupt60)	0		
27	SETENA (Interrupt59)	0		
26	SETENA (Interrupt58)	0		
25	SETENA (Interrupt57)	0		
24	SETENA (Interrupt56)	0		
23	SETENA (Interrupt55)	0		
22	SETENA (Interrupt54)	0		
21	SETENA (Interrupt53)	0		
20	SETENA (Interrupt52)	0		
19	SETENA (Interrupt51)	0		
18	SETENA (Interrupt50)	0		
17	SETENA (Interrupt49)	0		
16	SETENA (Interrupt48)	0		
15	SETENA (Interrupt47)	0		
14	SETENA (Interrupt46)	0		
13	SETENA (Interrupt45)	0		
12	SETENA (Interrupt44)	0		
11	SETENA (Interrupt43)	0		
10	SETENA (Interrupt42)	0		
9	SETENA (Interrupt41)	0		
8	SETENA (Interrupt40)	0		
7	SETENA (Interrupt39)	0		
6	SETENA (Interrupt38)	0		
5	SETENA (Interrupt37)	0		
4	SETENA (Interrupt36)	0		
3	SETENA (Interrupt35)	0		
2	SETENA (Interrupt34)	0		
1	SETENA (Interrupt33)	0		
0	SETENA (Interrupt32)	0		

(c) Interrupt Set-Enable Register 2

Bit	Bit Symbol	After Reset	Type	Function
31:24	-	0	R	Read as "0"
23	SETENA (Interrupt87)	0	R/W	[Write] 1: Enable interrupt [Read] 0: Interrupt is disabled 1: Interrupt is enabled
22	SETENA (Interrupt86)	0		
21	SETENA (Interrupt85)	0		
20	SETENA (Interrupt84)	0		
19	SETENA (Interrupt83)	0		
18	SETENA (Interrupt82)	0		
17	SETENA (Interrupt81)	0		
16	SETENA (Interrupt80)	0		
15	SETENA (Interrupt79)	0		
14	SETENA (Interrupt78)	0		
13	SETENA (Interrupt77)	0		
12	SETENA (Interrupt76)	0		
11	SETENA (Interrupt75)	0		
10	SETENA (Interrupt74)	0		
9	SETENA (Interrupt73)	0		
8	SETENA (Interrupt72)	0		
7	SETENA (Interrupt71)	0		
6	SETENA (Interrupt70)	0		
5	SETENA (Interrupt69)	0		
4	SETENA (Interrupt68)	0		
3	SETENA (Interrupt67)	0		
2	SETENA (Interrupt66)	0		
1	SETENA (Interrupt65)	0		
0	SETENA (Interrupt64)	0		

5.6.5.2. Interrupt Clear-Enable Register

Each bit corresponds to the specified number of interrupts. It can disable interrupts and check if interrupts are disabled.

Writing "1" to a bit in this register disables the corresponding interrupt.

Writing "0" has no effect.

Reading the bits can see the enable/disable condition of the corresponding interrupts.

(a) Interrupt Clear-Enable Register 0

Bit	Bit Symbol	After Reset	Type	Function
31	CLRENA (Interrupt31)	0	R/W	[Write] 1: Disable Interrupt [Read] 0: Interrupt is disabled 1: Interrupt is enabled
30	CLRENA (Interrupt30)	0		
29	CLRENA (Interrupt29)	0		
28	CLRENA (Interrupt28)	0		
27	CLRENA (Interrupt27)	0		
26	CLRENA (Interrupt26)	0		
25	CLRENA (Interrupt25)	0		
24	CLRENA (Interrupt24)	0		
23	CLRENA (Interrupt23)	0		
22	CLRENA (Interrupt22)	0		
21	CLRENA (Interrupt21)	0		
20	CLRENA (Interrupt20)	0		
19	CLRENA (Interrupt19)	0		
18	CLRENA (Interrupt18)	0		
17	CLRENA (Interrupt17)	0		
16	CLRENA (Interrupt16)	0		
15	CLRENA (Interrupt15)	0		
14	CLRENA (Interrupt14)	0		
13	CLRENA (Interrupt13)	0		
12	CLRENA (Interrupt12)	0		
11	CLRENA (Interrupt11)	0		
10	CLRENA (Interrupt10)	0		
9	CLRENA (Interrupt9)	0		
8	CLRENA (Interrupt8)	0		
7	CLRENA (Interrupt7)	0		
6	CLRENA (Interrupt6)	0		
5	CLRENA (Interrupt5)	0		
4	CLRENA (Interrupt4)	0		
3	CLRENA (Interrupt3)	0		
2	CLRENA (Interrupt2)	0		
1	CLRENA (Interrupt1)	0		
0	CLRENA (Interrupt0)	0		

