Renesas Electronics America Inc - UPD70F3771GF-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	Obsolete
Core Processor	V850ES
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	CANbus, CSI, EBI/EMI, I²C, UART/USART, USB
Peripherals	DMA, LVD, PWM, WDT
Number of I/O	96
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	40K 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/upd70f3771gf-gat-ax

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

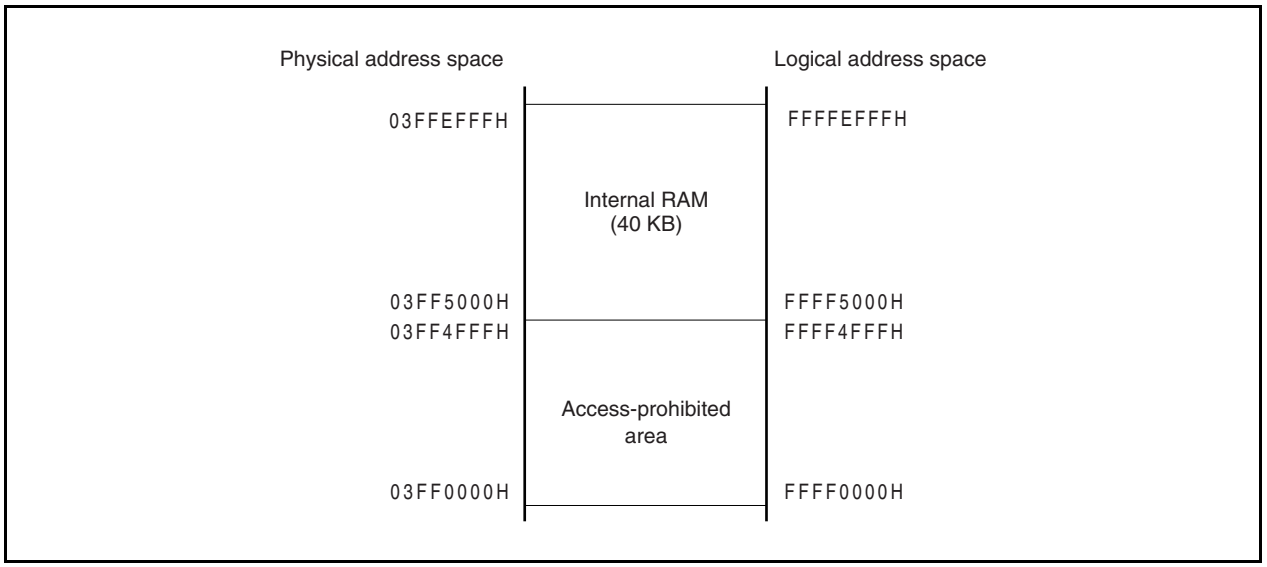
(2012.4)

(b) Internal RAM (40 KB)

40 KB are allocated to addresses 03FF5000H to 03FFEFFFH in the following products.
Accessing addresses 03FF0000H to 03FF4FFFH is prohibited.

- μPD70F3761, 70F3766

Figure 3-8. Internal RAM Area (40 KB)

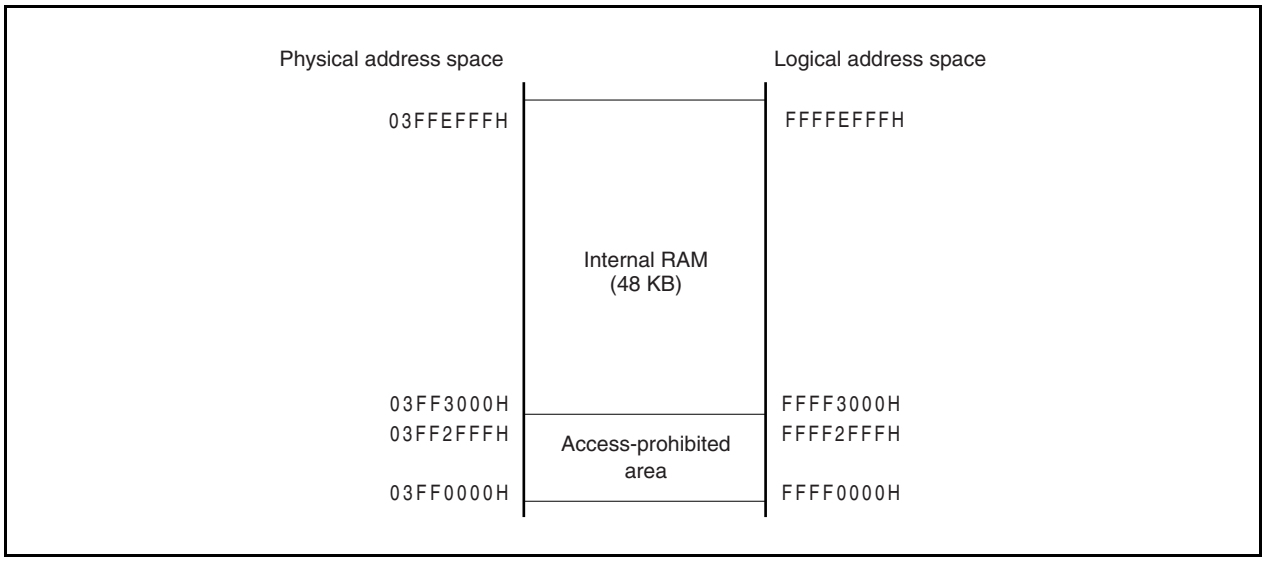


(c) Internal RAM (48 KB)

48 KB are allocated to addresses 03FF3000H to 03FFEFFFH in the following products.
Accessing addresses 03FF0000H to 03FF2FFFH is prohibited.

