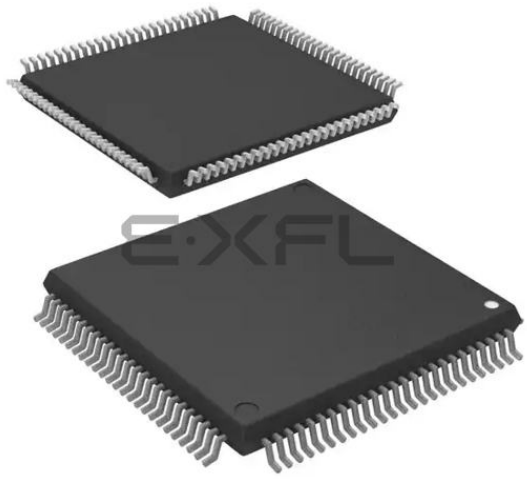


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

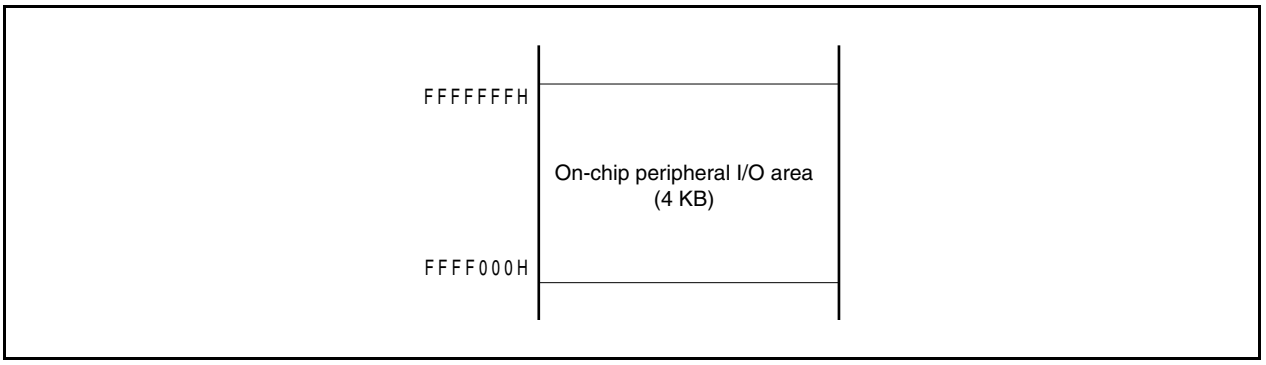
Details	
Product Status	Obsolete
Core Processor	V850E1
Core Size	32-Bit Single-Core
Speed	100MHz
Connectivity	CSI, I²C, UART/USART
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	55
Program Memory Size	480KB (480K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	24K x 8
Voltage - Supply (Vcc/Vdd)	1.35V ~ 1.65V
Data Converters	A/D 12x10b, 7x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd70f3915gf-r-gas-ax

(3) On-chip peripheral I/O area

4 KB of memory, addresses FFFF000H to FFFFFFFH, is provided as an on-chip peripheral I/O area.
An image of addresses FFFF000H to FFFFFFFH can be seen at addresses 3FFF000H to 3FFFFFFH[Note].

Note Addresses 3FFF000H to 3FFFFFFH are access-prohibited. To access the on-chip peripheral I/O, specify addresses FFFF000H to FFFFFFFH.

Figure 3-9. On-Chip Peripheral I/O Area



On-chip peripheral I/O registers associated with the operating mode specification and the state monitoring for the on-chip peripheral I/O are all memory-mapped to the on-chip peripheral I/O area. Program fetches cannot be executed from this area.

Cautions 1. In the V850E/IG4 and V850E/IH4, if a register is word accessed, halfword access is performed twice in the order of lower address, then higher address of the word area, disregarding the lower 2 bits of the address.

2. For registers in which byte access is possible, if halfword access is executed, the higher 8 bits become undefined during the read operation, and the lower 8 bits of data are written to the register during the write operation.

3. Addresses that are not defined as registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.
Addresses 3FFF000H to 3FFFFFFH cannot be specified as the source/destination address of DMA transfer. Be sure to use addresses FFFF000H to FFFFFFFH for the source/destination address of DMA transfer.