(b) Interrupt Clear-Enable Register 1

Bit	Bit Symbol	After Reset	Type	Function
31	CLRENA (Interrupt63)	0	R/W	[Write] 1: Disable Interrupt [Read] 0: Interrupt is disabled 1: Interrupt is enabled
30	CLRENA (Interrupt62)	0		
29	CLRENA (Interrupt61)	0		
28	CLRENA (Interrupt60)	0		
27	CLRENA (Interrupt59)	0		
26	CLRENA (Interrupt58)	0		
25	CLRENA (Interrupt57)	0		
24	CLRENA (Interrupt56)	0		
23	CLRENA (Interrupt55)	0		
22	CLRENA (Interrupt54)	0		
21	CLRENA (Interrupt53)	0		
20	CLRENA (Interrupt52)	0		
19	CLRENA (Interrupt51)	0		
18	CLRENA (Interrupt50)	0		
17	CLRENA (Interrupt49)	0		
16	CLRENA (Interrupt48)	0		
15	CLRENA (Interrupt47)	0		
14	CLRENA (Interrupt46)	0		
13	CLRENA (Interrupt45)	0		
12	CLRENA (Interrupt44)	0		
11	CLRENA (Interrupt43)	0		
10	CLRENA (Interrupt42)	0		
9	CLRENA (Interrupt41)	0		
8	CLRENA (Interrupt40)	0		
7	CLRENA (Interrupt39)	0		
6	CLRENA (Interrupt38)	0		
5	CLRENA (Interrupt37)	0		
4	CLRENA (Interrupt36)	0		
3	CLRENA (Interrupt35)	0		
2	CLRENA (Interrupt34)	0		
1	CLRENA (Interrupt33)	0		
0	CLRENA (Interrupt32)	0		

(c) Interrupt Clear-Enable Register 2

Bit	Bit Symbol	After Reset	Type	Function
31:24	-	0	R	Read as "0"
23	CLRENA (Interrupt87)	0	R/W	[Write] 1: Disable Interrupt [Read] 0: Interrupt is disabled 1: Interrupt is enabled
22	CLRENA (Interrupt86)	0		
21	CLRENA (Interrupt85)	0		
20	CLRENA (Interrupt84)	0		
19	CLRENA (Interrupt83)	0		
18	CLRENA (Interrupt82)	0		
17	CLRENA (Interrupt81)	0		
16	CLRENA (Interrupt80)	0		
15	CLRENA (Interrupt79)	0		
14	CLRENA (Interrupt78)	0		
13	CLRENA (Interrupt77)	0		
12	CLRENA (Interrupt76)	0		
11	CLRENA (Interrupt75)	0		
10	CLRENA (Interrupt74)	0		
9	CLRENA (Interrupt73)	0		
8	CLRENA (Interrupt72)	0		
7	CLRENA (Interrupt71)	0		
6	CLRENA (Interrupt70)	0		
5	CLRENA (Interrupt69)	0		
4	CLRENA (Interrupt68)	0		
3	CLRENA (Interrupt67)	0		
2	CLRENA (Interrupt66)	0		
1	CLRENA (Interrupt65)	0		
0	CLRENA (Interrupt64)	0		

5.6.5.3. Interrupt Set-Pending Register

Each bit corresponds to the specified number of interrupts. It can force interrupts into the pending state and determines which interrupts are currently pending.

Writing "1" to a bit in this register 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 bit returns the current state of the corresponding interrupts.

Writing "1" to a corresponding bit in the Interrupt Clear-Pending Register clears the bit in this register.

