Renesas Electronics America Inc - UPD70F3767GF-GAT-AX Datasheet

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	Not For New Designs
Core Processor	V850ES
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	CSI, EBI/EMI, I²C, UART/USART, USB
Peripherals	DMA, LVD, PWM, WDT
Number of I/O	96
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	56K x 8
Voltage - Supply (Vcc/Vdd)	2.85V ~ 3.6V
Data Converters	A/D 12x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	128-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd70f3767gf-gat-ax

6.4 Operation

6.4.1 Operation of each clock

The following table shows the operation status of each clock.

Table 6-1. Operation Status of Each Clock

Register Setting and Operation Status / Target Clock	PCC Register								
	CLK Bit = 0, MCK Bit = 0					CLS Bit = 1, MCK Bit = 0		CLS Bit = 1, MCK Bit = 1	
	During Reset	During Oscillation Stabilization Time Count	HALT Mode	IDLE1, IDLE2 Mode	STOP Mode	Subclock Mode	Sub-IDLE Mode	Subclock Mode	Sub-IDLE Mode
Main clock oscillator (f_X)	×	○	○	○	×	○	○	×	×
Subclock oscillator (f_{XT})	○	○	○	○	○	○	○	○	○
CPU clock (f_{CPU})	×	×	×	×	×	○	×	○	×
Internal system clock (f_{CLK})	×	×	○	×	×	○	×	○	×
Main clock (in PLL mode, f_{XX})	×	○Note	○	×	×	○	○	×	×
Peripheral clock (f_{XX} to f_{XX}/1,024)	×	×	○	×	×	○	×	×	×
WT clock (main)	×	○	○	○	×	○	○	×	×
WT clock (sub)	○	○	○	○	○	○	○	○	○
WDT2 clock (internal oscillation)	×	○	○	○	○	○	○	○	○
WDT2 clock (main)	×	×	○	×	×	○	×	×	×
WDT2 clock (sub)	○	○	○	○	○	○	○	○	○

Note Lockup time

Remark ○: Operable
×: Stopped

6.4.2 Clock output function

The clock output function is used to output the internal system clock (f_{CLK}) from the CLKOUT pin.

The internal system clock (f_{CLK}) is selected by using the PCC.CK3 to PCC.CK0 bits.

The CLKOUT pin functions alternately as the PCM1 pin and functions as a clock output pin if so specified by the control register of port CM.

The status of the CLKOUT pin is the same as the internal system clock in Table 6-1 and the pin can output the clock when it is in the operable status. It outputs a low level in the stopped status. However, the CLKOUT pin is in the port mode (PCM1 pin: input mode) after reset and until it is set in the output mode. Therefore, the status of the pin is Hi-Z.

7.5.7 Pulse width measurement mode (TAAnMD2 to TAAnMD0 bits = 110)

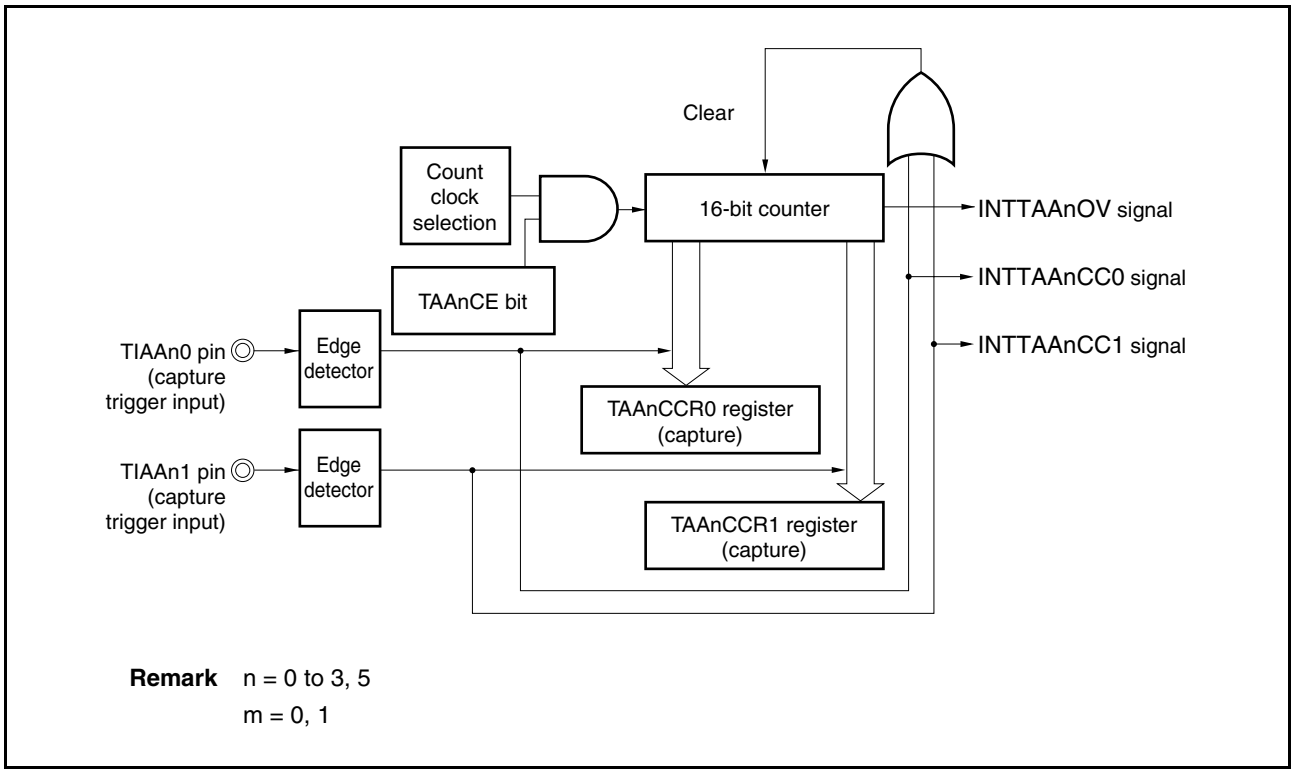
In the pulse width measurement mode, 16-bit timer/event counter AA starts counting when the TAAnCTL0.TAAnCE bit is set to 1. Each time the valid edge input to the TIAAnm pin has been detected, the count value of the 16-bit counter is stored in the TAAnCCRm register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TAAnCCRm register after a capture interrupt request signal (INTTAAnCCm) occurs.

Select either the TIAAn0 or TIAAn1 pin as the capture trigger input pin. Specify “No edge detection” for the unused pins by using the TAAnIOC1 register.

When an external clock is used as the count clock, measure the pulse width of the TIAAn1 pin because the external clock is fixed to the TIAAn0 pin. At this time, clear the TAAnIOC1.TAAnIS1 and TAAnIOC1.TAAnIS0 bits to 00 (capture trigger input (TIAAn0 pin): No edge detection).