(7/18)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF1FEH	Power save control register	PSC	R/W	√	√		00H
FFFFF200H	A/D0 conversion result register 0	AD0CR0	R			√	0000H
FFFFF201H	A/D0 conversion result register 0H	AD0CR0H			√		00H
FFFFF202H	A/D0 conversion result register 1	AD0CR1				√	0000H
FFFFF203H	A/D0 conversion result register 1H	AD0CR1H			√		00H
FFFFF204H	A/D0 conversion result register 2	AD0CR2				√	0000H
FFFFF205H	A/D0 conversion result register 2H	AD0CR2H			√		00H
FFFFF206H	A/D0 conversion result register 3	AD0CR3				√	0000H
FFFFF207H	A/D0 conversion result register 3H	AD0CR3H			√		00H
FFFFF208H	A/D0 conversion result register 4	AD0CR4				√	0000H
FFFFF209H	A/D0 conversion result register 4H	AD0CR4H			√		00H
FFFFF20AH	A/D0 conversion result register 5	AD0CR5				√	0000H
FFFFF20BH	A/D0 conversion result register 5H	AD0CR5H			√		00H
FFFFF20CH	A/D0 conversion result register 6	AD0CR6				√	0000H
FFFFF20DH	A/D0 conversion result register 6H	AD0CR6H			√		00H
FFFFF20EH	A/D0 conversion result register 7	AD0CR7				√	0000H
FFFFF20FH	A/D0 conversion result register 7H	AD0CR7H			√		00H
FFFFF210H	A/D0 conversion result register 8	AD0CR8				√	0000H
FFFFF211H	A/D0 conversion result register 8H	AD0CR8H			√		00H
FFFFF212H	A/D0 conversion result register 9	AD0CR9				√	0000H
FFFFF213H	A/D0 conversion result register 9H	AD0CR9H			√		00H
FFFFF214H	A/D0 conversion result register 10	AD0CR10				√	0000H
FFFFF215H	A/D0 conversion result register 10H	AD0CR10H			√		00H
FFFFF216H	A/D0 conversion result register 11	AD0CR11				√	
FFFFF217H	A/D0 conversion result register 11H	AD0CR11H			√		00H
FFFFF218H	A/D0 conversion result register 12	AD0CR12				√	0000H
FFFFF219H	A/D0 conversion result register 12H	AD0CR12H			√		00H
FFFFF21AH	A/D0 conversion result register 13	AD0CR13				√	0000H
FFFFF21BH	A/D0 conversion result register 13H	AD0CR13H			√		00H
FFFFF21CH	A/D0 conversion result register 14	AD0CR14				√	0000H
FFFFF21DH	A/D0 conversion result register 14H	AD0CR14H			√		00H
FFFFF21EH	A/D0 conversion result register 15	AD0CR15				√	0000H
FFFFF21FH	A/D0 conversion result register 15H	AD0CR15H			√		00H
FFFFF220H	A/D converter 0 scan mode register	AD0SCM	R/W			√	0000H
FFFFF220H	A/D converter 0 scan mode register L	AD0SCML		√	√		00H
FFFFF221H	A/D converter 0 scan mode register H	AD0SCMH		√	√		00H
FFFFF222H	A/D converter 0 conversion time control register	AD0CTC		√	√		00H
FFFFF224H	A/D converter 0 conversion channel specification register	AD0CHEN				√	0000H
FFFFF224H	A/D converter 0 conversion channel specification register L	AD0CHENL		√	√		00H
FFFFF225H	A/D converter 0 conversion channel specification register H	AD0CHENH		√	√		00H

(1) Registers

(a) Port 0 register (P0)

After reset: Undefined R/W Address: FFFFF400H

	7	6	5	4	3	2	1	0
P0	P07	P06	P05	P04	P03	P02	P01	P00

P0n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Remark n = 0 to 7

(b) Port 0 mode register (PM0)

After reset: FFH R/W Address: FFFFF420H

	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Remark n = 0 to 7

(c) Pull-up resistor option register 9 (PU9)

After reset: 00H R/W Address: FFFFFC52H

	7	6	5	4	3	2	1	0
PU9	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90

PU9n	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note]

Note An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark n = 0 to 7

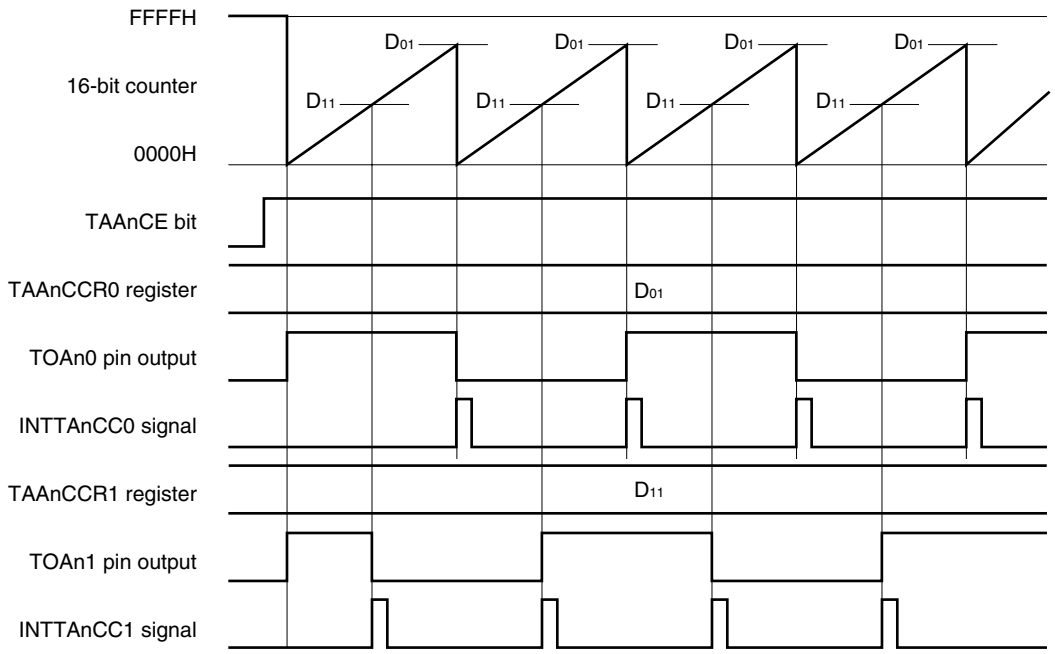
When the TAAnCCR1 register is set to the same value as the TAAnCCR0 register, the INTTAnCC0 signal is generated at the same timing as the INTTAnCC1 signal and the TOAn1 pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOAn1 pin.

The following shows the operation when the TAAnCCR1 register is set to other than the value set in the TAAnCCR0 register.

If the set value of the TAAnCCR1 register is less than the set value of the TAAnCCR0 register, the INTTAnCC1 signal is generated once per cycle. At the same time, the output of the TOAn1 pin is inverted.

The TOAn1 pin outputs a PWM waveform with a duty factor of 50% after outputting a short-width pulse.