- μPD70F3762, 70F3767

Figure 3-9. Internal RAM Area (48 KB)



(11/14)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFA24H	UARTC2 status register	UC2STR	R/W	√	√		00H
FFFFFA26H	UARTC2 receive data register	UC2RX	R			√	01FFH
FFFFFA26H	UARTC2 receive data register L	UC2RXL			√		FFH
FFFFFA28H	UARTC2 transmit data register	UC2TX	R/W			√	01FFH
FFFFFA28H	UARTC2 transmit data register L	UC2TXL			√		FFH
FFFFFA2AH	UARTC2 option control register 1	UC2OPT1		√	√		00H
FFFFFA30H	UARTC3 control register 0	UC3CTL0		√	√		10H
FFFFFA31H	UARTC3 control register 1	UC3CTL1			√		00H
FFFFFA32H	UARTC3 control register 2	UC3CTL2			√		FFH
FFFFFA33H	UARTC3 option control register 0	UC3OPT0		√	√		14H
FFFFFA34H	UARTC3 status register	UC3STR		√	√		00H
FFFFFA36H	UARTC3 receive data register	UC3RX	R			√	01FFH
FFFFFA36H	UARTC3 receive data register L	UC3RXL			√		FFH
FFFFFA38H	UARTC3 transmit data register	UC3TX	R/W			√	01FFH
FFFFFA38H	UARTC3 transmit data register L	UC3TXL			√		FFH
FFFFFA3AH	UARTC3 option control register 1	UC3OPT1		√	√		00H
FFFFFA40H	UARTC4 control register 0	UC4CTL0		√	√		10H
FFFFFA41H	UARTC4 control register 1	UC4CTL1			√		00H
FFFFFA42H	UARTC4 control register 2	UC4CTL2			√		FFH
FFFFFA43H	UARTC4 option control register 0	UC4OPT0		√	√		14H
FFFFFA44H	UARTC4 status register	UC4STR		√	√		00H
FFFFFA46H	UARTC4 receive data register	UC4RX	R			√	01FFH
FFFFFA46H	UARTC4 receive data register L	UC4RXL			√		FFH
FFFFFA48H	UARTC4 transmit data register	UC4TX	R/W			√	01FFH
FFFFFA48H	UARTC4 transmit data register L	UC4TXL			√		FFH
FFFFFA4AH	UARTC4 option control register 1	UC4OPT1		√	√		00H
FFFFFA80H	TMM0 control register 0	TM0CTL0		√	√		00H
FFFFFA84H	TMM0 compare register 0	TM0CMP0				√	0000H
FFFFFA90H	TMM1 control register 0	TM1CTL0		√	√		00H
FFFFFA94H	TMM1 compare register 0	TM1CMP0				√	0000H
FFFFFAA0H	TMM2 control register 0	TM2CTL0		√	√		00H
FFFFFAA4H	TMM2 compare register 0	TM2CMP0				√	0000H
FFFFFAB0H	TMM3 control register 0	TM3CTL0		√	√		00H
FFFFFAB4H	TMM3 compare register 0	TM3CMP0				√	0000H
FFFFFAD0H	Sub-count register	RC1SUBC	R			√	0000H
FFFFFAD2H	Second count register	RC1SEC	R/W		√		00H
FFFFFAD3H	Minute count register	RC1MIN			√		00H
FFFFFAD4H	Hour count register	RC1HOUR			√		12H
FFFFFAD5H	Week count register	RC1WEEK			√		00H
FFFFFAD6H	Day count register	RC1DAY			√		01H
FFFFFAD7H	Month count register	RC1MONTH			√		01H
FFFFFAD8H	Year count register	RC1YEAR			√		00H
FFFFFAD9H	Time error correction register	RC1SUBU		√	√		00H

(13/14)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFD14H	CSIF1 receive data register	CF1RX	R			√	0000H
FFFFFD14H	CSIF1 receive data register L	CF1RXL			√		00H
FFFFFD16H	CSIF1 transmit data register	CF1TX	R/W			√	0000H
FFFFFD16H	CSIF1 transmit data register L	CF1TXL			√		00H
FFFFFD20H	CSIF2 control register 0	CF2CTL0		√	√		01H
FFFFFD21H	CSIF2 control register 1	CF2CTL1		√	√		00H
FFFFFD22H	CSIF2 control register 2	CF2CTL2			√		00H
FFFFFD23H	CSIF2 status register	CF2STR		√	√		00H
FFFFFD24H	CSIF2 receive data register	CF2RX	R			√	0000H
FFFFFD24H	CSIF2 receive data register L	CF2RXL			√		00H
FFFFFD26H	CSIF2 transmit data register	CF2TX	R/W			√	0000H
FFFFFD26H	CSIF2 transmit data register L	CF2TXL			√		00H
FFFFFD30H	CSIF3 control register 0	CF3CTL0		√	√		01H
FFFFFD31H	CSIF3 control register 1	CF3CTL1		√	√		00H
FFFFFD32H	CSIF3 control register 2	CF3CTL2			√		00H
FFFFFD33H	CSIF3 status register	CF3STR		√	√		00H
FFFFFD34H	CSIF3 receive data register	CF3RX	R			√	0000H
FFFFFD34H	CSIF3 receive data register L	CF3RXL			√		00H
FFFFFD36H	CSIF3 transmit data register	CF3TX	R/W			√	0000H
FFFFFD36H	CSIF3 transmit data register L	CF3TXL			√		00H
FFFFFD40H	CSIF4 control register 0	CF4CTL0		√	√		01H
FFFFFD41H	CSIF4 control register 1	CF4CTL1		√	√		00H
FFFFFD42H	CSIF4 control register 2	CF4CTL2			√		00H
FFFFFD43H	CSIF4 status register	CF4STR		√	√		00H
FFFFFD44H	CSIF4 receive data register	CF4RX	R			√	0000H
FFFFFD44H	CSIF4 receive data register L	CF4RXL			√		00H
FFFFFD46H	CSIF4 transmit data register	CF4TX	R/W			√	0000H
FFFFFD46H	CSIF4 transmit data register L	CF4TXL			√		00H
FFFFFD80H	IIC shift register 0	IIC0			√		00H
FFFFFD82H	IIC control register 0	IICC0		√	√		00H
FFFFFD83H	Slave address register 0	SVA0			√		00H
FFFFFD84H	IIC clock select register 0	IICCL0		√	√		00H
FFFFFD85H	IIC function expansion register 0	IICX0		√	√		00H
FFFFFD86H	IIC status register 0	IICS0	R	√	√		00H
FFFFFD8AH	IIC flag register 0	IICF0	R/W	√	√		00H
FFFFFD90H	IIC shift register 1	IIC1			√		00H
FFFFFD92H	IIC control register 1	IICC1		√	√		00H
FFFFFD93H	Slave address register 1	SVA1			√		00H
FFFFFD94H	IIC clock select register 1	IICCL1		√	√		00H
FFFFFD95H	IIC function expansion register 1	IICX1		√	√		00H
FFFFFD96H	IIC status register 1	IICS1	R	√	√		00H
FFFFFD9AH	IIC flag register 1	IICF1	R/W	√	√		00H
FFFFFDA0H	IIC shift register 2	IIC2			√		00H

(1) Main clock oscillator

The main clock oscillator oscillates the following frequencies (f_X).