Figure 7-39. Configuration in Pulse Width Measurement Mode

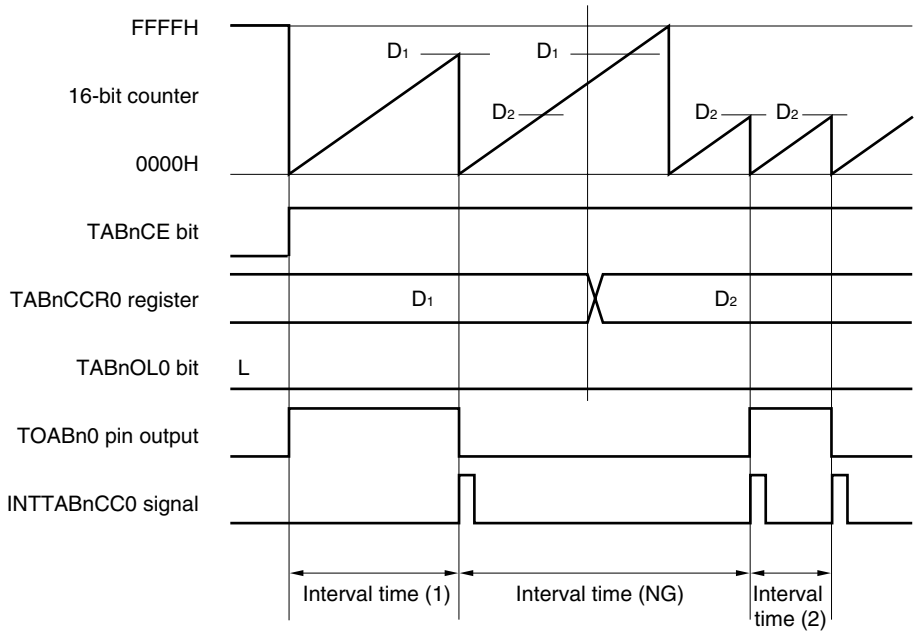


Remark n = 0 to 3, 5
m = 0, 1

(c) Notes on rewriting TABnCCR0 register

To change the value of the TABnCCR0 register to a smaller value, stop counting once and then change the set value.

If the value of the TABnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.



Remarks 1. **Interval time (1): $(D_1 + 1) \times$ Count clock cycle**

Interval time (NG): $(10000H + D_2 + 1) \times$ Count clock cycle

Interval time (2): $(D_2 + 1) \times$ Count clock cycle

2. **n = 0, 1**

If the value of the TABnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TABnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTABnCC0 signal is generated and the output of the TOABn0 pin is inverted.

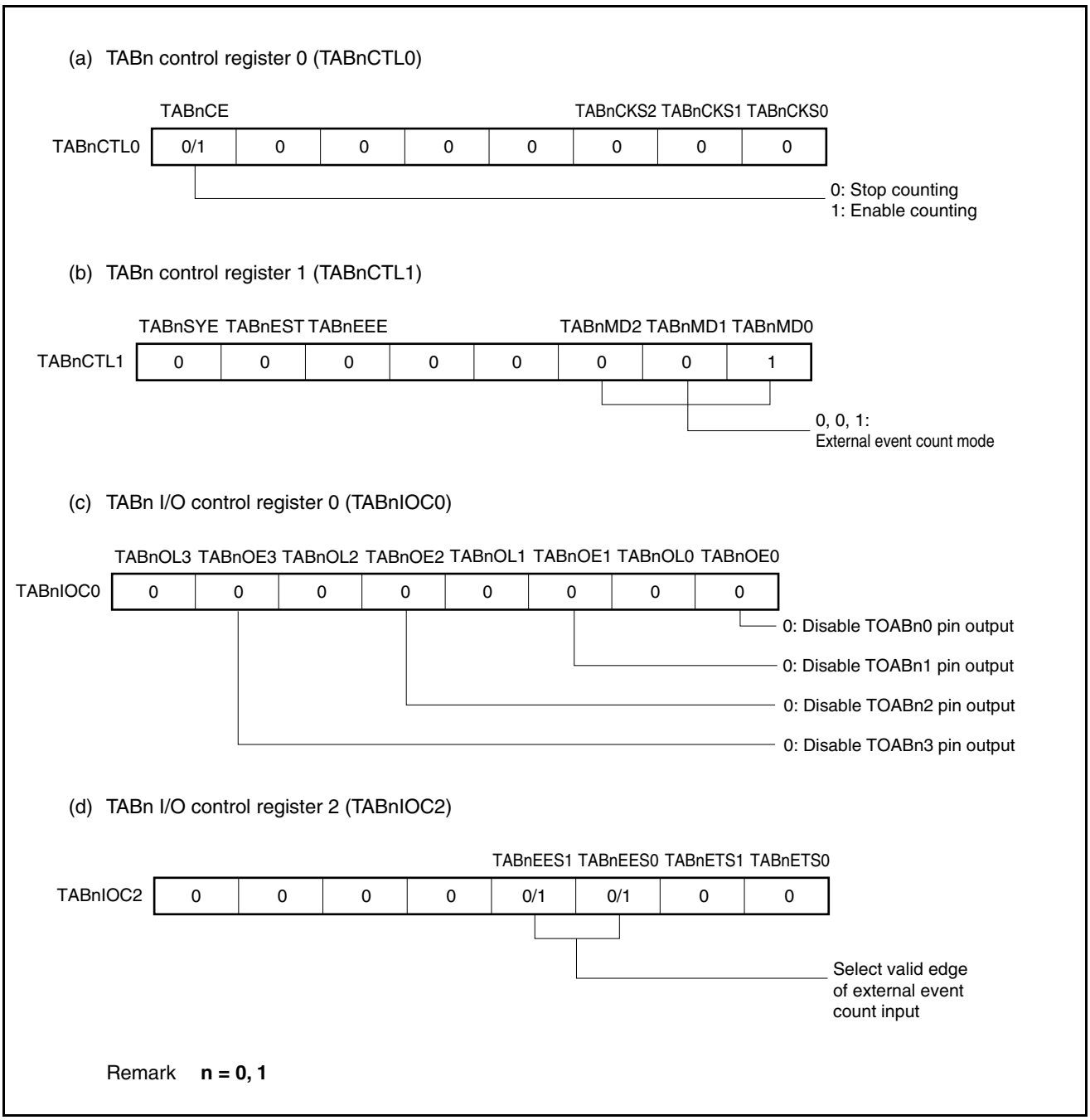
Therefore, the INTTABnCC0 signal may not be generated at the interval time "$(D_1 + 1) \times$ Count clock cycle" or "$(D_2 + 1) \times$ Count clock cycle" originally expected, but may be generated at an interval of "$(10000H + D_2 + 1) \times$ Count clock period".

When the TABnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of the external event count input is detected. Additionally, the set value of the TABnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTABnCC0) is generated.

The INTTABnCC0 signal is generated each time the valid edge of the external event count input has been detected (set value of TABnCCR0 register + 1) times.