(a) Interrupt Set-Pending Register 0

Bit	Bit Symbol	After Reset	Type	Function
31	SETPEND (Interrupt31)	Undefined	R/W	[Write] 1: Pend interrupt [Read] 0: Not pending 1: Pending
30	SETPEND (Interrupt30)	Undefined		
29	SETPEND (Interrupt29)	Undefined		
28	SETPEND (Interrupt28)	Undefined		
27	SETPEND (Interrupt27)	Undefined		
26	SETPEND (Interrupt26)	Undefined		
25	SETPEND (Interrupt25)	Undefined		
24	SETPEND (Interrupt24)	Undefined		
23	SETPEND (Interrupt23)	Undefined		
22	SETPEND (Interrupt22)	Undefined		
21	SETPEND (Interrupt21)	Undefined		
20	SETPEND (Interrupt20)	Undefined		
19	SETPEND (Interrupt19)	Undefined		
18	SETPEND (Interrupt18)	Undefined		
17	SETPEND (Interrupt17)	Undefined		
16	SETPEND (Interrupt16)	Undefined		
15	SETPEND (Interrupt15)	Undefined		
14	SETPEND (Interrupt14)	Undefined		
13	SETPEND (Interrupt13)	Undefined		
12	SETPEND (Interrupt12)	Undefined		
11	SETPEND (Interrupt11)	Undefined		
10	SETPEND (Interrupt10)	Undefined		
9	SETPEND (Interrupt9)	Undefined		
8	SETPEND (Interrupt8)	Undefined		
7	SETPEND (Interrupt7)	Undefined		
6	SETPEND (Interrupt6)	Undefined		
5	SETPEND (Interrupt5)	Undefined		
4	SETPEND (Interrupt4)	Undefined		
3	SETPEND (Interrupt3)	Undefined		
2	SETPEND (Interrupt2)	Undefined		
1	SETPEND (Interrupt1)	Undefined		
0	SETPEND (Interrupt0)	Undefined		

(b) Interrupt Set-Pending Register 1

Bit	Bit Symbol	After Reset	Type	Function
31	SETPEND (Interrupt63)	Undefined	R/W	[Write] 1: Pend interrupt [Read] 0: Not pending 1: Pending
30	SETPEND (Interrupt62)	Undefined		
29	SETPEND (Interrupt61)	Undefined		
28	SETPEND (Interrupt60)	Undefined		
27	SETPEND (Interrupt59)	Undefined		
26	SETPEND (Interrupt58)	Undefined		
25	SETPEND (Interrupt57)	Undefined		
24	SETPEND (Interrupt56)	Undefined		
23	SETPEND (Interrupt55)	Undefined		
22	SETPEND (Interrupt54)	Undefined		
21	SETPEND (Interrupt53)	Undefined		
20	SETPEND (Interrupt52)	Undefined		
19	SETPEND (Interrupt51)	Undefined		
18	SETPEND (Interrupt50)	Undefined		
17	SETPEND (Interrupt49)	Undefined		
16	SETPEND (Interrupt48)	Undefined		
15	SETPEND (Interrupt47)	Undefined		
14	SETPEND (Interrupt46)	Undefined		
13	SETPEND (Interrupt45)	Undefined		
12	SETPEND (Interrupt44)	Undefined		
11	SETPEND (Interrupt43)	Undefined		
10	SETPEND (Interrupt42)	Undefined		
9	SETPEND (Interrupt41)	Undefined		
8	SETPEND (Interrupt40)	Undefined		
7	SETPEND (Interrupt39)	Undefined		
6	SETPEND (Interrupt38)	Undefined		
5	SETPEND (Interrupt37)	Undefined		
4	SETPEND (Interrupt36)	Undefined		
3	SETPEND (Interrupt35)	Undefined		
2	SETPEND (Interrupt34)	Undefined		
1	SETPEND (Interrupt33)	Undefined		
0	SETPEND (Interrupt32)	Undefined		

(c) Interrupt Set-Pending Register 2

Bit	Bit Symbol	After Reset	Type	Function
31:24	-	0	R	Read as "0"
23	SETPEND (Interrupt87)	Undefined	R/W	[Write] 1: Pend interrupt [Read] 0: Not pending 1: Pending
22	SETPEND (Interrupt86)	Undefined		
21	SETPEND (Interrupt85)	Undefined		
20	SETPEND (Interrupt84)	Undefined		
19	SETPEND (Interrupt83)	Undefined		
18	SETPEND (Interrupt82)	Undefined		
17	SETPEND (Interrupt81)	Undefined		
16	SETPEND (Interrupt80)	Undefined		
15	SETPEND (Interrupt79)	Undefined		
14	SETPEND (Interrupt78)	Undefined		
13	SETPEND (Interrupt77)	Undefined		
12	SETPEND (Interrupt76)	Undefined		
11	SETPEND (Interrupt75)	Undefined		
10	SETPEND (Interrupt74)	Undefined		
9	SETPEND (Interrupt73)	Undefined		
8	SETPEND (Interrupt72)	Undefined		
7	SETPEND (Interrupt71)	Undefined		
6	SETPEND (Interrupt70)	Undefined		
5	SETPEND (Interrupt69)	Undefined		
4	SETPEND (Interrupt68)	Undefined		
3	SETPEND (Interrupt67)	Undefined		
2	SETPEND (Interrupt66)	Undefined		
1	SETPEND (Interrupt65)	Undefined		
0	SETPEND (Interrupt64)	Undefined		

5.6.5.4. Interrupt Clear-Pending Register

Each bit corresponds to the specified number of interrupt. It can clear pending interrupts and determines which interrupts are currently pending.