Figure 6-13. Timing Chart When $D_{01} \geq D_{11}$

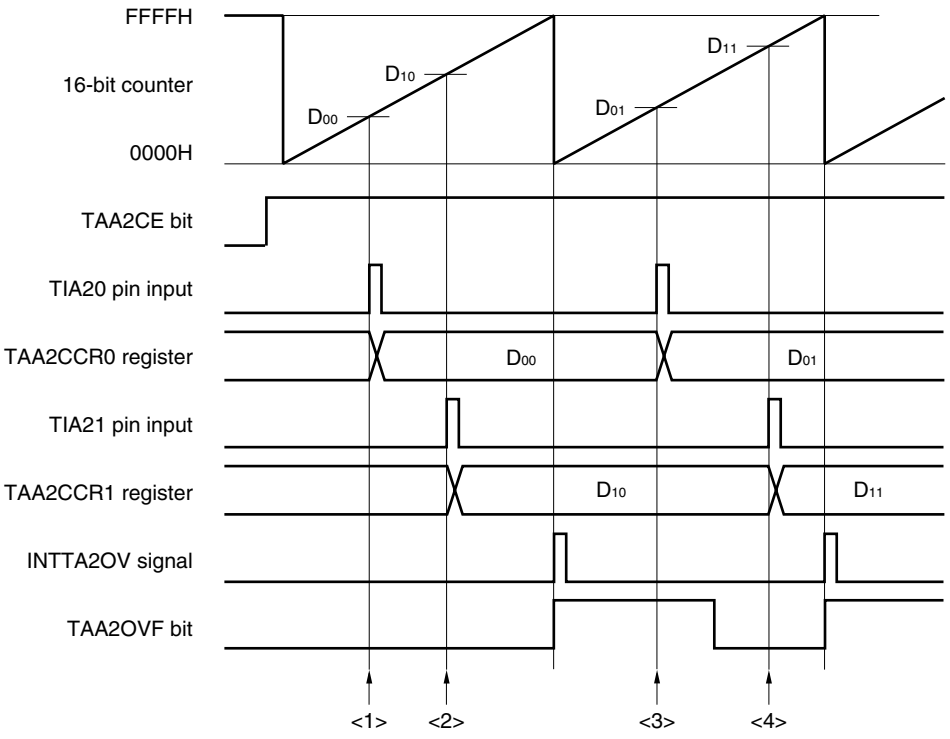


Remark n = 0 to 2

(c) Processing of overflow when two capture registers are used

Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.

Example of incorrect processing when two capture registers are used



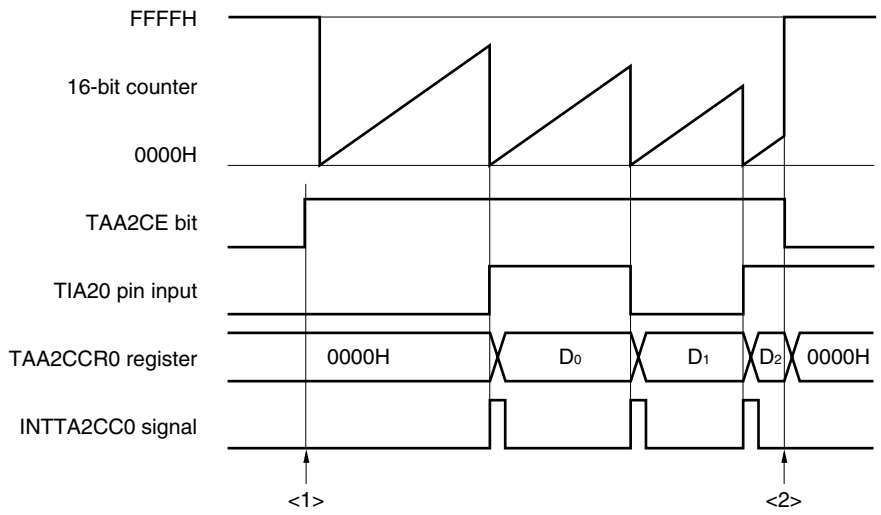
The following problem may occur when two pulse widths are measured in the free-running timer mode.

<1> Read the TAA2CCR0 register (setting of the default value of the TIA20 pin input).
<2> Read the TAA2CCR1 register (setting of the default value of the TIA21 pin input).
<3> Read the TAA2CCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).
<4> Read the TAA2CCR1 register.
Read the overflow flag. Because the flag is cleared in <3>, 0 is read.
Because the overflow flag is 0, the pulse width can be calculated by (D_{11} − D_{10}) (incorrect).

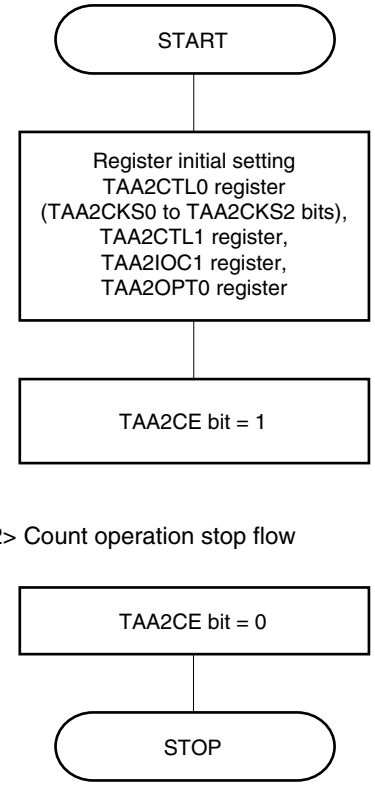
When two capture registers are used, and if the overflow flag is cleared to 0 by one capture register, the other capture register may not obtain the correct pulse width.
Use software when using two capture registers. An example of how to use software is shown below.

(1) Operation flow in pulse width measurement mode

Figure 6-47. Software Processing Flow in Pulse Width Measurement Mode



<1> Count operation start flow



Initial setting of these registers is performed before setting the TAA2CE bit to 1.

The TAA2CKS0 to TAA2CKS2 bits can be set at the same time when counting has been started (TAA2CE bit = 1).

<2> Count operation stop flow

The counter is initialized and counting is stopped by clearing the TAA2CE bit to 0.

- INTTBnCC2 interrupt: This signal functions as a match interrupt request signal of the CCR2 buffer register and as a capture interrupt request signal to the TABnCCR2 register.
- INTTBnCC3 interrupt: This signal functions as a match interrupt request signal of the CCR3 buffer register and as a capture interrupt request signal to the TABnCCR3 register.
- INTTBnOV interrupt: This signal functions as an overflow interrupt request signal.