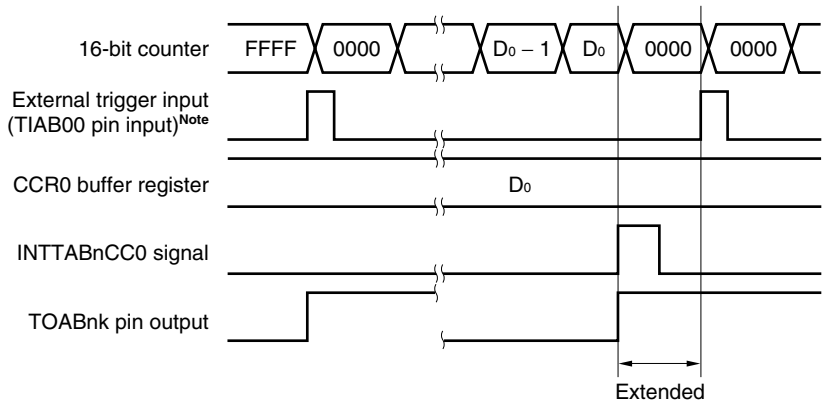
Figure 8-11. Register Setting for Operation in External Event Count Mode (1/2)



Remark **n = 0, 1**

(d) Conflict between trigger detection and match with CCR0 buffer register

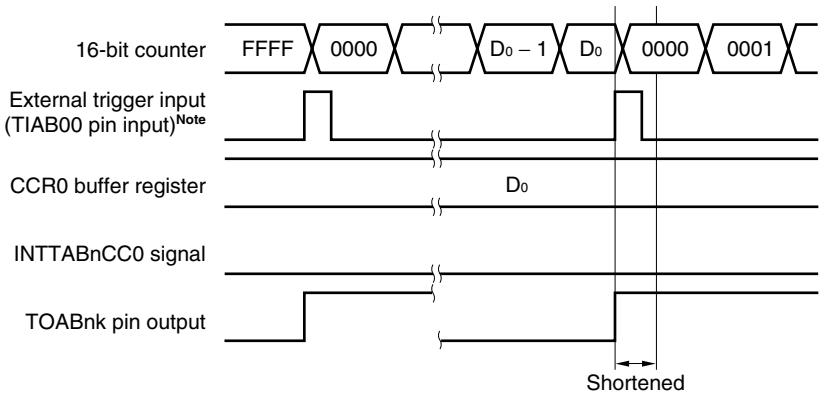
If the trigger is detected immediately after the INTTABnCC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOABnk pin is extended by time from generation of the INTTABnCC0 signal to trigger detection.



Note **TAB1: TRGAB1 pin**

Remark **k = 1 to 3,**
n = 0, 1

If the trigger is detected immediately before the INTTABnCC0 signal is generated, the INTTABnCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOABnk pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Note **TAB1: TRGAB1 pin**

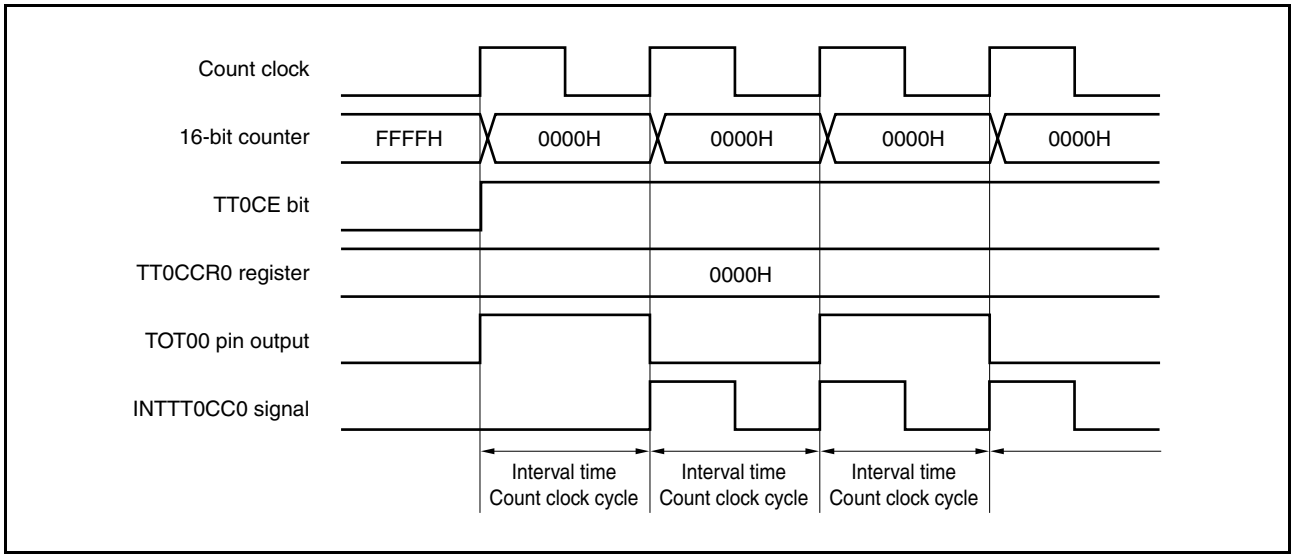
Remark **k = 1 to 3,**
n = 0, 1

(2) Interval timer mode operation timing

(a) Operation if TT0CCR0 register is set to 0000H

If the TT0CCR0 register is set to 0000H, the INTTT0CC0 signal is generated at each count clock, and the output of the TOT00 pin is inverted.