Writing "1" to a bit in this register 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 bit returns the current state of the corresponding interrupts.

(a) Interrupt Clear-Pending Register 0

Bit	Bit Symbol	After Reset	Type	function
31	CLRPEND (Interrupt31)	Undefined	R/W	[Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending
30	CLRPEND (Interrupt30)	Undefined		
29	CLRPEND (Interrupt29)	Undefined		
28	CLRPEND (Interrupt28)	Undefined		
27	CLRPEND (Interrupt27)	Undefined		
26	CLRPEND (Interrupt26)	Undefined		
25	CLRPEND (Interrupt25)	Undefined		
24	CLRPEND (Interrupt24)	Undefined		
23	CLRPEND (Interrupt23)	Undefined		
22	CLRPEND (Interrupt22)	Undefined		
21	CLRPEND (Interrupt21)	Undefined		
20	CLRPEND (Interrupt20)	Undefined		
19	CLRPEND (Interrupt19)	Undefined		
18	CLRPEND (Interrupt18)	Undefined		
17	CLRPEND (Interrupt17)	Undefined		
16	CLRPEND (Interrupt16)	Undefined		
15	CLRPEND (Interrupt15)	Undefined		
14	CLRPEND (Interrupt14)	Undefined		
13	CLRPEND (Interrupt13)	Undefined		
12	CLRPEND (Interrupt12)	Undefined		
11	CLRPEND (Interrupt11)	Undefined		
10	CLRPEND (Interrupt10)	Undefined		
9	CLRPEND (Interrupt9)	Undefined		
8	CLRPEND (Interrupt8)	Undefined		
7	CLRPEND (Interrupt7)	Undefined		
6	CLRPEND (Interrupt6)	Undefined		
5	CLRPEND (Interrupt5)	Undefined		
4	CLRPEND (Interrupt4)	Undefined		
3	CLRPEND (Interrupt3)	Undefined		
2	CLRPEND (Interrupt2)	Undefined		
1	CLRPEND (Interrupt1)	Undefined		
0	CLRPEND (Interrupt0)	Undefined		

(b) Interrupt Clear-Pending Register 1

Bit	Bit Symbol	After Reset	Type	Function
31	CLRPEND (Interrupt63)	Undefined	R/W	[Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending
30	CLRPEND (Interrupt62)	Undefined		
29	CLRPEND (Interrupt61)	Undefined		
28	CLRPEND (Interrupt60)	Undefined		
27	CLRPEND (Interrupt59)	Undefined		
26	CLRPEND (Interrupt58)	Undefined		
25	CLRPEND (Interrupt57)	Undefined		
24	CLRPEND (Interrupt56)	Undefined		
23	CLRPEND (Interrupt55)	Undefined		
22	CLRPEND (Interrupt54)	Undefined		
21	CLRPEND (Interrupt53)	Undefined		
20	CLRPEND (Interrupt52)	Undefined		
19	CLRPEND (Interrupt51)	Undefined		
18	CLRPEND (Interrupt50)	Undefined		
17	CLRPEND (Interrupt49)	Undefined		
16	CLRPEND (Interrupt48)	Undefined		
15	CLRPEND (Interrupt47)	Undefined		
14	CLRPEND (Interrupt46)	Undefined		
13	CLRPEND (Interrupt45)	Undefined		
12	CLRPEND (Interrupt44)	Undefined		
11	CLRPEND (Interrupt43)	Undefined		
10	CLRPEND (Interrupt42)	Undefined		
9	CLRPEND (Interrupt41)	Undefined		
8	CLRPEND (Interrupt40)	Undefined		
7	CLRPEND (Interrupt39)	Undefined		
6	CLRPEND (Interrupt38)	Undefined		
5	CLRPEND (Interrupt37)	Undefined		
4	CLRPEND (Interrupt36)	Undefined		
3	CLRPEND (Interrupt35)	Undefined		
2	CLRPEND (Interrupt34)	Undefined		
1	CLRPEND (Interrupt33)	Undefined		
0	CLRPEND (Interrupt32)	Undefined		