(b) Batch write

In this mode, data is transferred all at once from the TABnCCR0 to TABnCCR3 registers to the CCR0 to CCR3 buffer registers during timer operation. This data is transferred upon a match between the value of the CCR0 buffer register and the value of the 16-bit counter. Transfer is enabled by writing to the TABnCCR1 register.

Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TABnCCR1 register.

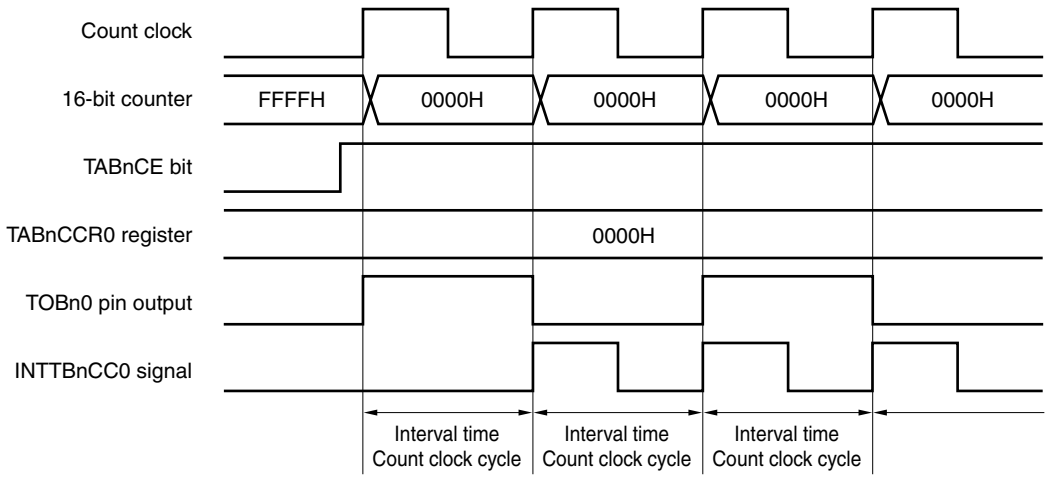
In order for the set value when the TABnCCR0 to TABnCCR3 registers are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be transferred to the CCR0 to CCR3 buffer registers), it is necessary to rewrite TABnCCR0 and finally write to the TABnCCR1 register before the 16-bit counter value and the CCR0 buffer register value match. The values of the TABnCCR0 to TABnCCR3 registers are transferred to the CCR0 to CCR3 buffer registers upon a match between the count value of the 16-bit counter and the value of the CCR0 buffer register. Thus, even when wishing only to rewrite the value of the TABnCCR0, TABnCCR2, or TABnCCR3 register, also write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

(2) Interval timer mode operation timing

(a) Operation if TABnCCR0 register is set to 0000H

If the TABnCCR0 register is set to 0000H, the INTTBnCC0 signal is generated at each count clock, and the output of the TOBn0 pin is inverted.

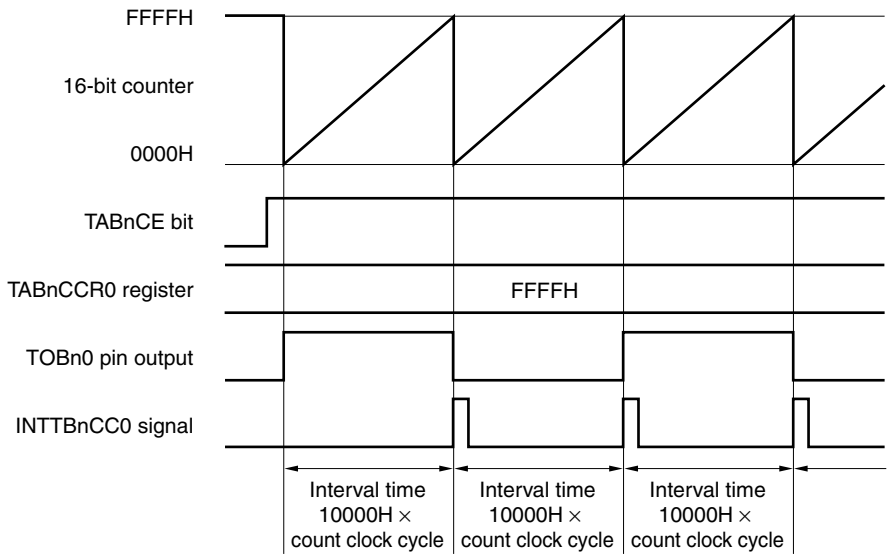
The value of the 16-bit counter is always 0000H.