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



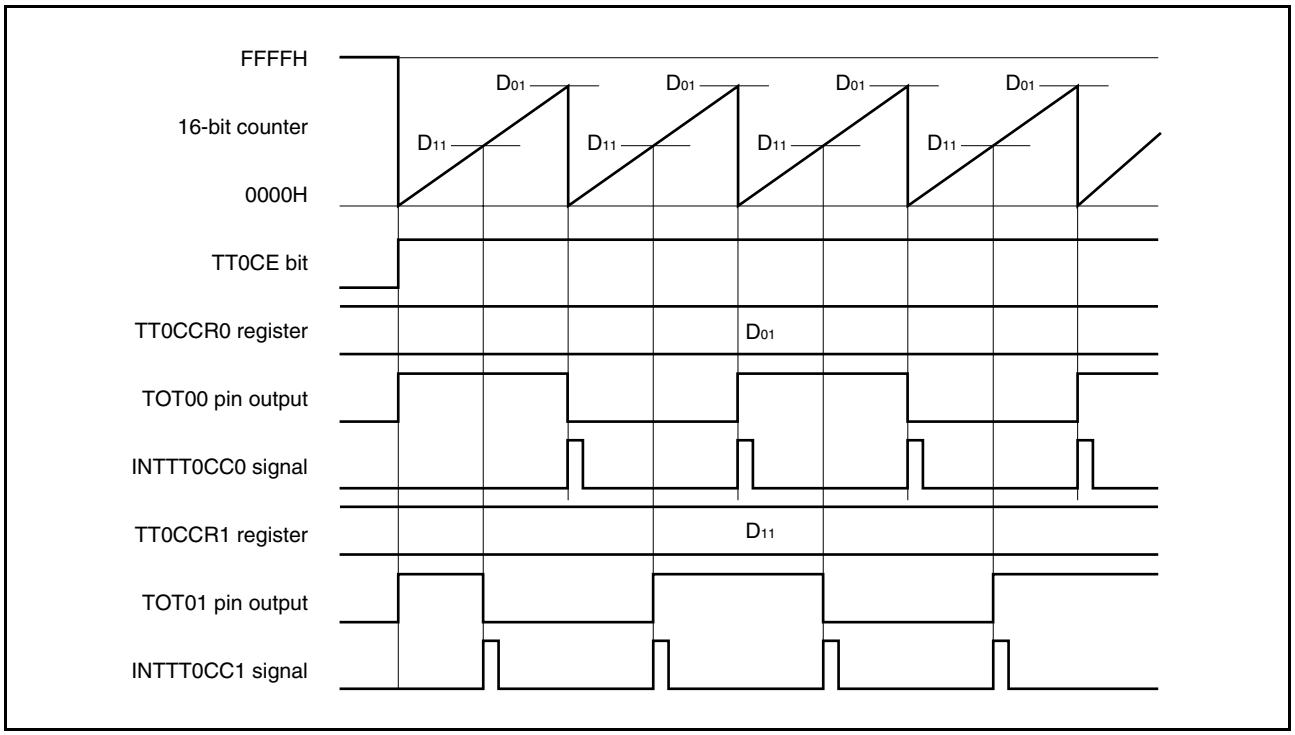
When the TT0CCR1 register is set to the same value as the TT0CCR0 register, the INTTT0CC0 signal is generated at the same timing as the INTTT0CC1 signal and the TOT01 pin output is inverted. In other words, a square wave can be output from the TOT01 pin.

The following shows the operation when the TT0CCR1 register is set to other than the value set in the TT0CCR0 register.

If the set value of the TT0CCR1 register is less than the set value of the TT0CCR0 register, the INTTT0CC1 signal is generated once per cycle. At the same time, the output of the TOT01 pin is inverted.

The TOT01 pin outputs a square wave after outputting a short-width pulse.

Figure 9-12. Timing Chart When $D_{01} \geq D_{11}$



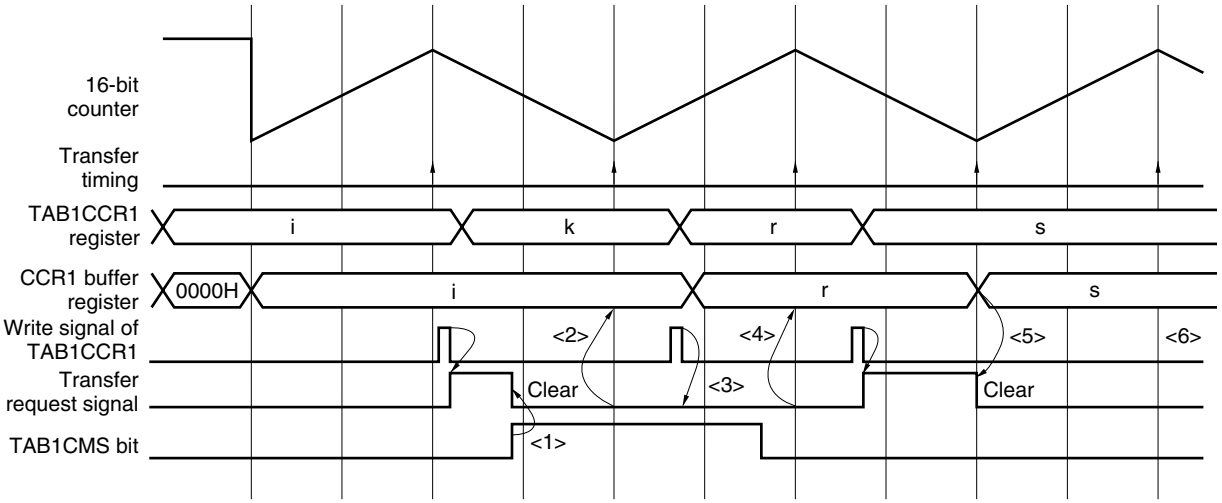
(4) Rewriting TAB1OPT0.TAB1CMS bit

The TAB1CMS bit can select the anytime rewrite mode and batch rewrite mode. This bit can be rewritten during timer operation (when TAB1CTL0.TAB1CE bit = 1). However, the operation and caution illustrated in Figure 11-36 are necessary.

If the TAB1CCR1 register is written when the TAB1CMS bit is cleared to 0, a transfer request signal (internal signal) is set.

When the transfer request signal is set, the register is transferred at the next transfer timing, and the transfer request signal is cleared. This transfer request signal is also cleared when the TAB1CMS bit is set to 1.

Figure 11-36. Rewriting TAB1CMS Bit



<1> If the TAB1CCR1 register is rewritten when the TAB1CMS bit is 0, the transfer request signal is set.
If the TAB1CMS bit is set to 1 in this status, the transfer request signal is cleared.

<2> The register is not transferred because the TAB1CMS bit is set to 1 and the transfer request signal is cleared.

<3> The transfer request signal is not set even if the TAB1CCR1 register is written when the TAB1CMS bit is 1.

<4> The transfer request signal is not set even if the TAB1CCR1 register is written when the TAB1CMS bit is 1, so even if the TAB1CMS bit is cleared to 0, transfer does not occur at the subsequent transfer timing.

<5> The transfer request signal is set if the TAB1CCR1 register is written when the TAB1CMS bit is 0.
Transfer is performed at the subsequent transfer timing and the transfer request signal is cleared.

<6> Once transfer has been performed, the transfer request signal is cleared. Therefore, transfer is not performed at the next transfer timing.

18.4 Registers

The following registers are used to control CSIFn.

- CSIFn control register 0 (CFnCTL0)
- CSIFn control register 1 (CFnCTL1)
- CSIFn control register 2 (CFnCTL2)
- CSIFn status register (CFnSTR)

(1) CSIFn control register 0 (CFnCTL0)

CFnCTL0 is a register that controls the CSIFn serial transfer operation.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 01H.

(1/3)

After reset: 01H R/W Address: CF0CTL0 FFFFFD00H, CF1CTL0 FFFFFD10H, CF2CTL0 FFFFFD20H, CF3CTL0 FFFFFD30H, CF4CTL0 FFFFFD40H

CFnCTL0 (n = 0 to 4)

<7>	<6>	<5>	<4>	3	2	1	<0>
CFnPWR	CFnTXE[Note]	CFnRXE[Note]	CFnDIR[Note]	0	0	CFnTMS[Note]	CFnSCE

CFnPWR	Specification of CSIFn operation disable/enable
0	Disables CSIFn operation and resets the CFnSTR register
1	Enables CSIFn operation
• The CFnPWR bit controls the CSIFn operation and resets the internal circuit.	