- In clock-through mode
  f_X = 3.0 to 6.0 MHz
- In PLL mode
  f_X = 3.0 to 6.0 MHz (×8)

(2) Subclock oscillator

The sub-resonator oscillates a frequency of 32.768 kHz (f_{XT}).

(3) Main clock oscillator stop control

This circuit generates a control signal that stops oscillation of the main clock oscillator.
Oscillation of the main clock oscillator is stopped in the STOP mode or when the PCC.MCK bit = 1 (valid only when the PCC.CLS bit = 1).

(4) Internal oscillator

Oscillates a frequency (f_R) of 220 kHz (TYP.).

(5) Prescaler 1

This prescaler generates the clock (f_{XX} to f_{XX}/1,024) to be supplied to the following on-chip peripheral functions: TAA, TAB, TMM, TMT, CSIF, UARTC, I^2C, CAN, ADC, DAC, WDT2

(6) Prescaler 2

This circuit divides the main clock (f_{XX}).
The clock generated by prescaler 2 (f_{XX} to f_{XX}/32) is supplied to the selector that generates the CPU clock (f_{CPU}) and internal system clock (f_{CLK}).
f_{CLK} is the clock supplied to the INTC, ROM, and RAM blocks, and can be output from the CLKOUT pin.

(7) Prescaler 3

This circuit divides the clock generated by the main clock oscillator (f_X) to a specific frequency (32.768 kHz) and supplies that clock to the real-time counter (RTC) block.

(8) PLL

This circuit multiplies the clock generated by the main clock oscillator (f_X) by 8.
It operates in two modes: clock-through mode in which f_X is output as is, and PLL mode in which a multiplied clock is output. These modes can be selected by using the PLLCTL.SELPLL bit.

If the set value of the TAAnCCR1 register is less than the set value of the TAAnCCR0 register, the INTTAAnCC1 signal is generated once per cycle. At the same time, the output of the TOAAn1 pin is inverted. The TOAAn1 pin outputs a square wave with the same cycle as that output by the TOAAn0 pin.

Figure 7-12. Timing Chart When $D_{01} \geq D_{11}$

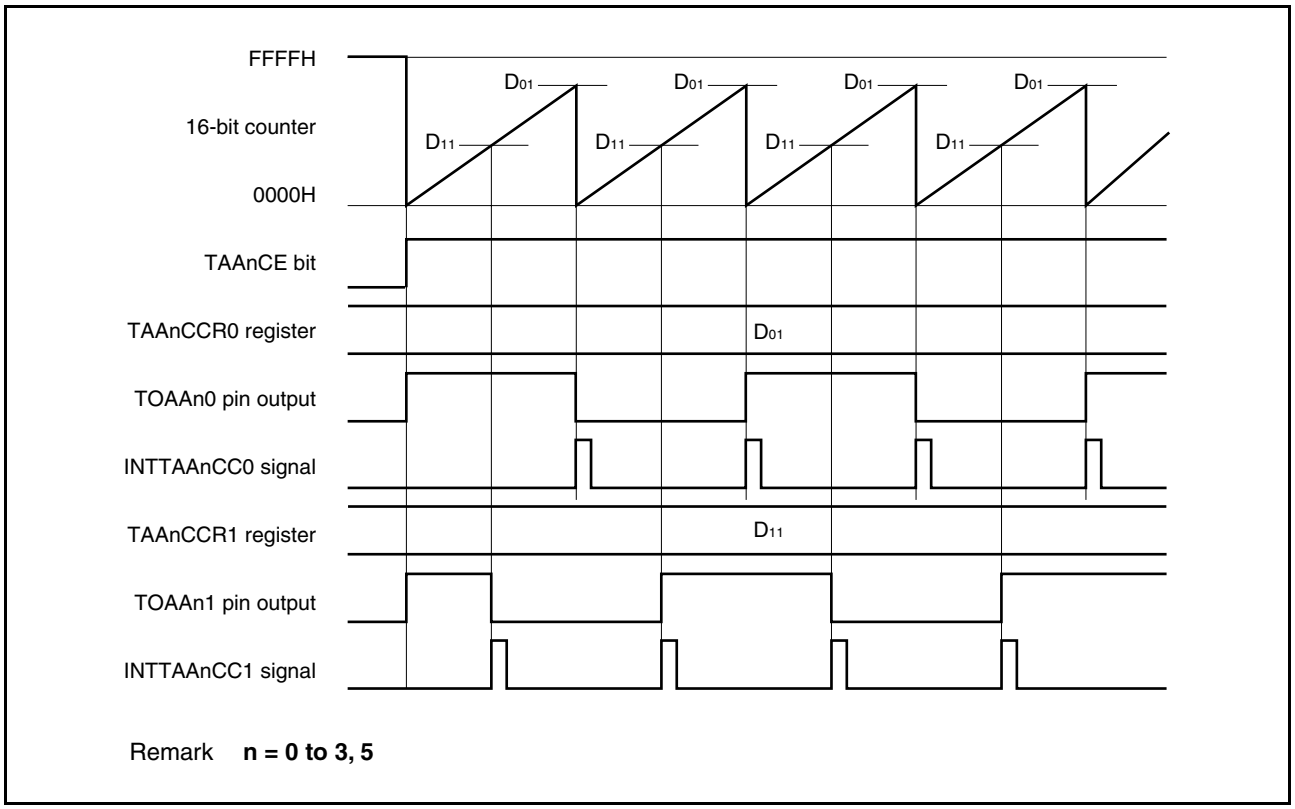
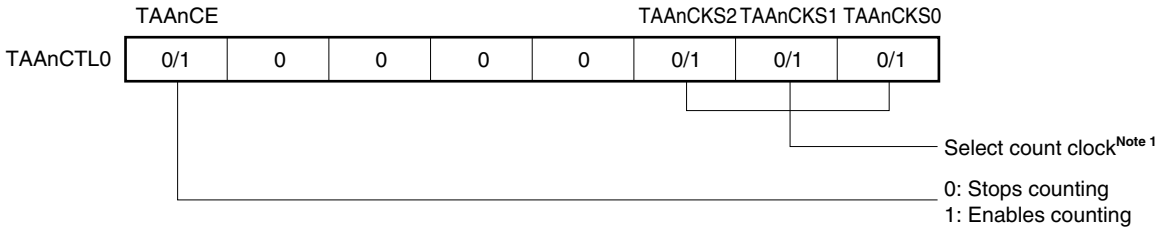
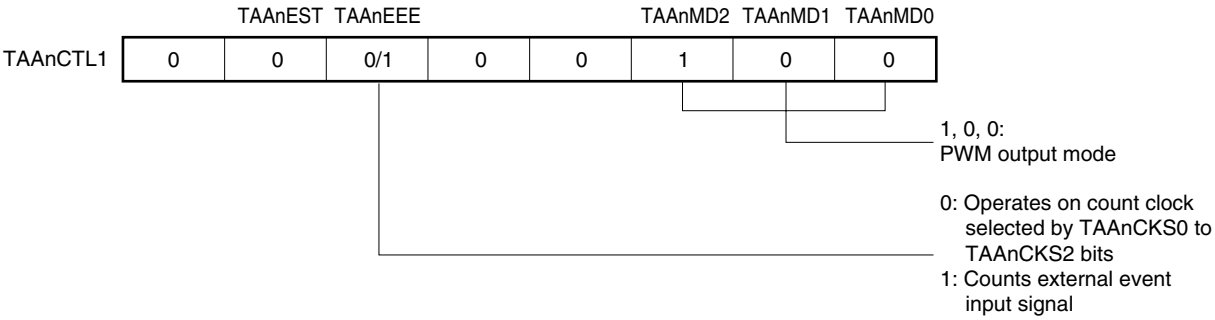