Remark n = 0, 1

(b) Operation if TABnCCR0 register is set to FFFFH

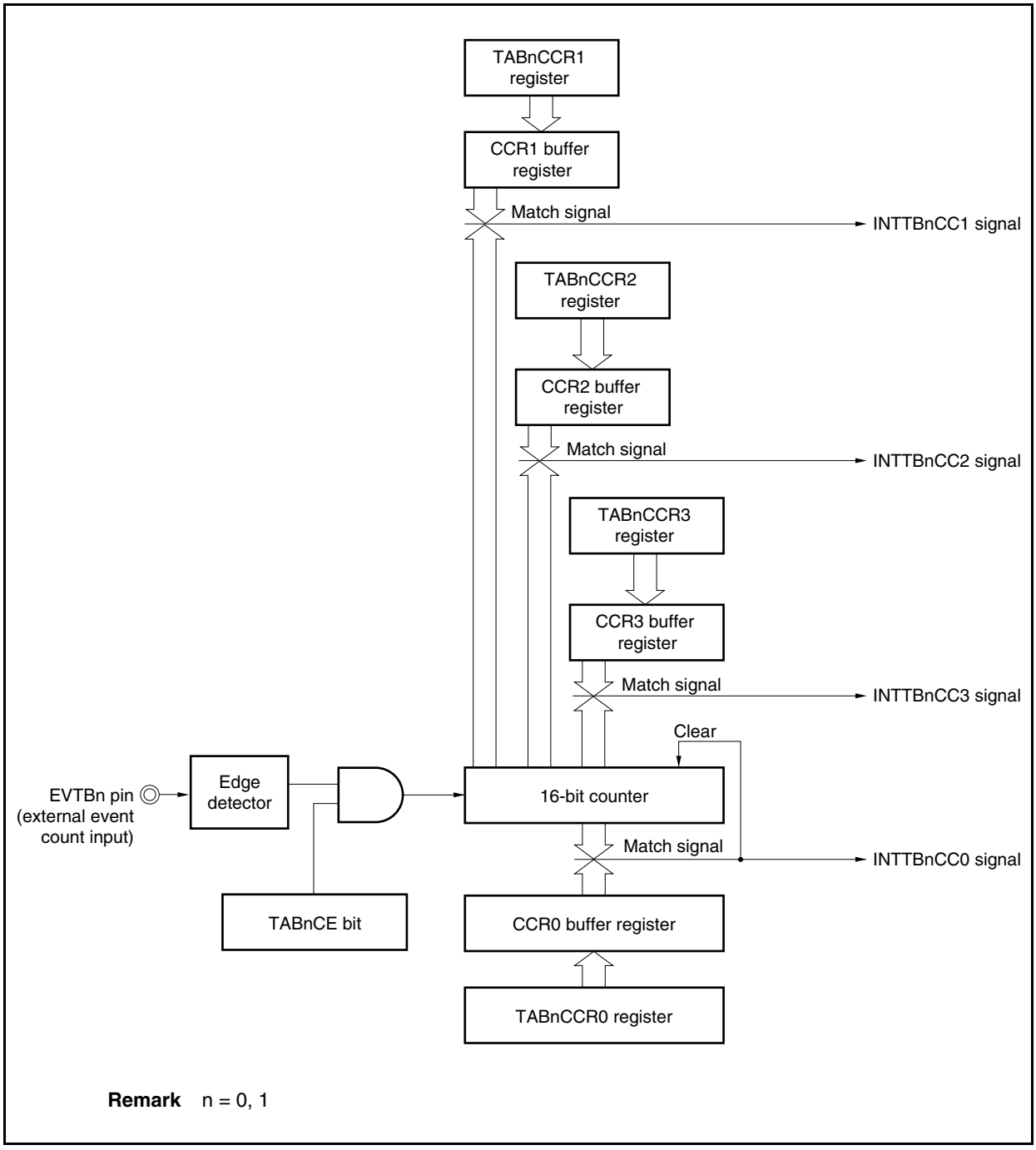
If the TABnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTBnCC0 signal is generated and the output of the TOBn0 pin is inverted. At this time, an overflow interrupt request signal (INTTBnOV) is not generated, nor is the overflow flag (TABnOPT0.TABnOVF bit) set to 1.



Remark n = 0, 1

(e) Operation of TABnCCR1 to TABnCCR3 registers

Figure 7-17. Configuration of TABnCCR1 to TABnCCR3 Registers



Remark n = 0, 1

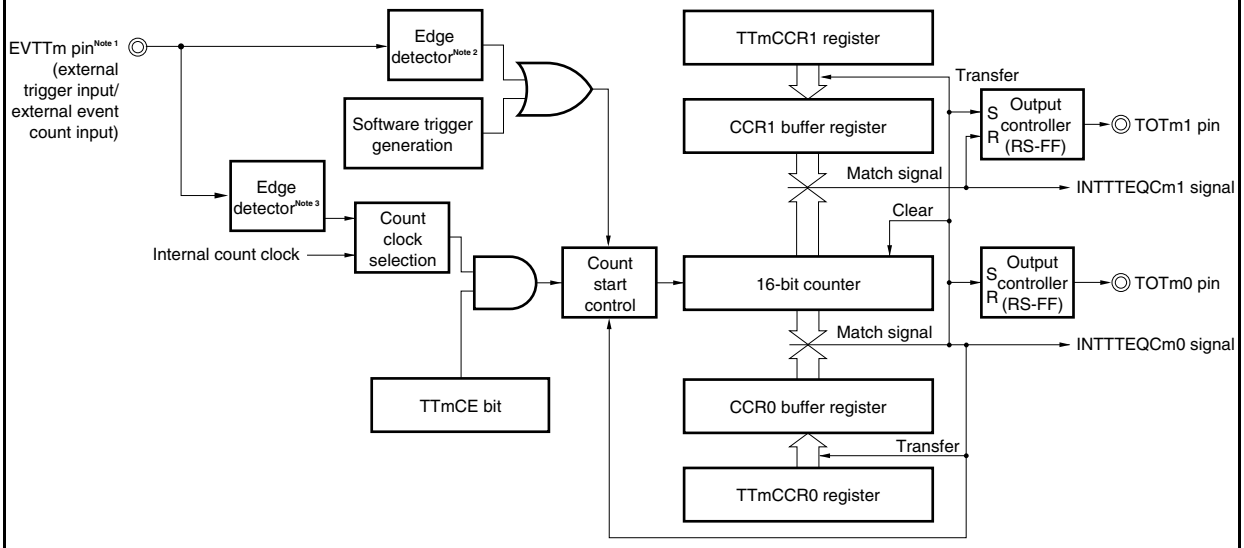
8.6.4 One-shot pulse output mode (TTnMD3 to TTnMD0 bits = 0011)

In the one-shot pulse output mode, 16-bit timer/event counter T waits for a trigger when the TTnCTL0.TTnCE bit is set to 1. When the valid edge of an external trigger input (the EVTTm pin in the case of TMT0 and TMT1, and the TITk0 pin in the case of TMT2 and TMT3) is detected, 16-bit timer/event counter T starts counting, and outputs a one-shot pulse from the TOTn1 pin.

In the case of TMT0 and TMT1, the TOTm0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger). Instead of the external trigger input, a software trigger can also be generated to output the pulse.