(c) Interrupt Clear-Pending Register 2

Bit	Bit Symbol	After Reset	Type	Function
31:24	-	0	R	Read as "0"
23	CLRPEND (Interrupt87)	Undefined	R/W	[Write] 1: Clear pending interrupt [Read] 0: Not pending 1: Pending
22	CLRPEND (Interrupt86)	Undefined		
21	CLRPEND (Interrupt85)	Undefined		
20	CLRPEND (Interrupt84)	Undefined		
19	CLRPEND (Interrupt83)	Undefined		
18	CLRPEND (Interrupt82)	Undefined		
17	CLRPEND (Interrupt81)	Undefined		
16	CLRPEND (Interrupt80)	Undefined		
15	CLRPEND (Interrupt79)	Undefined		
14	CLRPEND (Interrupt78)	Undefined		
13	CLRPEND (Interrupt77)	Undefined		
12	CLRPEND (Interrupt76)	Undefined		
11	CLRPEND (Interrupt75)	Undefined		
10	CLRPEND (Interrupt74)	Undefined		
9	CLRPEND (Interrupt73)	Undefined		
8	CLRPEND (Interrupt72)	Undefined		
7	CLRPEND (Interrupt71)	Undefined		
6	CLRPEND (Interrupt70)	Undefined		
5	CLRPEND (Interrupt69)	Undefined		
4	CLRPEND (Interrupt68)	Undefined		
3	CLRPEND (Interrupt67)	Undefined		
2	CLRPEND (Interrupt66)	Undefined		
1	CLRPEND (Interrupt65)	Undefined		
0	CLRPEND (Interrupt64)	Undefined		

5.6.6. Interrupt Priority Register

Each interrupt is provided with eight bits of an Interrupt Priority Register.
The following shows the addresses of the Interrupt Priority Registers corresponding to interrupt numbers.

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

The number of bits to be used for assigning a priority varies with each product. This product uses four bits for assigning a priority.
The following shows the fields of the Interrupt Priority Registers for interrupt numbers 0 to 3. The Interrupt Priority Registers for all other interrupt numbers have the identical fields. 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> field.
			R	Register key Read as 0xFA05.
15	ENDIANESS	0	R/W	Endianness bit: (Note 1) 1: Big endian 0: Little endian
14:11	-	0	R	Read as "0"
10:8	PRIGROUP[2:0]	000	R/W	Interrupt priority grouping 000: seven bits of pre-emption priority, one bit of sub priority 001: six bits of pre-emption priority, two bits of sub priority 010: five bits of pre-emption priority, three bits of sub priority 011: four bits of pre-emption priority, four bits of sub priority 100: three bits of pre-emption priority, five bits of sub priority 101: two bits of pre-emption priority, six bits of sub priority 110: one bit of pre-emption priority, seven bits of sub priority 111: no pre-emption priority, eight bits of sub priority The bit configuration to split the interrupt priority register <PRI_n> into pre-emption priority and sub priority.
7:3	-	0	R	Read as "0"
2	SYSRESETREQ	0	R/W	System Reset Request 1=CPU outputs a SYSRESETREQ signal. (Note 2)
1	VECTCLRACTIVE	0	R/W	Clear active vector bit 1: clear all state information for active NMI, fault, and interrupts. 0: do not clear. This bit self-clears. It is the responsibility of the application to reinitialize the stack.
0	VECTRESET	0	R/W	System Reset bit 1: reset system. 0: do not reset system. Resets the system, with the exception of debug components (FPB, DWT and ITM) by setting "1" and this bit is also zero 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 a System Handler Priority Register.
The following shows the addresses of the System Handler Priority Registers corresponding to each exception.

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

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

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

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

5.6.10. System Handler Control and State Register

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

Note: You must clear or set the active bits with extreme caution because clearing and setting these bits does not repair stack contents.