Figure 7-31. Setting of Registers in PWM Output Mode (1/2)

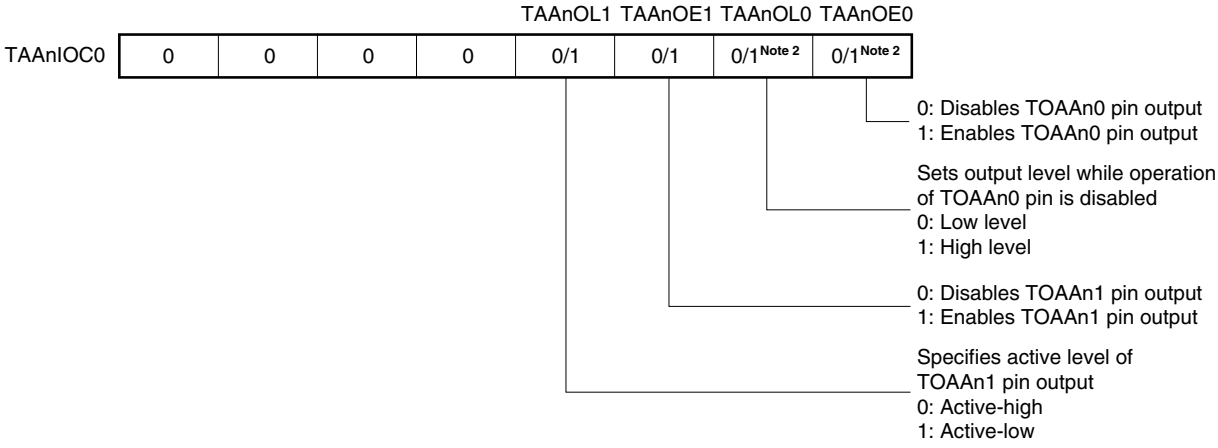
(a) TAAn control register 0 (TAAnCTL0)



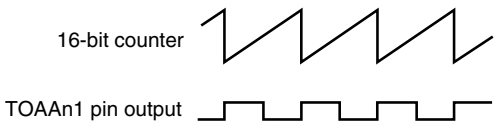
(b) TAAn control register 1 (TAAnCTL1)



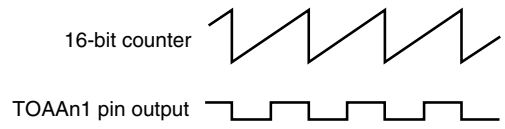
(c) TAAn I/O control register 0 (TAAnIOC0)



- When TAAnOL1 bit = 0



- When TAAnOL1 bit = 1



Notes 1. The setting is invalid when the TAAnCTL1.TAAnEEE bit = 1.

2. Clear this bit to 0 when the TOAAn0 pin is not used in the PWM output mode.

In order to transfer data from the TT0CCRn register to the CCRn buffer register, the TT0CCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TT0CCR0 register and then set the active level width to the TT0CCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TT0CCR0 register, and then write the same value (same as preset value of the TT0CCR1 register) to the TT0CCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TT0CCR1 register has to be set.

After data is written to the TT0CCR1 register, the value written to the TT0CCRn register is transferred to the CCRn buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TT0CCR0 or TT0CCR1 register again after writing the TT0CCR1 register once, do so after the INTTT0CC0 signal is generated. Otherwise, the value of the CCRn buffer register may become undefined because the timing of transferring data from the TT0CCRn register to the CCRn buffer register conflicts with writing the TT0CCRn register.

Remark n = 0, 1

(9) Day count register (RC1DAY)

The RC1DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.

It counts up when the hour counter overflows.

This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February in leap year)
- 01 to 28 (February in normal year)

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (2 x 32.768 kHz) later. Set a decimal value of 00 to 31 to this register in BCD code.

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

Reset sets this register to 01H.

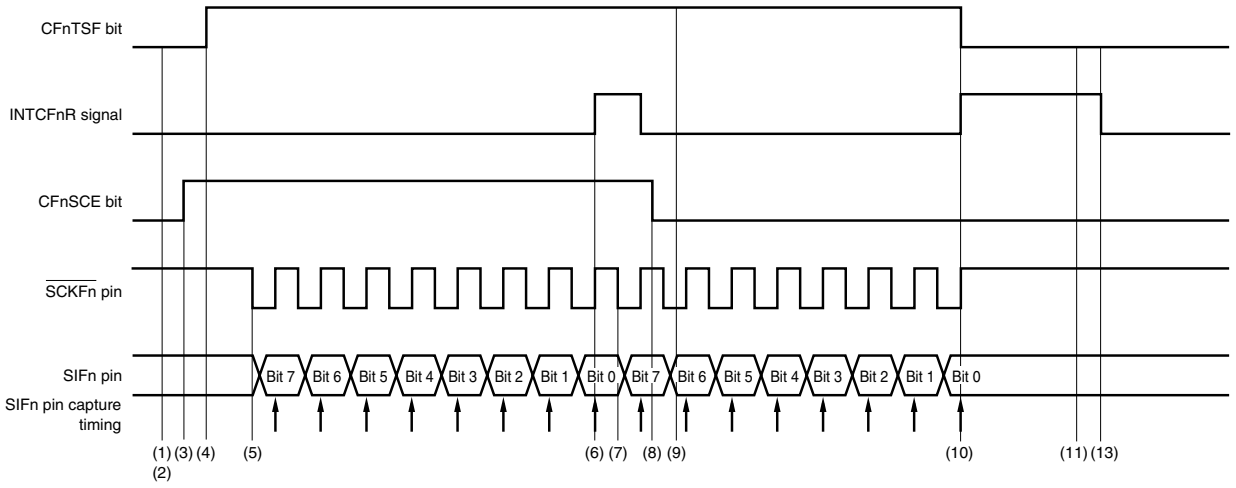
Caution Setting a value other than 01 to 31 to the RC1DAY register is prohibited. Setting a value outside the above-mentioned count range, such as "February 30" is also prohibited.