In the case of TMT2 and TMT3, instead of the external trigger input, a software trigger can also be generated to output the pulse. When the software trigger is used, the TOTk0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger).

Figure 8-27. Configuration of TMT0 and TMT1 in One-Shot Pulse Output Mode



Notes 1. Because the external trigger input pin (EVTTm) and external event count input pin (EVTTm) share the same alternate-function pin, the two functions cannot be used at the same time.

2. Edge detector for external trigger input.
Set by the TTmIOC2.TTmETS1 and TTmIOC2.TTmETS0 bits.

3. Edge detector for external event count input.
Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

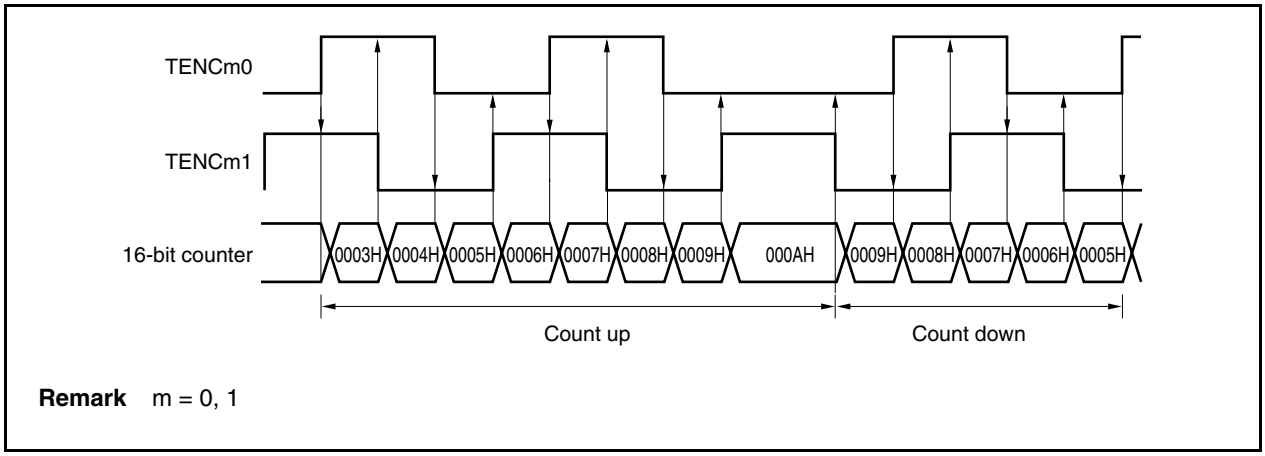
Remark m = 0, 1

- When TTmUDS1 and TTmUDS0 bits = 11

TENCm0 Pin	TENCm1 Pin	Count Operation
Low level	Falling edge	Count down
Rising edge	Low level	
High level	Rising edge	
Falling edge	High level	
Rising edge		Count up
High level	Falling edge	
Falling edge	Low level	
Low level	Rising edge	
Simultaneous input to TENCm0 and TENCm1 pins		Counter does not perform count operation but holds value immediately before.

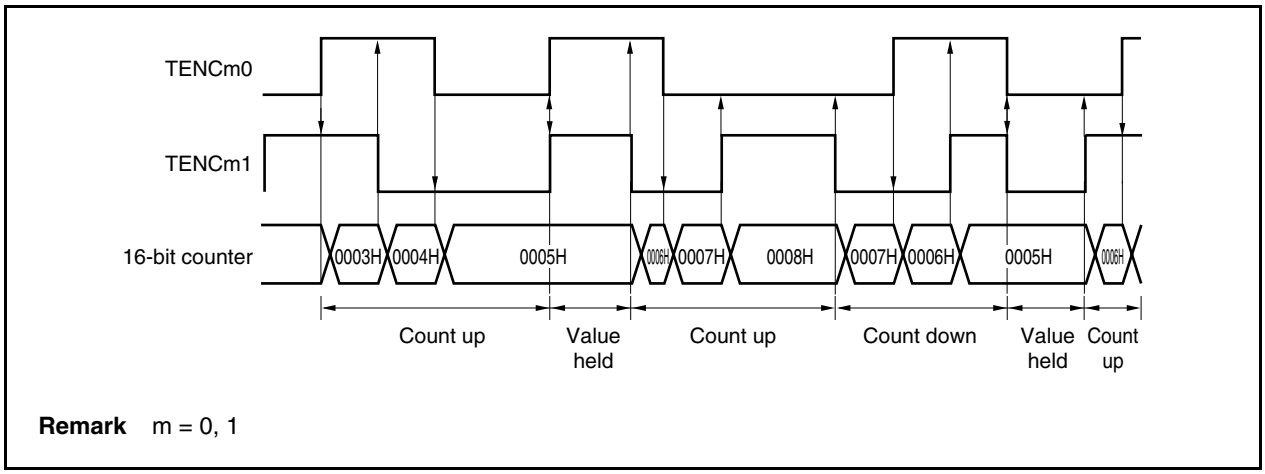
Caution Specification of the valid edge of the TENCm0 and TENCm1 pins is invalid.

Figure 8-58. Operation Example (Count Operation When Valid Edges of TENCm0 and TENCm1 Pins Do Not Overlap)



Remark m = 0, 1

Figure 8-59. Operation Example (Count Operation When Valid Edges of TENCm0 and TENCm1 Pins Overlap)



Remark m = 0, 1

15.7.5 Reception error

In the single mode (UBFIC0.UBMOD bit = 0), the three types of errors that can occur during a receive operation are a parity error, framing error, and overrun error. In the FIFO mode (UBFIC0.UBMOD bit = 1), the three types of errors that can occur during a receive operation are a parity error, framing error, and overflow error.

As a result of data reception, the UBSTR.UBPE, UBSTR.UBFE, or UBSTR.UBOVE bit is set to 1 if a parity error, framing error, or overrun error occurs in the single mode. The UBSTR.UBOVF bit is set to 1 if an overflow error occurs in the FIFO mode. The UBRXAP.UBPEF or UBRXAP.UBFEF bit is set to 1 if a parity error or framing error occurs in the FIFO mode. At the same time, a reception error interrupt request signal (INTUBTIRE) occurs. The contents of the error can be detected by reading the contents of the UBSTR or UBRXAP register.