6. List of Interrupt Sources for Each Product

6.1. TMPM4K4/TMPM4K2/TMPM4K1/TMPM4K0

M4K4	M4K2	M4K1	M4K0	Interrupt No	Interrupt Source	Interrupt Request	Interrupt Control Register	Interrupt Monitor Register
✓	✓	✓	✓	NMI	INTLVD	LVD Interrupt	*[IANIC00]*	*[IMNFLGNMI]* <INT000FLG>
					INTWDT0	WDT Interrupt	*[IBNIC00]*	*[IMNFLGNMI]* <INT016FLG>
✓	✓	✓	✓	0	INT00	External interrupt 00a	*[IAIMC00]*	*[IMNFLG1]* <INT032FLG>
✓	-	-	-			External interrupt 00b	*[IAIMC32]*	*[IMNFLG2]* <INT064FLG>
✓	✓	✓	✓	1	INT01	External interrupt 01a	*[IAIMC01]*	*[IMNFLG1]* <INT033FLG>
✓	-	-	-			External interrupt 01b	*[IAIMC33]*	*[IMNFLG2]* <INT065FLG>
✓	✓	✓	✓	2	INT02	External interrupt 02a	*[IAIMC02]*	*[IMNFLG1]* <INT034FLG>
✓	✓	✓	-			External interrupt 02b	*[IAIMC34]*	*[IMNFLG2]* <INT066FLG>
✓	✓	✓	✓	3	INT03	External interrupt 03a	*[IAIMC03]*	*[IMNFLG1]* <INT035FLG>
✓	✓	✓	-			External interrupt 03b	*[IAIMC35]*	*[IMNFLG2]* <INT067FLG>
✓	✓	✓	✓	4	INT04	External interrupt 04	*[IBIMC033]*	*[IMNFLG4]* <INT129FLG>
✓	✓	✓	✓	5	INT05	External interrupt 05	*[IBIMC034]*	*[IMNFLG4]* <INT130FLG>
✓	✓	✓	-	6	INT06	External interrupt 06	*[IBIMC035]*	*[IMNFLG4]* <INT131FLG>
✓	✓	✓	-	7	INT07	External interrupt 07a	*[IBIMC036]*	*[IMNFLG4]* <INT132FLG>
✓	-	-	-			External interrupt 07b	*[IBIMC040]*	*[IMNFLG4]* <INT136FLG>
✓	✓	✓	-	8	INT08	External interrupt 08	*[IBIMC037]*	*[IMNFLG4]* <INT133FLG>
✓	✓	-	-	9	INT09	External interrupt 09	*[IBIMC038]*	*[IMNFLG4]* <INT134FLG>
✓	-	-	-	10	INT10	External interrupt 10	*[IBIMC039]*	*[IMNFLG4]* <INT135FLG>
✓	✓	✓	✓	11	INTVCN0	A-VE+ ch0 Schedule end interrupt		
✓	✓	✓	✓	12	INTVCT0	A-VE+ ch0 Task end interrupt		
✓	✓	✓	✓	13	INTEMG0	A-PMD ch0 EMG interrupt		
✓	✓	✓	✓	14	INTEMG1	A-PMD ch1 EMG interrupt		
✓	✓	✓	✓	15	INTOVV0	A-PMD ch0 OVV interrupt		
✓	✓	✓	✓	16	INTOVV1	A-PMD ch1 OVV interrupt		
✓	✓	✓	✓	17	INTPWM0	A-PMD ch0 PWM interrupt		
✓	✓	✓	✓	18	INTPWM1	A-PMD ch1 PWM interrupt		
✓	✓	✓	✓	19	INTENC00	A-ENC ch0 Encoder input interrupt 0		
✓	✓	✓	✓	20	INTENC01	A-ENC ch0 Encoder input interrupt 1		
✓	✓	✓	✓	21	INTADAPDA	ADC unit A PMD trigger interrupt A		
✓	✓	✓	✓	22	INTADAPDB	ADC unit A PMD trigger interrupt B		
✓	✓	✓	✓	23	INTADAPDC	ADC unit A PMD trigger interrupt C		
✓	✓	✓	✓	24	INTADAPDD	ADC unit A PMD trigger interrupt D		
✓	✓	✓	✓	25	INTADAPFLG	ADC unit A Priority interrupt		
✓	✓	✓	✓	26	INTADACP0	ADC unit A Monitor function 0 interrupt		
✓	✓	✓	✓	27	INTADACP1	ADC unit A Monitor function 1 interrupt		