Remark See **12.4.1 Initial settings**, **12.4.2 Rewriting each counter during the real-time counter operation**, and **12.4.3 Reading each counter during the real-time counter operation** when reading or writing the RC1DAY register.

After reset: 01H R/W Address: FFFFFAD6H

	7	6	5	4	3	2	1	0
RC1DAY	0	0						

(2) Operation timing



(1) Write 07H to the CFnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKFn}$), and slave mode.
(2) Write 00H to the CFnCTL2 register, and set the transfer data length to 8 bits.
(3) Write A3H to the CFnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).
(4) The CFnSTR.CFnTSF bit is set to 1 by performing a dummy read of the CFnRX register, and the device waits for a serial clock input.
(5) When a serial clock is input, capture the receive data of the SIFn pin in synchronization with the serial clock.
(6) When reception is completed, the reception completion interrupt request signal (INTCFnR) is generated, and reading of the CFnRX register is enabled.
(7) When a serial clock is input in the CFnCTL0.CFnSCE bit = 1 status, continuous reception is started.
(8) To end continuous reception with the current reception, write the CFnSCE bit = 0.
(9) Read the CFnRX register.
(10) When reception is completed, the INTCFnR signal is generated, and reading of the CFnRX register is enabled. When CFnSCE bit = 0 is set before communication completion, clear the CFnTSF bit to 0 to end the receive operation.
(11) Read the CFnRX register.
(12) If an overrun error occurs, write CFnSTR.CFnOVE bit = 0, and clear the error flag.
(13) To release the reception enable status, write CFnCTL0.CFnPWR bit = 0 and CFnCTL0.CFnRXE bit = 0 after checking that the CFnTSF bit = 0.

Remark n = 0 to 4

19.7.4 Operation without communication

(1) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

Δ1

Δ 1: IICSn register = 00000001B

Remarks 1. Δ: Generated only when SPIEn bit = 1

2. n = 0 to 2

19.7.5 Arbitration loss operation (operation as slave after arbitration loss)

(1) When arbitration loss occurs during transmission of slave address data

<1> When IICCn.WTIMn bit = 0

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn register = 0101X110B (Example: When IICSn.ALDn bit is read during interrupt servicing)

▲2: IICSn register = 0001X000B

▲3: IICSn register = 0001X000B

Δ 4: IICSn register = 00000001B

Remarks 1. ▲: Always generated

Δ: Generated only when IICCn.SPIEn bit = 1

X: don't care

2. n = 0 to 2

<2> When WTIMn bit = 1

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn register = 0101X110B (Example: When ALDn bit is read during interrupt servicing)

▲2: IICSn register = 0001X100B

▲3: IICSn register = 0001XX00B

Δ 4: IICSn register = 00000001B

Remarks 1. ▲: Always generated

Δ: Generated only when SPIEn bit = 1

X: don't care

2. n = 0 to 2

20.1.2 Overview of functions

Table 20-1 presents an overview of the CAN controller functions.

Table 20-1. Overview of Functions

Function	Details
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Maximum 1 Mbps (CAN clock input ≥ 8 MHz)
Data storage	Storing messages in the CAN RAM
Number of messages	• 32 message buffers/channels • Each message buffer can be set to be either a transmit message buffer or a receive message buffer.
Message reception	• Unique ID can be set to each message buffer. • Mask setting of four patterns is possible for each channel. • A receive completion interrupt is generated each time a message is received and stored in a message buffer. • Two or more receive message buffers can be used as a FIFO receive buffer (multi-buffer receive block function). • Receive history list function
Message transmission	• Unique ID can be set to each message buffer. • Transmit completion interrupt for each message buffer • Message buffer numbers 0 to 7 specified as transmit message buffers can be used for automatic block transfer. Message transmission interval is programmable (automatic block transmission function (hereafter referred to as “ABT”)). • Transmission history list function
Remote frame processing	Remote frame processing by transmit message buffer
Time stamp function	• The time stamp function can be set for a receive message when a 16-bit timer is used in combination. • The time stamp capture trigger can be selected (SOF or EOF in a CAN message frame can be detected).
Diagnostic function	• Readable error counters • “Valid protocol operation flag” for verification of bus connections • Receive-only mode • Single-shot mode • CAN protocol error type decoding • Self-test mode
Release from bus-off state	• Can be forcibly released from bus-off by software (timing restrictions are ignored). • Cannot be automatically released from bus-off (release request by software is required).
Power save mode	• CAN sleep mode (can be woken up by CAN bus) • CAN stop mode (cannot be woken up by CAN bus)

(2/2)

C0MDATA45m

15	14	13	12	11	10	9	8
MDATA45 15	MDATA45 14	MDATA45 13	MDATA45 12	MDATA45 11	MDATA45 10	MDATA45 9	MDATA45 8
7	6	5	4	3	2	1	0
MDATA45 7	MDATA45 6	MDATA45 5	MDATA45 4	MDATA45 3	MDATA45 2	MDATA45 1	MDATA45 0

C0MDATA4m

7	6	5	4	3	2	1	0
MDATA4 7	MDATA4 6	MDATA4 5	MDATA4 4	MDATA4 3	MDATA4 2	MDATA4 1	MDATA4 0

C0MDATA5m

7	6	5	4	3	2	1	0
MDATA5 7	MDATA5 6	MDATA5 5	MDATA5 4	MDATA5 3	MDATA5 2	MDATA5 1	MDATA5 0

C0MDATA67m

15	14	13	12	11	10	9	8
MDATA67 15	MDATA67 14	MDATA67 13	MDATA67 12	MDATA67 11	MDATA67 10	MDATA67 9	MDATA67 8
7	6	5	4	3	2	1	0
MDATA67 7	MDATA67 6	MDATA67 5	MDATA67 4	MDATA67 3	MDATA67 2	MDATA67 1	MDATA67 0