The contents of the UBSTR register are reset when 0 is written to the UBOVF, UBPE, UBFE, or UBOVE bit, or the UBCTL0.UBPWR or UBCTL0.UBRXE bit. The contents of the UBRXAP register are reset when 0 is written to the UBCTL0.UBPWR bit.

Table 15-5. Reception Error Causes

Error Flag	Valid Operation Mode	Error Flag	Reception Error	Cause
UBPE	Single mode	UBPE	Parity error	The parity specification during transmission does not match the parity of the receive data
UBFE		UBFE	Framing error	No stop bit detected
UBOVE		UBOVE	Overrun error	The reception of the next data is ended before data is read from the UBRX register
UBOVF	FIFO mode	UBOVF	Overflow error	The reception of the next data is ended while receive FIFO is full and before data is read.
UBPEF		UBPEF	Parity error	The parity specification during transmission does not match the parity of the data to be received.
UBFEF		UBFEF	Framing error	The stop bit is not detected when the target data is loaded.

16.3.2 Mode switching between CSIF1 and UARTA2

In the V850E/IG4 and V850E/IH4, CSIF1 and UARTA2 share a pin, and these functions cannot be used at the same time. To use the pin for the CSIF1 function, set up the PMC3 and PFC3 registers in advance.

Switching the operation mode between CSIF1 and UARTA2, the serial interfaces, is described below.

Caution The operations related to transmission and reception of CSIF1 or UARTA2 are not guaranteed if the operation mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 16-3. Operation Mode Switch Settings of CSIF1 and UARTA2

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

PMC34	PFC34	Operation mode
0	×	Port I/O mode
1	0	$\overline{\text{SCKF1}}$

PMC3n	PFC3n	Operation mode
0	×	Port I/O mode
1	0	CSIF1 mode
1	1	UARTA2 mode

Remarks 1. n = 2, 3

2. × = 0 or 1

(2) IIC status register 0 (IICS0)

The IICS0 register indicates the status of the I²C bus.

The IICS0 register is read-only, in 8-bit or 1-bit units.

However, the IICS0 register can only be read when the IICC0.STT0 bit is 1 or during the wait period.

Reset sets this register to 00H.

(1/3)

After reset: 00H R Address: FFFFFD86H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master device status
0	Slave device status or communication standby status
1	Master device communication status

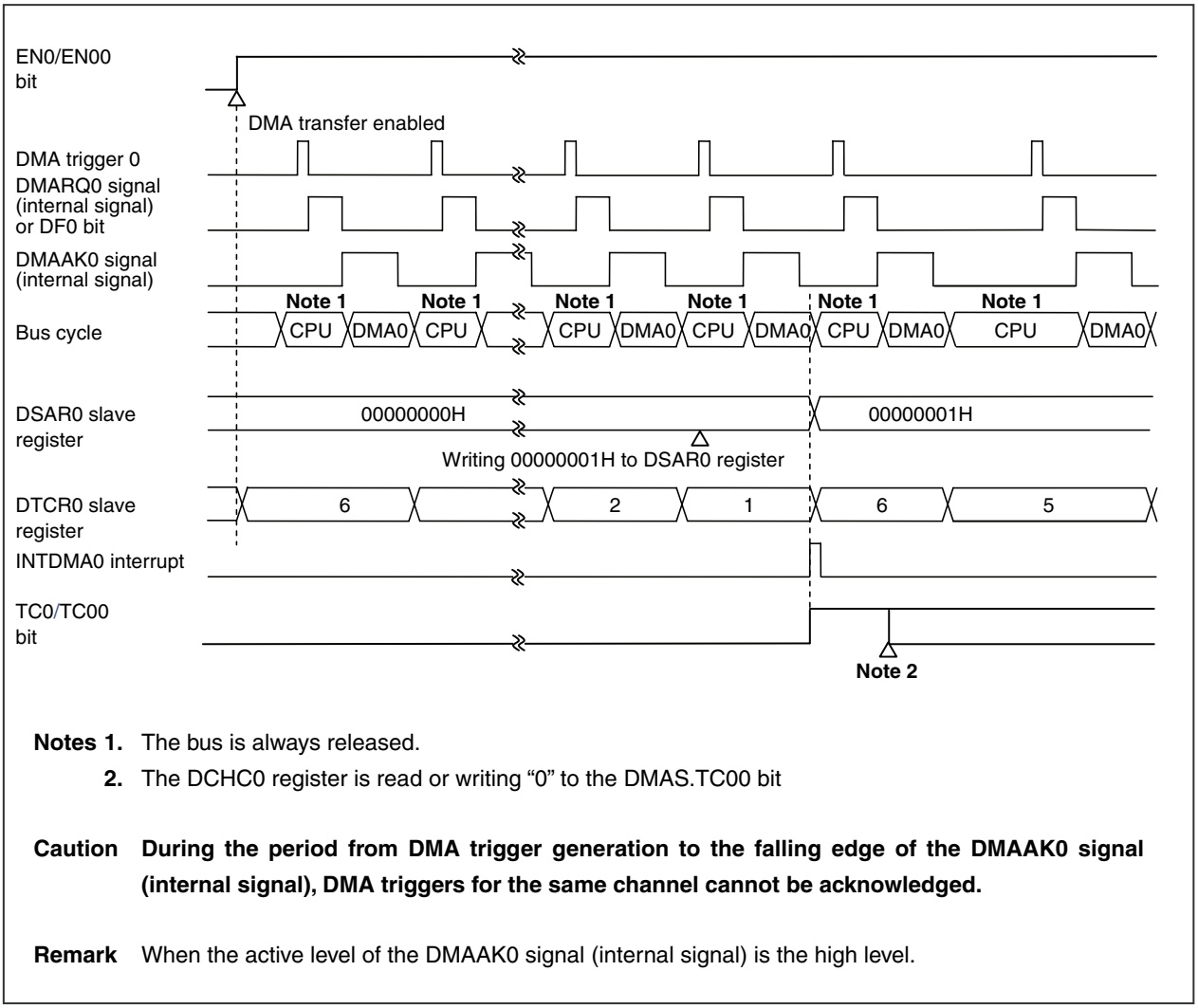
Condition for clearing (MSTS0 bit = 0)	Condition for setting (MSTS0 bit = 1)
• When a stop condition is detected • When the ALD0 bit = 1 (arbitration loss) • Cleared by the IICC0.LREL0 bit = 1 (exit from communications) • When the IICC0.IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD0	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. The MSTS0 bit is cleared to 0.