M4K4	M4K2	M4K1	M4K0	Interrupt No	Interrupt Source	Interrupt Request	Interrupt Control Register	Interrupt Monitor Register
✓	✓	✓	✓	28	INTADATRG	ADC unit A General purpose trigger interrupt		
✓	✓	✓	✓	29	INTADASGL	ADC unit A Single conversion interrupt		
✓	✓	✓	✓	30	INTADACNT	ADC unit A Continuous conversion interrupt		
✓	✓	✓	✓	31	INTSC0RX	TSPI ch0 Receive interrupt UART ch0 Reception interrupt		
✓	✓	✓	✓	32	INTSC0TX	TSPI ch0 Transmit interrupt UART ch0 Transmission interrupt		
✓	✓	✓	✓	33	INTSC0ERR	TSPI ch0 Error interrupt UART ch0 Error interrupt		
✓	✓	-	-	34	INTSC1RX	TSPI ch1 Receive interrupt UART ch1 Reception interrupt		
✓	✓	-	-	35	INTSC1TX	TSPI ch1 Transmit interrupt UART ch1 Transmission interrupt		
✓	✓	-	-	36	INTSC1ERR	TSPI ch1 Error interrupt UART ch1 Error interrupt		
✓	✓	✓	✓	37	INTSC2RX	TSPI ch2 Receive interrupt UART ch2 Reception interrupt		
✓	✓	✓	✓	38	INTSC2TX	TSPI ch2 Transmit interrupt UART ch2 Transmission interrupt		
✓	✓	✓	✓	39	INTSC2ERR	TSPI ch2 Error interrupt UART ch2 Error interrupt		
✓	-	-	-	40	INTSC3RX	TSPI ch3 Receive interrupt UART ch3 Reception interrupt		
✓	-	-	-	41	INTSC3TX	TSPI ch3 Transmit interrupt UART ch3 Transmission interrupt		
✓	-	-	-	42	INTSC3ERR	TSPI ch3 Error interrupt UART ch3 Error interrupt		
✓	✓	✓	-	43	INTI2C0	I^2C ch0 I^2C interrupt		
✓	✓	✓	-	44	INTI2C0AL	I^2C ch0 I^2C arbitration lost detection interrupt		
✓	✓	✓	-	45	INTI2C0BF	I^2C ch0 I^2C bus free detection interrupt		
✓	✓	✓	-	46	INTI2C0NA	I^2C ch0 I^2C NACK detection interrupt		
✓	✓	✓	✓	47	INTT32A0AC	T32A ch0 timer A/C match, overflow, and underflow		
✓	✓	✓	✓	48	INTT32A0ACCAP0	T32A ch0 timer A/C capture 0		
✓	✓	✓	✓	49	INTT32A0ACCAP1	T32A ch0 timer A/C capture 1		
✓	✓	✓	✓	50	INTT32A0B	T32A ch0 timer B match, overflow, and underflow		
✓	✓	✓	✓	51	INTT32A0BCAP0	T32A ch0 timer B capture 0		
✓	✓	✓	✓	52	INTT32A0BCAP1	T32A ch0 timer B capture 1		
✓	✓	✓	✓	53	INTT32A1AC	T32A ch1 timer A/C match, Overflow, and underflow		
✓	✓	✓	✓	54	INTT32A1ACCAP0	T32A ch1 timer A/C capture 0		
✓	✓	✓	✓	55	INTT32A1ACCAP1	T32A ch1 timer A/C capture 1		
✓	✓	✓	✓	56	INTT32A1B	T32A ch1 timer B match, overflow, and underflow		
✓	✓	✓	✓	57	INTT32A1BCAP0	T32A ch1 timer B capture 0		
✓	✓	✓	✓	58	INTT32A1BCAP1	T32A ch1 timer B capture 1		
✓	✓	✓	✓	59	INTT32A2AC	T32A ch2 timer A/C match, overflow, and underflow		
✓	✓	✓	✓	60	INTT32A2ACCAP0	T32A ch2 timer A/C capture 0		
✓	✓	✓	✓	61	INTT32A2ACCAP1	T32A ch2 timer A/C capture 1		
✓	✓	✓	✓	62	INTT32A2B	T32A ch2 timer B match, overflow, and underflow		
✓	✓	✓	✓	63	INTT32A2BCAP0	T32A ch2 timer B capture 0		
✓	✓	✓	✓	64	INTT32A2BCAP1	T32A ch2 timer B capture 1		
✓	✓	✓	✓	65	INTT32A3AC	T32A ch3 timer A/C match, overflow, and underflow		
✓	✓	✓	✓	66	INTT32A3ACCAP0	T32A ch3 timer A/C capture 0		