C0MDATA6m

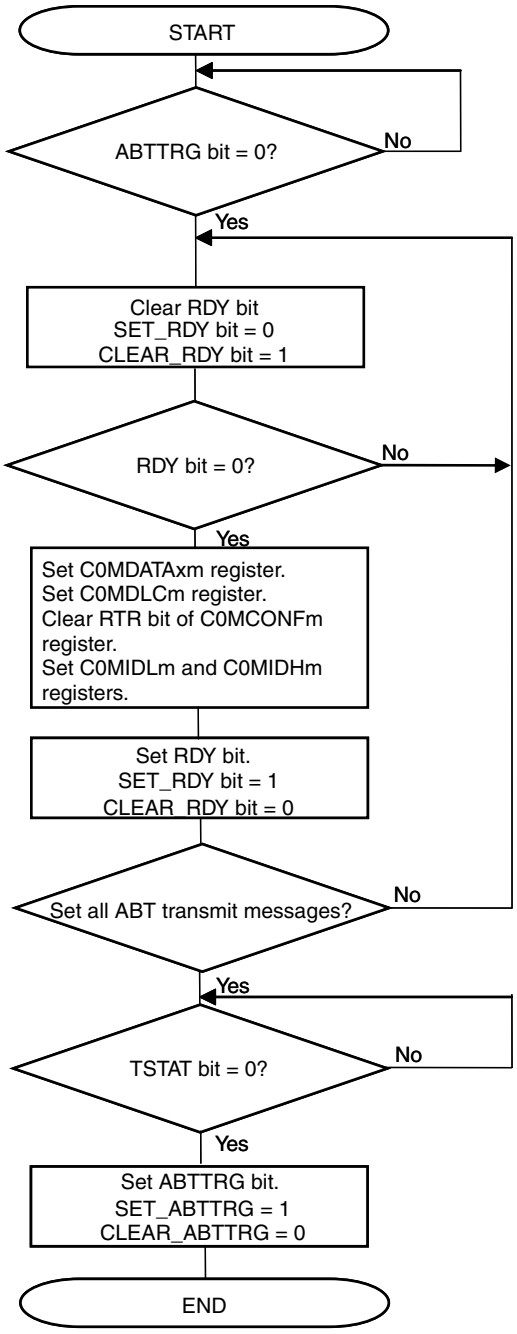
7	6	5	4	3	2	1	0
MDATA6 7	MDATA6 6	MDATA6 5	MDATA6 4	MDATA6 3	MDATA6 2	MDATA6 1	MDATA6 0

C0MDATA7m

7	6	5	4	3	2	1	0
MDATA7 7	MDATA7 6	MDATA7 5	MDATA7 4	MDATA7 3	MDATA7 2	MDATA7 1	MDATA7 0

Figure 20-42 shows the processing for a transmit message buffer (C0MCONFm.MT2 to C0MCONFm.MT0 bits = 000B).

Figure 20-42. ABT Message Transmit Processing



Caution The ABTTRG bit should be set to 1 after the TSTAT bit is cleared to 0. The checking of the TSTAT bit and the setting for the ABTTRG bit to 1 must be continuous.

Remark This processing (message transmit processing with ABS) can only be applied to message buffers 0 to 7. For message buffers other than the ABT message buffers, refer to **Figure 20-41**.

21.6.3 EPC control registers

(1) UF0 EP0NAK register (UF0E0N)

This register controls NAK of Endpoint0 (except an automatically executed request).

This register can be read or written in 8-bit units (however, bit 0 can only be read).

It takes five USB clocks to reflect the status on this register after the UF0FIC0 and UF0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate a write signal that accesses the UF0FIC0 and UF0FIC1 registers from a read signal that accesses the UF0EPS0, UF0EPS1, UF0EPS2, UF0E0N, and UF0EN registers by at least four USB clocks.

While NAK is being transmitted to Endpoint0 Read, Endpoint2, and Endpoint4, a write access to the EP0NKR bit is ignored.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0N	0	0	0	0	0	0	EP0NKR	EP0NKW	00200000H	00H

Bit position	Bit name	Function
1	EP0NKR	This bit controls NAK to the OUT token to Endpoint0 (except an automatically executed request). It is automatically set to 1 by hardware when Endpoint0 has correctly received data. It is also cleared to 0 by hardware when the data of the UF0E0R register has been read by FW (counter value = 0). 1: Transmit NAK. 0: Do not transmit NAK (default value). Set this bit to 1 by FW when data should not be received from the USB bus for some reason even when USBF is ready for receiving data. In this case, USBF continues transmitting NAK until this bit is cleared to 0 by FW. This bit is also cleared to 0 as soon as the UF0E0R register has been cleared.
0	EP0NKW	This bit indicates how NAK to the IN token to Endpoint0 is controlled (except an automatically executed request). This bit is automatically cleared to 0 by hardware when the data of Endpoint0 is transmitted and the host correctly receives the transmitted data. The data of the UF0E0W register is retained until this bit is cleared. Therefore, it is not necessary to rewrite this bit even in the case of a retransmission request that is made if the host could not receive data correctly. To send a short packet, be sure to set the E0DED bit of the UF0DEND register to 1. This bit is automatically set to 1 when the FIFO is full. As soon as the E0DED bit of the UF0DEND register is set to 1, the EP0NKW bit is automatically set to 1 at the same time. 1: Do not transmit NAK. 0: Transmit NAK (default value). If control transfer enters the status stage while ACK cannot be correctly received in the data stage, this bit is cleared to 0 as soon as the UF0E0W register is cleared. This bit is also cleared to 0 when UF0E0W is cleared by FW.

(10) UF0 EP status 2 register (UF0EPS2)

This register indicates the USB bus status and the presence or absence of register data.

This register is read-only, in 8-bit units.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1 to 4, 7) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0EPS2	0	0	HALT7	HALT4	HALT3	HALT2	HALT1	HALT0	00200012H	00H

Bit position	Bit name	Function
5 to 0	HALTn	These bits indicate that Endpoint n is currently stalled. They are set to 1 when a stall condition, such as occurrence of an overrun and reception of an undefined request, is satisfied. These bits are automatically set to 1 by hardware. 1: Endpoint is stalled. 0: Endpoint is not stalled (default value). The SNDSTL bit is set to 1 as soon as the HALT0 bit has been set to 1 as a result of occurrence of an overrun or reception of an undefined request. If the next SETUP token is received in this status, the SNDSTL bit is cleared to 0 and, therefore, the HALT0 bit is also cleared to 0. If Endpoint0 is stalled by the SET_FEATURE Endpoint0 request, this bit is not cleared to 0 until the CLEAR_FEATURE Endpoint0 request is received or Halt Feature is cleared by FW. If the GET_STATUS Endpoint0, CLEAR_FEATURE Endpoint0, or SET_FEATURE Endpoint0 request is received, or if a request to be processed by FW is received due to the CPUDEC interrupt request, the HALT0 bit is masked and cleared to 0, until the next SETUP token is received. The HALTn bit is not cleared to 0 until Endpoint n receives the CLEAR_FEATURE Endpoint request, Halt Feature is cleared by the SET_INTERFACE or SET_CONFIGURATION request to the interface to which the endpoint is linked, or Halt Feature is cleared by FW. When the SET_INTERFACE or SET_CONFIGURATION request is correctly processed, the Halt Feature of all the target endpoints, except Endpoint0, is cleared after the request has been processed, even if the wValue is the same as the currently set value, and these bits are also cleared to 0. Halt Feature of Endpoint0 cannot be cleared if it is set because the STALL response is made in response to the SET_INTERFACE and SET_CONFIGURATION requests.