Condition for clearing (ALD0 bit = 0)	Condition for setting (ALD0 bit = 1)
• Automatically cleared after the IICS0 register is read[Note] • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Note This bit is also cleared when a bit manipulation instruction is executed for another bit in the IICS0 register.

Figure 18-4. Single Transfer Example 3 (with Next Address Setting Function Enabled)



Notes 1. The bus is always released.

2. The DCHC0 register is read or writing “0” to the DMAS.TC00 bit

Caution During the period from DMA trigger generation to the falling edge of the DMAAK0 signal (internal signal), DMA triggers for the same channel cannot be acknowledged.

Remark When the active level of the DMAAK0 signal (internal signal) is the high level.

18.14 Cautions

(1) Memory boundary

Transfer is not guaranteed if the transfer source or the transfer destination address exceeds the DMA area (internal RAM or on-chip peripheral I/O) during DMA transfer.

(2) Transfer of misaligned data

32-bit or 16-bit misaligned data cannot be transferred by DMA.

(3) Bus arbitration for CPU

Because the DMAC takes precedence over the CPU in acquiring bus mastership, if the CPU requests access during DMA transfer, the CPU will not be able to execute the access until the DMA transfer is completed and the bus is released to the CPU.

However, the CPU can access the internal ROM and internal RAM, as long as they are not being accessed by the DMAC.

The CPU can also access the internal ROM while the DMAC is executing DMA transfer between an on-chip peripheral I/O and the internal RAM.

(4) Program execution in internal RAM and DMA transfer

The CPU may deadlock under the following conditions. In this case, the only action that can be executed is a reset.

Conditions

A DMA transfer to transfer data to/from the internal RAM is executed while any of the following instructions is being executed.

- A bit manipulation instruction located in the internal RAM (SET1, CLR1, NOT1)
- A data access instruction that accesses a misaligned address located in the internal RAM

Measures this problem can be avoided by taking either of the following measures.

Measures

- Do not execute a bit manipulation instruction (SET1, CLR1, or NOT1) located in the internal RAM or a data access instruction that accesses a misaligned address when DMA transfer is being executed to transfer data to/from the internal RAM.
- Do not execute DMA transfer that transfers data to/from the internal RAM when a bit manipulation instruction (SET1, CLR1, or NOT1) located in the internal RAM or a data access instruction that accesses a misaligned address is being executed.

(5) Delay in start of DMA transfer

The start of DMA transfer may be delayed by an internal RAM access or on-chip peripheral I/O access by the CPU.

(6) Special register settings while using DMA

See **3.4.8 (1) Setting data to special registers**.

24.3.4 Cautions

(1) Handling of device that was used for debugging

Do not mount a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed. Moreover, do not embed the debug monitor program into mass-produced products.

(2) When breaks cannot be executed

Forced breaks cannot be executed if one of the following conditions is satisfied.

- Interrupts are disabled (DI)
- Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the V850E/IG4 or V850E/IH4, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the V850E/IG4 or V850E/IH4 is UARTA0, and the peripheral clock has been stopped

(3) When pseudo real-time RAM monitor (RRM) function and DMM function do not operate

The pseudo RRM function and DMM function do not operate if one of the following conditions is satisfied.

- Interrupts are disabled (DI)
- Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the V850E/IG4 or V850E/IH4, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the V850E/IG4 or V850E/IH4 is UARTA0, and the peripheral clock has been stopped
- Mode for communication between MINICUBE2 and the V850E/IG4 or V850E/IH4 is UARTA0, and a clock different from the one specified in the debugger is used for communication

(4) Standby release with pseudo RRM and DMM functions enabled

The standby mode is released by the pseudo RRM function and DMM function if one of the following conditions is satisfied.

- Mode for communication between MINICUBE2 and the V850E/IG4 or V850E/IH4 is CSIF0
- Mode for communication between MINICUBE2 and the V850E/IG4 or V850E/IH4 is UARTA0, and the peripheral clock has not stopped.

(5) Writing to peripheral I/O registers that requires a specific sequence, using DMM function

Peripheral I/O registers that requires a specific sequence cannot be written with the DMM function.

(6) Flash self programming

If a space where the debug monitor program is allocated is rewritten by flash self programming, the debugger can no longer operate normally.

26.1.9 Characteristics of A/D converter 2

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = V_{DD2} = 1.35 to 1.65 V,
EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V,
V_{SS0} = V_{SS1} = V_{SS2} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Note 1]					±4.0	LSB
Conversion time	t_{CONV}		3.00		10.00	μs
Zero scale error[Note 1]					±4.0	LSB
Full-scale error[Note 1]					±4.0	LSB
Integral linearity error[Note 1]					±4.0	LSB
Differential linearity error[Note 1]					±2.0	LSB
Analog reference voltage	AV_{REF}		4.0		5.5	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{DD}	V
AV_{DD} supply current	AI_{DD}	During operation		3.5	7	mA
	AI_{DDS}	In STOP mode[Note 2]		1	10	μA

Notes 1. Excludes quantization error (±0.5 LSB).

2. Stop A/D converter 2 (AD2M0.AD2CE bit = 0) before setting STOP mode.

Remark LSB: Least Significant Bit