CFnTXE[Note]	Specification of transmit operation disable/enable
0	Disables transmit operation
1	Enables transmit operation
• The SOFn output is low level when the CFnTXE bit is 0.	

CFnRXE[Note]	Specification of receive operation disable/enable
0	Disables receive operation
1	Enables receive operation
• No reception completion interrupt is output even when the prescribed data is transferred, and the receive data (CFnRX register) is not updated, because the receive operation is disabled by clearing the CFnRXE bit to 0.	

Note These bits can only be rewritten when the CFnPWR bit = 0. However, CFnPWR bit = 1 can also be set at the same time as rewriting these bits.

Caution To forcibly suspend transmission/reception, clear the CFnPWR bit to 0 instead of the CFnRXE and CFnTXE bits. At this time, the clock output is stopped.

(4/4)

Bit position	Bit name	Function
0	BKI1NK	This bit controls NAK to Endpoint1 (bulk 1 transfer (IN)). 1: Do not transmit NAK. 0: Transmit NAK (default value). This bit is cleared to 0 only when the FIFO connected to the SIE side of the UF0BI1 register (64-byte FIFO of bank configuration) cannot receive data. It is set to 1 when a toggle operation is performed (the data of the UF0BI1 register is retained until transmission has been correctly completed). The bank is changed (toggle operation) when the following conditions are satisfied. • Data is correctly written to the FIFO connected to the CPU bus side (writing has been completed and the FIFO is full or the UF0DEND register is set). • The value of the FIFO counter connected to the SIE side is 0. This bit is automatically set to 1 and data transmission is started when the FIFO on the CPU side becomes full and a FIFO toggle operation is performed as a result of writing data to the FIFO. However, if the FIFO on the CPU side becomes full as a result of writing data to it by DMA while the BKI1T bit of the UF0DEND register is cleared to 0, the toggle operation is not performed because the condition of the toggle operation is not satisfied until the BKI1DED bit of the UF0DEND register is set to 1. To send a short packet that does not make the FIFO on the CPU side full, set the BKI1DED bit to 1 after completing writing data. When the BKI1DED bit is set to 1, a toggle operation is performed and at the same time, this bit is automatically set to 1. This bit is also cleared to 0 as soon as the UF0BI1 register has been cleared.

Cautions 1. If DMA is enabled while data is being written to the UF0BI1 register in the PIO mode, a DMA request is immediately issued.

2. If 64-byte data is written in the DMA transfer mode, the DMA request signal becomes inactive. If the BKI1NK bit is then set to 1, data is transmitted in synchronization with an IN token. The DMA request signal becomes active again as long as the DMA request is not masked as soon as the FIFO is toggled. If the BKI1NK bit is not set, data is not transmitted even if an IN token has been received. In this case, set the BKI1DED bit of the UF0DEND register to 1.

3. If the TC signal is received in the DMA transfer mode, the DMA request signal becomes inactive. At the same time, the DMA request is masked. If the BKI1NK bit is not set to 1, data is not transmitted even if an IN token is received. When the BKI1DED bit of the UF0DEND register is set to 1 by FW, data is transmitted in synchronization with the IN token. To execute DMA transfer again, unmask the DMA request.

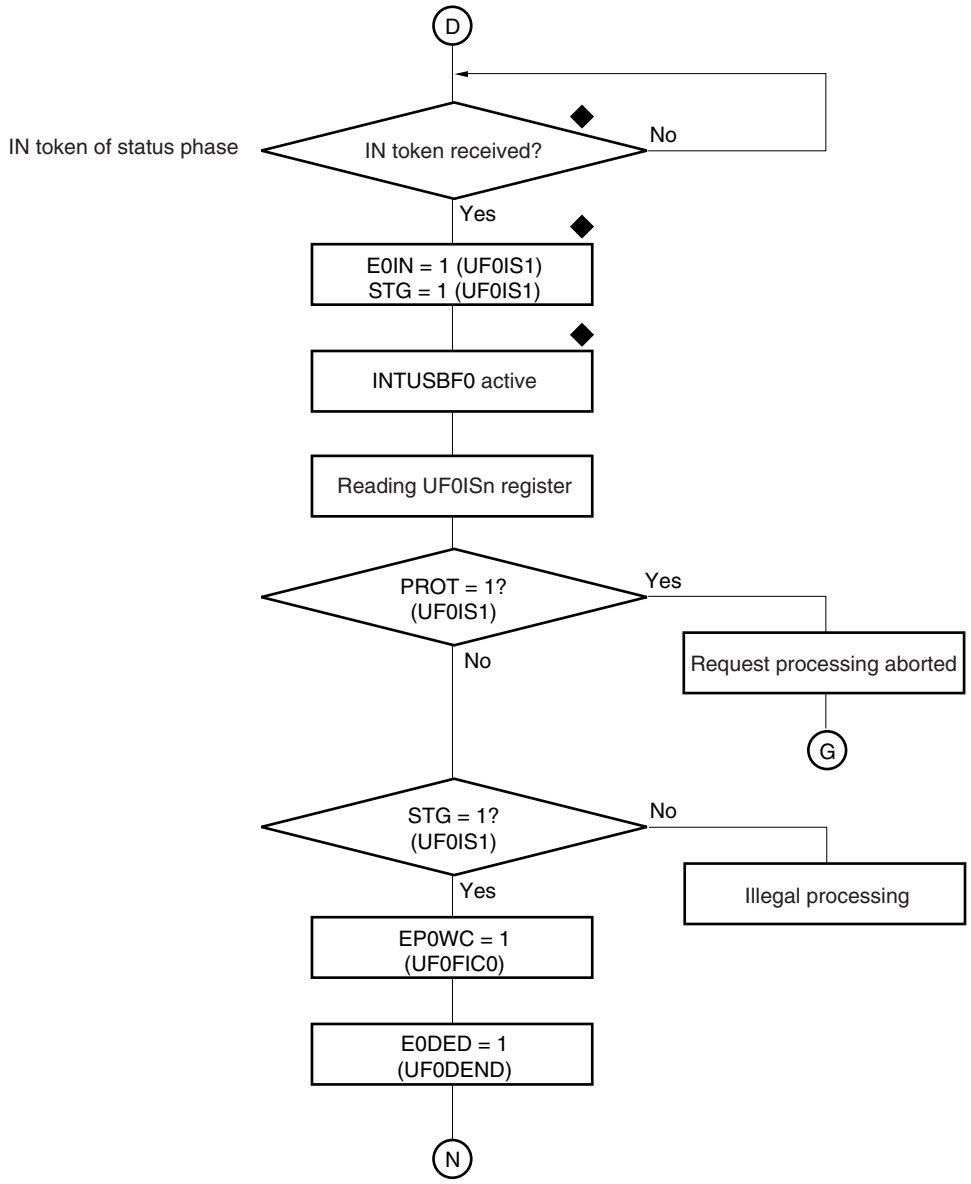
21.8 Register Values in Specific Status

Table 21-8. Register Values in Specific Status (1/2)