Remark n = 0 to 4, 7, 8

(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.

(11) Read values of DSAn and DDAn registers

Values in the middle of updating may be read from the DSAn and DDAn registers during DMA transfer (n = 0 to 3). For example, if the DSAnH register and then the DSAnL register are read when the DMA transfer source address (DSAn register) is 0000FFFFH and the count direction is incremental (DADCn.SAD1 and DADCn.SAD0 bits = 00), the value of the DSAn register differs as follows, depending on whether DMA transfer is executed immediately after the DSAnH register is read.

(a) If DMA transfer does not occur while DSAn register is read

<1> Read value of DSAnH register: DSAnH = 0000H

<2> Read value of DSAnL register: DSAnL = FFFFH

(b) If DMA transfer occurs while DSAn register is read

<1> Read value of DSAnH register: DSAnH = 0000H

<2> Occurrence of DMA transfer

<3> Incrementing DSAn register: DSAn = 00100000H

<4> Read value of DSAnL register: DSAnL = 0000H

Figure 26-2. Timing of Reset Operation by $\overline{\text{RESET}}$ Pin Input

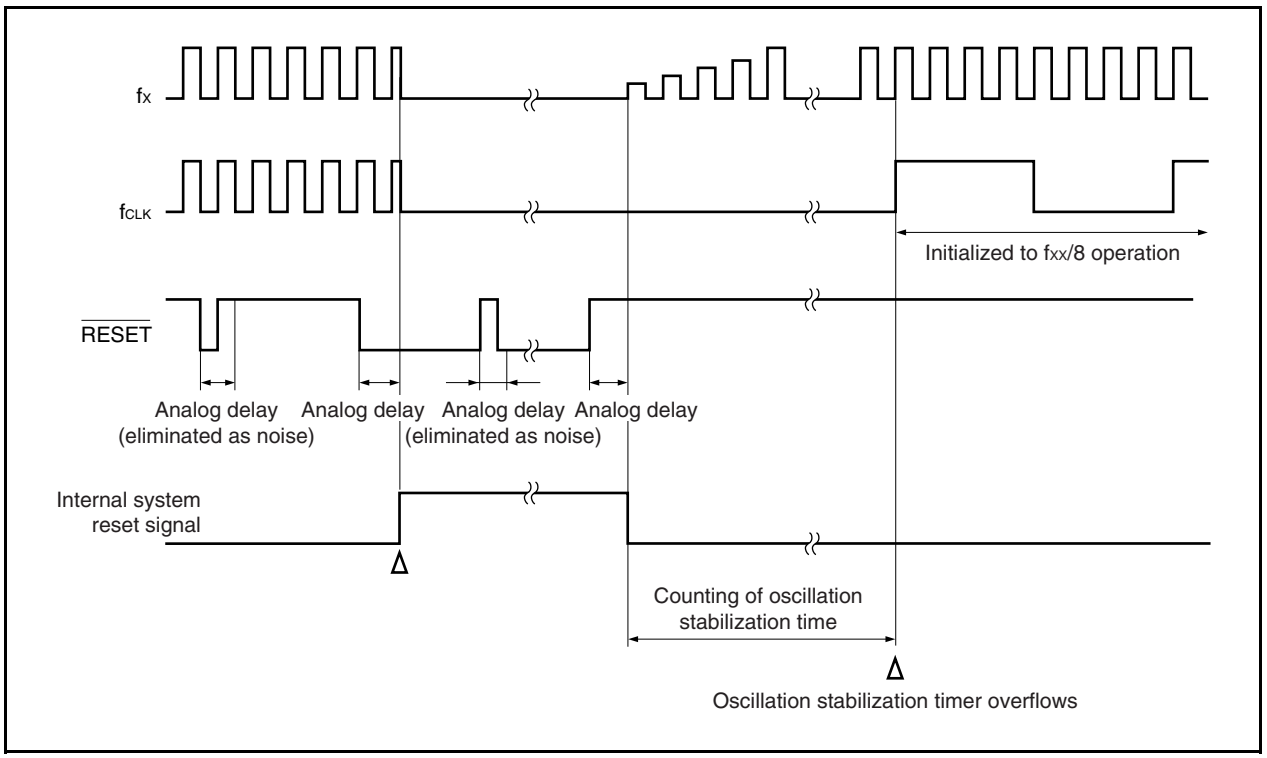
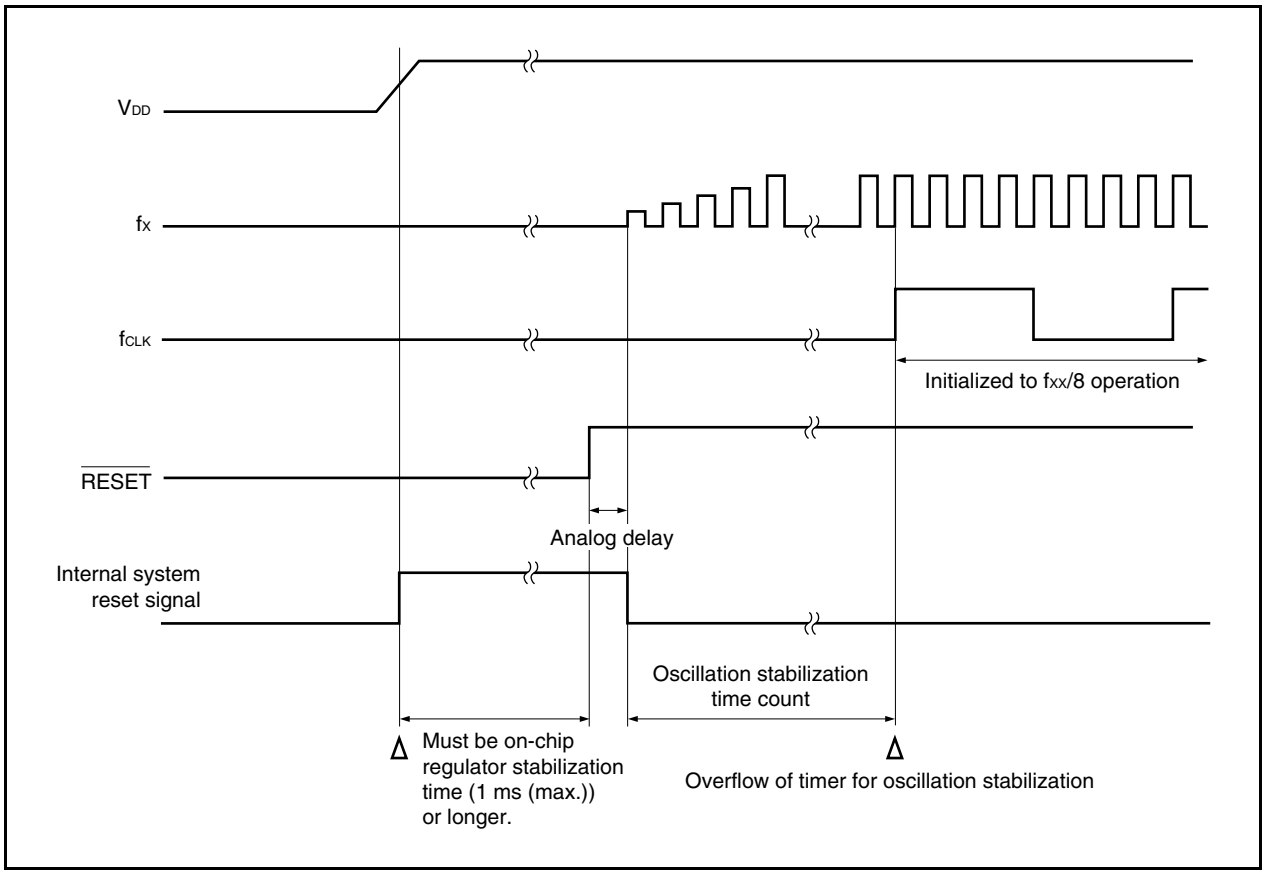