M4K4	M4K2	M4K1	M4K0	Interrupt No	Interrupt Source	Interrupt Request	Interrupt Control Register	Interrupt Monitor Register
✓	✓	✓	✓	67	INTT32A3ACCAP1	T32A ch3 timer A/C capture 1		
✓	✓	✓	✓	68	INTT32A3B	T32A ch3 timer B match, overflow, and underflow		
✓	✓	✓	✓	69	INTT32A3BCAP0	T32A ch3 timer B capture 0		
✓	✓	✓	✓	70	INTT32A3BCAP1	T32A ch3 timer B capture 1		
✓	✓	✓	✓	71	INTT32A4AC	T32A ch4 timer A/C match, overflow, and underflow		
✓	✓	✓	✓	72	INTT32A4ACCAP0	T32A ch4 timer A/C capture 0		
✓	✓	✓	✓	73	INTT32A4ACCAP1	T32A ch4 timer A/C capture 1		
✓	✓	✓	✓	74	INTT32A4B	T32A ch4 timer B match, overflow, and underflow		
✓	✓	✓	✓	75	INTT32A4BCAP0	T32A ch4 timer B capture 0		
✓	✓	✓	✓	76	INTT32A4BCAP1	T32A ch4 timer B capture 1		
✓	✓	✓	✓	77	INTT32A5AC	T32A ch5 timer A/C match, overflow, and underflow		
✓	✓	✓	✓	78	INTT32A5ACCAP0	T32A ch5 timer A/C capture 0		
✓	✓	✓	✓	79	INTT32A5ACCAP1	T32A ch5 timer A/C capture 1		
✓	✓	✓	✓	80	INTT32A5B	T32A ch5 timer B match, overflow, and underflow		
✓	✓	✓	✓	81	INTT32A5BCAP0	T32A ch5 timer B capture 0		
✓	✓	✓	✓	82	INTT32A5BCAP1	T32A ch5 timer B capture 1		
✓	✓	✓	✓	83	INTPARI	RAMP RAM Parity interrupt		
✓	✓	✓	✓	84	INTDMAATC	DMAC unit A transmission end interrupt (ch0 to 31)	***[IBIMC000]*** to ***[IBIMC031]***	***[IMNFLG3]*** <INT096FLG> to <INT127FLG>
✓	✓	✓	✓	85	INTDMAAERR	DMAC unit A transmission error interrupt	***[IBIMC032]***	***[IMNFLG4]*** <INT128FLG>
✓	✓	✓	✓	87	INTFLCRDY	Code FLASH Ready interrupt		

Note: ✓: Available, -: N/A

7. Revision History

Table 7.1 Revision History

Revision	Date	Description
1.0	2017-11-06	New Release
1.1	2018-09-11	- "Conventions" Modified explanation of Trademark - "Terms and Abbreviations" Added NVIC Deleted TRGSEL and TRM - "4.3. Interrupt Request" Figure 4.1: Added Interrupt Monitor Register Table 4.1: "Interrupt from Voltage Detection Circuit"→"LVD interrupt" "Interrupt by Watchdog Timer"→"WDT interrupt" "DMAC transfer end"→"DMAC transmission end interrupt" "DMAC transfer error"→"DMAC transmission error interrupt" - "4.4.List of Interrupt Sources" Table 4.2 to Table 4.5, Table 4.7: Modified row of Interrupt Request Table 4.7 Interrupt Monitor Register row: INT96FLG→INT096FLG Table 4.8 Interrupt Request: "External interrupt pin"→"External interrupt" Table 4.9 Interrupt Source: "DMAC transfer end"→ "DMAC unit A transmission end interrupt" Interrupt Monitor Register row: INT99FLG, INT98FLG, INT97FLG, INT96FLG → INT099FLG, INT098FLG, INT097FLG, INT096FLG - "4.5.Interrupt detection level" Modified explanation - "4.6.Interrupt Handling" "4.6.1." Details of 1st term: "low power consumption mode" →"Interrupt Control Registers" 2nd and 3rd terms: "interrupt signal"→" interrupt request" "4.6.3." Modified explanation of 2nd stage - "5.1.Register List" Deleted (Note2) under "Interrupt Control Registers A", "Interrupt Control Registers B", "Reset Flag Registers" "5.5.4." Bit Symbol row: INT99FLG, INT98FLG, INT97FLG, INT96FLG→ INT099FLG, INT098FLG, INT097FLG, INT096FLG - "6.1.TMPM4K4/TMPM4K2/TMPM4K1/TMPM4K0" Modified Interrupt Request row of table Interrupt Monitor Register row: INT96FLG→INT096FLG - "RESTRICTIONS ON PRODUCT USE" update

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.**

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

https://toshiba.semicon-storage.com/