Register Name	After CPU Reset ($\overline{\text{RESET}}$)	After Bus Reset
UF0E0N register	00H	Value is held.
UF0E0NA register	00H	Value is held.
UF0EN register	00H	Value is held.
UF0ENM register	00H	Value is held.
UF0SDS register	00H	Value is held.
UF0CLR register	00H	Value is held.
UF0SET register	00H	Value is held.
UF0EPS0 register	00H	Value is held.
UF0EPS1 register	00H	Value is held.
UF0EPS2 register	00H	Value is held.
UF0IS0 register	00H	Value is held.
UF0IS1 register	00H	Value is held.
UF0IS2 register	00H	Value is held.
UF0IS3 register	00H	Value is held.
UF0IS4 register	00H	Value is held.
UF0IM0 register	00H	Value is held.
UF0IM1 register	00H	Value is held.
UF0IM2 register	00H	Value is held.
UF0IM3 register	00H	Value is held.
UF0IM4 register	00H	Value is held.
UF0IC0 register	FFH	Value is held.
UF0IC1 register	FFH	Value is held.
UF0IC2 register	FFH	Value is held.
UF0IC3 register	FFH	Value is held.
UF0IC4 register	FFH	Value is held.
UF0IDR register	00H	Value is held.
UF0DMS0 register	00H	Value is held.
UF0DMS1 register	00H	Value is held.
UF0FIC0 register	00H	Value is held.
UF0FIC1 register	00H	Value is held.
UF0DEND register	00H	Value is held.
UF0GPR register	00H	Value is held.
UF0MODC register	00H	Value is held.
UF0MODS register	00H	Bit 2 (CONF): Cleared (0), Other bits: Value is held.
UF0AIFN register	00H	Value is held.
UF0AAS register	00H	Value is held.
UF0ASS register	00H	00H
UF0E1IM register	00H	Value is held.
UF0E2IM register	00H	Value is held.

Figure 21-24. CPUDEC Request for Control Transfer (11/12)

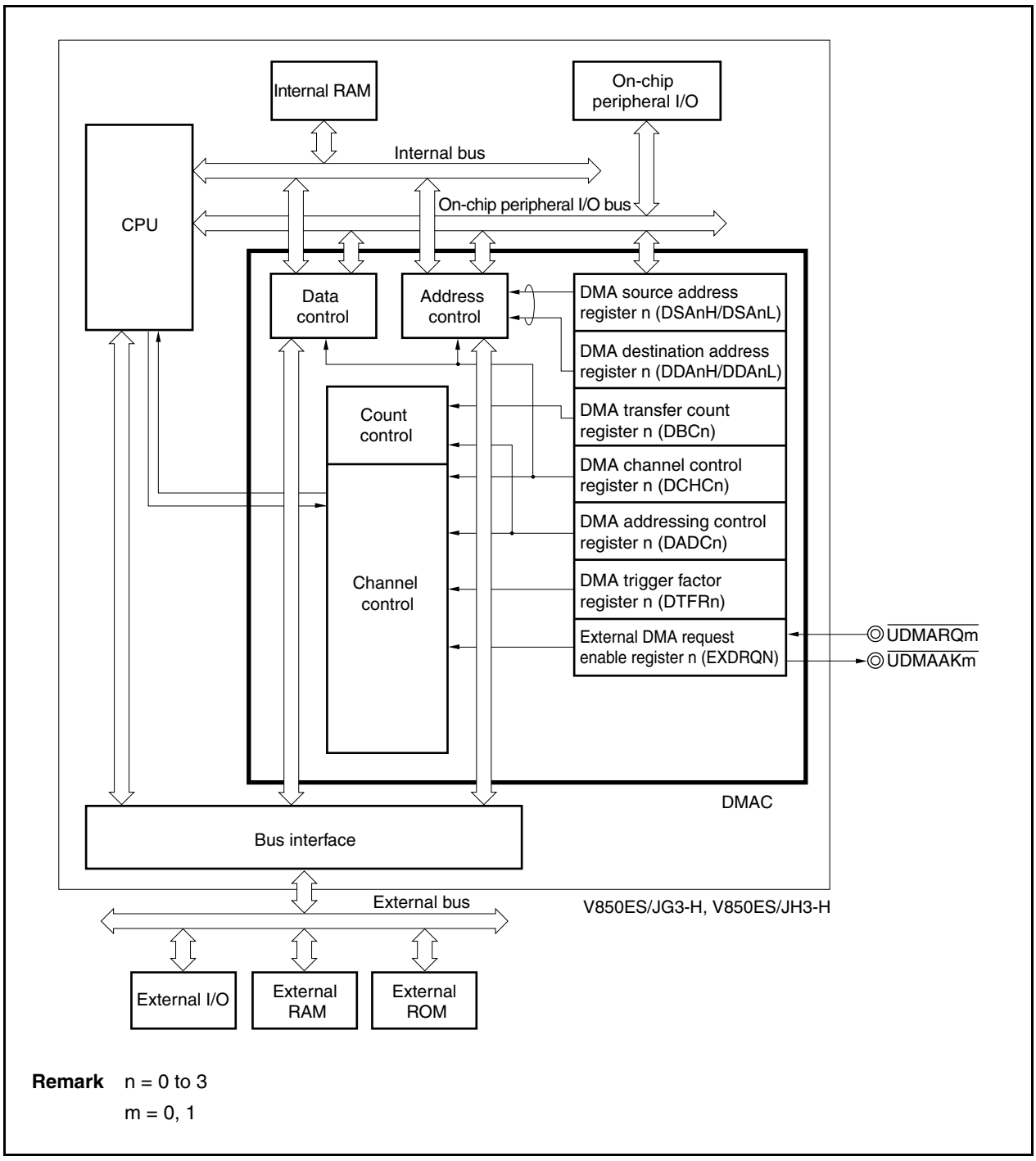
(d) Control transfer (without data stage) (1/2)



Remarks 1. n = 0, 1

2. ♦: Processing by hardware

22.2 Configuration



Remark n = 0 to 3
m = 0, 1

(2) DMA destination address registers 0 to 3 (DDA0 to DDA3)

The DDA0 to DDA3 registers set the DMA destination address (26 bits each) for DMA channel n (n = 0 to 3). These registers are divided into two 16-bit registers, DDAnH and DDAnL.

These registers can be read or written in 16-bit units.