Figure 26-3. Timing of Power-on Reset Operation

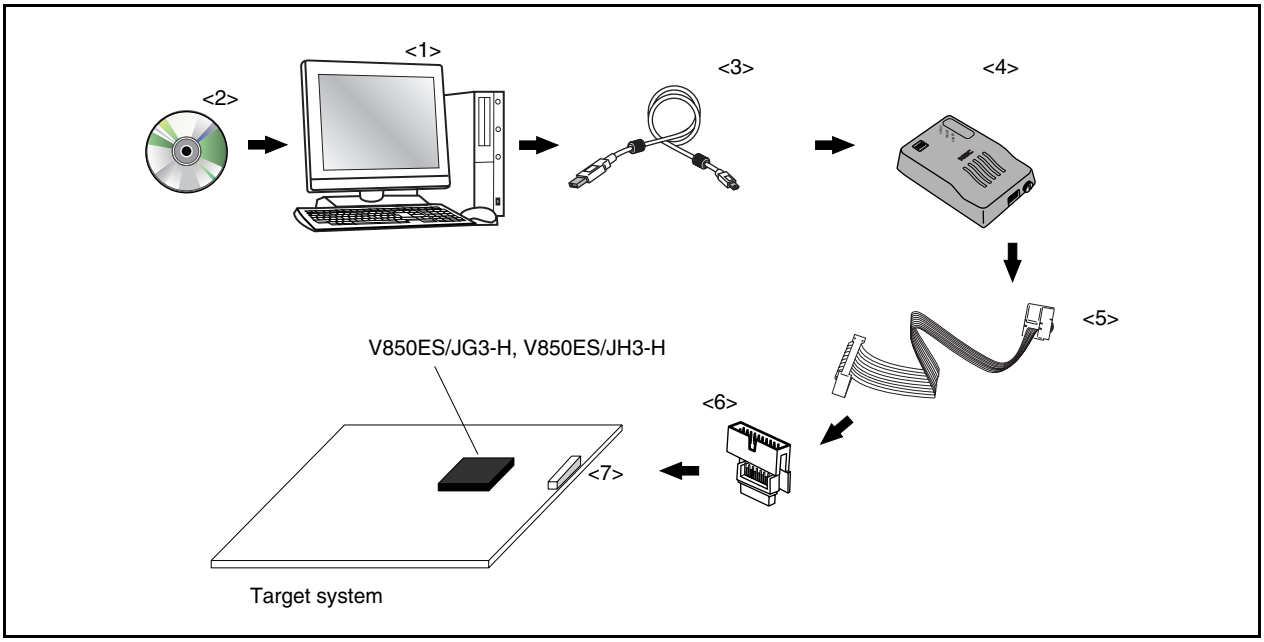


A.4.2 When using MINICUBE QB-V850MINI

(1) On-chip emulation using MINICUBE

The system configuration when connecting MINICUBE to the host machine (PC-9821 series, PC/AT compatible) is shown below.

Figure A-3. On-Chip Emulation System Configuration



<1> Host machine	PC with USB ports
<2> CD-ROM[Note 1]	Contents such as integrated debugger ID850QB, N-Wire Checker, device driver, and documents are included in CD-ROM. It is supplied with MINICUBE.
<3> USB interface cable	USB cable to connect the host machine and MINICUBE. It is supplied with MINICUBE. The cable length is approximately 2 m.
<4> MINICUBE On-chip debug emulator	This on-chip debug emulator serves to debug hardware and software when developing application systems using the V850ES/JG3-H or V850ES/JH3-H. It supports integrated debugger ID850QB.
<5> OCD cable	Cable to connect MINICUBE and the target system. It is supplied with MINICUBE. The cable length is approximately 20 cm.
<6> Connector conversion board KEL adapter	This conversion board is supplied with MINICUBE.
<7> MINICUBE connector KEL connector[Note 2]	8830E-026-170S (supplied with MINICUBE) 8830E-026-170L (sold separately)

Notes 1. Download the device file from the Renesas Electronics website.
http://www2.renesas.com/micro/en/ods/index.html

2. Product of KEL Corporation

Remark The numbers in the angular brackets correspond to the numbers in Figure A-3.