After reset: Undefined R/W Address: DDA0H FFFFF086H, DDA1H FFFFF08EH, DA2H FFFFF096H, DDA3H FFFFF09EH, DDA0L FFFFF084H, DDA1L FFFFF08CH, DDA2L FFFFF094H, DDA3L FFFFF09CH

DDAnH (n = 0 to 3)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IR	0	0	0	0	0	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16

DDAnL (n = 0 to 3)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

IR	Specification of DMA transfer destination
0	External memory or on-chip peripheral I/O
1	Internal RAM

DA25 to DA16	Set an address (A25 to A16) of DMA transfer destination (default value is undefined). During DMA transfer, the next DMA transfer destination address is held. When DMA transfer is completed, the DMA transfer source address set first is held.

DA15 to DA0	Set an address (A15 to A0) of DMA transfer destination (default value is undefined). During DMA transfer, the next DMA transfer destination address is held. When DMA transfer is completed, the DMA transfer source address set first is held.

Cautions 1. Be sure to clear bits 14 to 10 of the DDAnH register to 0.

2. Set the DDAnH and DDAnL registers at the following timing when DMA transfer is disabled (DCHCn.Enn bit = 0).

- **Period from after reset to start of first DMA transfer**
- **Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer**
- **Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer**

3. When the value of the DDAn register is read, two 16-bit registers, DDAnH and DDAnL, are read. If reading and updating conflict, a value being updated may be read (see 22.13 Cautions).

4. Following reset, set the DSAnH, DSAnL, DDAnH, DDAnL, and DBCn registers before starting DMA transfer. If these registers are not set, the operation when DMA transfer is started is not guaranteed.

(3) Oscillation stabilization time select register (OSTS)

The wait time until the oscillation stabilizes after the STOP mode is released or the wait time until the on-chip flash memory stabilizes after the IDLE2 mode is released is controlled by the OSTS register.

This register can be read or written in 8-bit units.

Reset sets this register to 06H.

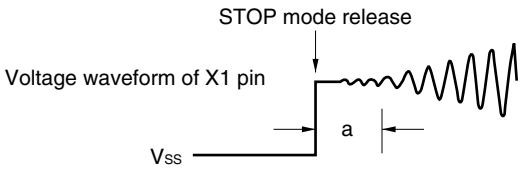
After reset: 06H R/W Address: FFFFF6C0H

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time/setup time[Note]	fx	
				3 MHz	6 MHz
0	0	0	$2^{10}/f_x$	0.341 ms	0.171 ms
0	0	1	$2^{11}/f_x$	0.683 ms	0.341 ms
0	1	0	$2^{12}/f_x$	1.365 ms	0.683 ms
0	1	1	$2^{13}/f_x$	2.730 ms	1.365 ms
1	0	0	$2^{14}/f_x$	5.461 ms	2.731 ms
1	0	1	$2^{15}/f_x$	10.923 ms	5.461 ms
1	1	0	$2^{16}/f_x$	21.85 ms	10.92 ms
1	1	1	Setting prohibited		

Note The oscillation stabilization time and setup time are required when the STOP mode and IDLE2 mode are released, respectively.

Cautions 1. The wait time following release of the STOP mode does not include the time until the clock oscillation starts ("a" in the figure below) following release of the STOP mode, regardless of whether the STOP mode is released by reset or the occurrence of an interrupt request signal.



2. Be sure to set bits 3 to 7 to "0".

3. The oscillation stabilization time following reset release is $2^{16}/f_x$ (because the initial value of the OSTS register = 06H).

Remark fx = Main clock oscillation frequency

After reset: 01H[Note] R/W Address: FFFFF9FCH

	7	6	5	4	3	2	1	<0>
OCDM	0	0	0	0	0	0	0	OCDM0

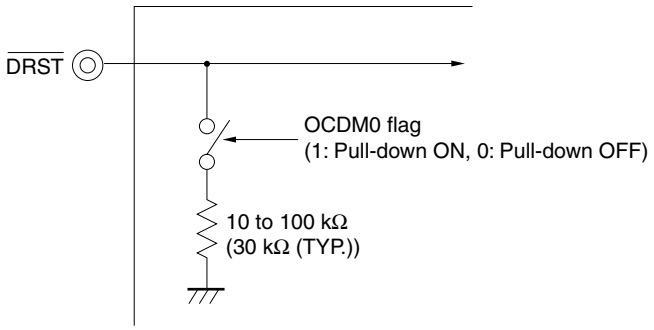
OCDM0	Operation mode
0	Selects normal operation mode (in which a pin that functions alternately as on-chip debug function pin is used as a port/peripheral function pin) and disconnects the on-chip pull-down resistor of the P56/INTP05/$\overline{\text{DRST}}$ pin.
1	When $\overline{\text{DRST}}$ pin is low: Normal operation mode (in which a pin that functions alternately as an on-chip debug function pin is used as a port/peripheral function pin) When $\overline{\text{DRST}}$ pin is high: On-chip debug mode (in which a pin that functions alternately as an on-chip debug function pin is used as an on-chip debug mode pin)

Note $\overline{\text{RESET}}$ input sets this register to 01H. After reset by the WDT2RES signal, clock monitor (CLM), or low-voltage detector (LVI), however, the value of the OCDM register is retained.

Cautions 1. When using the DDI, DDO, DCK, and DMS pins not as on-chip debug pins but as port pins after external reset, any of the following actions must be taken.

- **Input a low level to the P56/INTP05/$\overline{\text{DRST}}$ pin.**
- **Set the OCDM0 bit. In this case, take the following actions.**
  <1> Clear the OCDM0 bit to 0.
  <2> Fix the P56/INTP05/$\overline{\text{DRST}}$ pin to low level until <1> is completed.

2. The $\overline{\text{DRST}}$ pin has an on-chip pull-down resistor. This resistor is disconnected when the OCDM0 flag is set to 0.



33.5.2 Supply current

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = UV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Supply current[Notes 1, 2]	I_{DD1}	Normal operation	f_{XX} = 48 MHz (f_X = 6 MHz) Peripheral function operating			120	mA
			f_{XX} = 48 MHz (f_X = 6 MHz) USBF operating		54		mA
	I_{DD2}	HALT mode	f_{XX} = 48 MHz (f_X = 6 MHz) Peripheral function operating		42	60	mA
	I_{DD3}	IDLE1 mode	f_{XX} = 48 MHz (f_X = 6 MHz), PLL on		4	7	mA
	I_{DD4}	IDLE2 mode	f_{XX} = 6 MHz (f_X = 6 MHz), PLL off		0.5	1	mA
	I_{DD5}	Subclock operation mode	f_{XT} = 32.768 kHz, main clock stopped, internal oscillator stopped		120	600	μA
	I_{DD6}	Sub-IDLE mode	f_{XT} = 32.768 kHz, main clock stopped, internal oscillator stopped, –40≤T_A≤+25°C		13	25	μA
			25≤T_A≤85°C			95	μA
	I_{DD7}	STOP mode	Subclock stopped, internal oscillator stopped, –40≤T_A≤+25°C		10	20	μA
			25≤T_A≤85°C			90	μA
			Subclock operating, internal oscillator stopped, –40≤T_A≤+25°C		13	25	μA
			25≤T_A≤85°C			95	μA
	I_{DD8}	Flash memory programming mode	f_{XX} = 48 MHz (f_X = 6 MHz)		65	130	mA

Notes 1. Total of V_{DD}, EV_{DD}, and UV_{DD} currents. Currents flowing through the output buffers, A/D converter, D/A converter, and on-chip pull-down resistor are not included.

2. The V_{DD} of the TYP. value is 3.3 V.

(4/37)

Symbol	Name	Unit	Page
C0MCTRL26	CAN0 message control register 26	CAN	974
C0MCTRL27	CAN0 message control register 27	CAN	974
C0MCTRL28	CAN0 message control register 28	CAN	974
C0MCTRL29	CAN0 message control register 29	CAN	974
C0MCTRL30	CAN0 message control register 30	CAN	974
C0MCTRL31	CAN0 message control register 31	CAN	974
C0MDATA000	CAN0 message data byte 0 register 00	CAN	969
C0MDATA001	CAN0 message data byte 0 register 01	CAN	969
C0MDATA002	CAN0 message data byte 0 register 02	CAN	969
C0MDATA003	CAN0 message data byte 0 register 03	CAN	969
C0MDATA004	CAN0 message data byte 0 register 04	CAN	969
C0MDATA005	CAN0 message data byte 0 register 05	CAN	969
C0MDATA006	CAN0 message data byte 0 register 06	CAN	969
C0MDATA007	CAN0 message data byte 0 register 07	CAN	969
C0MDATA008	CAN0 message data byte 0 register 08	CAN	969
C0MDATA009	CAN0 message data byte 0 register 09	CAN	969
C0MDATA010	CAN0 message data byte 0 register 10	CAN	969
C0MDATA0100	CAN0 message data byte 01 register 00	CAN	969
C0MDATA0101	CAN0 message data byte 01 register 01	CAN	969
C0MDATA0102	CAN0 message data byte 01 register 02	CAN	969
C0MDATA0103	CAN0 message data byte 01 register 03	CAN	969
C0MDATA0104	CAN0 message data byte 01 register 04	CAN	969
C0MDATA0105	CAN0 message data byte 01 register 05	CAN	969
C0MDATA0106	CAN0 message data byte 01 register 06	CAN	969
C0MDATA0107	CAN0 message data byte 01 register 07	CAN	969
C0MDATA0108	CAN0 message data byte 01 register 08	CAN	969
C0MDATA0109	CAN0 message data byte 01 register 09	CAN	969
C0MDATA011	CAN0 message data byte 0 register 11	CAN	969
C0MDATA0110	CAN0 message data byte 01 register 10	CAN	969
C0MDATA0111	CAN0 message data byte 01 register 11	CAN	969
C0MDATA0112	CAN0 message data byte 01 register 12	CAN	969
C0MDATA0113	CAN0 message data byte 01 register 13	CAN	969
C0MDATA0114	CAN0 message data byte 01 register 14	CAN	969
C0MDATA0115	CAN0 message data byte 01 register 15	CAN	969
C0MDATA0116	CAN0 message data byte 01 register 16	CAN	969
C0MDATA0117	CAN0 message data byte 01 register 17	CAN	969
C0MDATA0118	CAN0 message data byte 01 register 18	CAN	969
C0MDATA0119	CAN0 message data byte 01 register 19	CAN	969
C0MDATA012	CAN0 message data byte 0 register 12	CAN	969
C0MDATA0120	CAN0 message data byte 01 register 20	CAN	969

(18/37)

Symbol	Name	Unit	Page
DCHC3	DMA channel control register 3	DMAC	1238
DDA0H	DMA destination address register 0H	DMAC	1235
DDA0L	DMA destination address register 0L	DMAC	1235
DDA1H	DMA destination address register 1H	DMAC	1235
DDA1L	DMA destination address register 1L	DMAC	1235
DDA2H	DMA destination address register 2H	DMAC	1235
DDA2L	DMA destination address register 2L	DMAC	1235
DDA3H	DMA destination address register 3H	DMAC	1235
DDA3L	DMA destination address register 3L	DMAC	1235
DMAIC0	Interrupt control register	INTC	1260
DMAIC1	Interrupt control register	INTC	1260
DMAIC2	Interrupt control register	INTC	1260
DMAIC3	Interrupt control register	INTC	1260
DSA0H	DMA source address register 0H	DMAC	1234
DSA0L	DMA source address register 0L	DMAC	1234
DSA1H	DMA source address register 1H	DMAC	1234
DSA1L	DMA source address register 1L	DMAC	1234
DSA2H	DMA source address register 2H	DMAC	1234
DSA2L	DMA source address register 2L	DMAC	1234
DSA3H	DMA source address register 3H	DMAC	1234
DSA3L	DMA source address register 3L	DMAC	1234
DTFR0	DMA trigger factor register 0	DMAC	1239
DTFR1	DMA trigger factor register 1	DMAC	1239
DTFR2	DMA trigger factor register 2	DMAC	1239
DTFR3	DMA trigger factor register 3	DMAC	1239
DWC0	Data wait control register 0	BCU	89
ECR	Interrupt source register	CPU	59
EIPC	Interrupt status saving register	CPU	58
EIPSW	Interrupt status saving register	CPU	58
EPCCLT	EPC macro control register	USBF	1164
ERRIC0	Interrupt control register	INTC	1265
EXDRQEN	External DMA request enable register	DMA	1242
FEPC	NMI status saving register	CPU	59
FEPSW	NMI status saving register	CPU	59
HZA0CTL0	High-impedance output control register 0	Motor	586
HZA0CTL1	High-impedance output control register 1	Motor	586
IIC0	IIC shift register 0	I^2C	835
IIC1	IIC shift register 1	I^2C	835
IIC2	IIC shift register 2	I^2C	835
IICC0	IIC control register 0	I^2C	822
IICC1	IIC control register 1	I^2C	822
IICC2	IIC control register 2	I^2C	822

(3/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock i	Execution Clock r	Execution Clock l	Flags CY	Flags OV	Flags S	Flags Z	Flags SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))		1	1	Note 11					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 **Note 12**	SR[regID]←GR[reg2]	Other than regID = PSW	1	1	1					
				regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Halfword)		1	1	Note 11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)		1	1	Note 11					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]		1	1	1					
	imm5,reg2	rrrrr010000iiiii	GR[reg2]←sign-extend(imm5)		1	1	1					
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii IIIIIIIIIIIIIIII	GR[reg1]←imm32		2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 ll 0^{16})		1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1] **Note 14**		1	4	5					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII00 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)		1	4	5					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2]Note 6xGR[reg1]Note 6		1	1	2					
	imm5,reg2	rrrrr010111iiiii	GR[reg2]←GR[reg2]Note 6xsign-extend(imm5)		1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]Note 6ximm16		1	1	2					
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1] **Note 14**		1	4	5					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII10 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)		1	4	5					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.		1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])		1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	