NEC

User's Manual

V850E/PH2™

32-Bit Single-Chip Microcontroller

Hardware

µPD70F3187

Document No. U16580EE2V0UD00
Date Published June 2006

Printed in Germany

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

② HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

③ PRECAUTION AGAINST ESD

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ STATUS BEFORE INITIALIZATION

Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

⑤ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

- **The information in this document is current as of June, 2006. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC 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 NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.
  "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.
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)
(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

For further information,
please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielski Strasse 166 B
30177 Hanover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52180
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
TEL: 010-8235-1155
http://www.cn.necel.com/

NEC Electronics Shanghai Ltd.
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai P.R. China P.C:200120
Tel: 021-5888-5400
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
12/F., Cityplaza 4,
12 Taikoo Wan Road, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

Seoul Branch
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

G06.6-1A

Preface

Readers

This manual is intended for users who want to understand the functions of the V850E/PH2 (PHOENIX-F).

Purpose

This manual presents the hardware manual of V850E/PH2.

Organization

This system specification describes the following sections:

- Pin function
- CPU function
- Internal peripheral function
- Flash memory

Legend

Symbols and notation are used as follows:

Weight in data notation	:	Left is high-order column, right is low order column
Active low notation	:	$\overline{xxx}$ (pin or signal name is over-scored) or /xxx (slash before signal name)
Memory map address:	:	High order at high stage and low order at low stage
Note	:	Explanation of (Note) in the text
Caution	:	Item deserving extra attention
Remark	:	Supplementary explanation to the text
Numeric notation	:	Binary... xxxx or xxxB Decimal... xxxx Hexadecimal... xxxxH or 0x xxxx

Prefixes representing powers of 2 (address space, memory capacity)

K (kilo): $2^{10} = 1024$

M (mega): $2^{20} = 1024^2 = 1,048,576$

G (giga): $2^{30} = 1024^3 = 1,073,741,824$

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

The V850E/PH2 (PHOENIX-F[Note]) is a product of the NEC Electronics single-chip microcontrollers "V850 series™". This chapter gives a short outline of the V850E/PH2 microcontroller.

1.1 Outline

The V850E/PH2 is a 32-bit single-chip microcontroller that realizes high-precision inverter control of a motor due to high-speed operation. It uses the V850E1 CPU (NU85EFC) of the V850 Series including single-precision floating point unit, and has on-chip ROM, RAM, bus interface, DMA controller, a real-time pulse unit including 3-phase PWM timer for inverter control, various serial interfaces including AFCAN, and peripheral facilities such as A/D converters, as well as an on-chip debug interface.

(1) V850E1 CPU

The V850E1 CPU (NU85EFC) supports a RISC instruction set that enhances the performance of the V850 CPU, which is the CPU core integrated in the V850 Series, and has added instructions supporting high-level languages, such as C-language switch statement processing, table look-up branching, stack frame creation/deletion, and data conversion. This enhances the performance of both data processing and control. It is possible to use the software resources of the V850 CPU integrated system since the instruction codes of the V850E1 are upwardly compatible at the object code level with those of the V850 CPU.
In addition, the V850E1 CPU (NU85EFC) incorporates a single-precision floating point unit, which supports high speed floating point arithmetic operations.

(2) External memory interface function

The V850E/PH2 microcontroller features n on-chip external memory interface including separately configured address (22 bits) and data (32 bits) buses. SRAM and ROM can be connected.

(3) On-chip flash memory

The V850E/PH2 microcontroller has a quickly accessible flash memory on-chip, that can shorten system development time since it is possible to rewrite a program with the V850E/PH2 microcontroller mounted in an application system. Moreover, it can greatly improve maintain ability after system ships.

(4) A full range of development environment products

A development environment system that includes an optimized C compiler, debugger, in-circuit emulator, simulator, system performance analyser, and other elements is also available.

Note: PHOENIX-F is the European name of the V850E/PH2 microcontroller.

1.2 Device Features

- Number of instructions: 96
- Instruction execution time: 15.625 ns (@ ϕ = 64 MHz)
- General-purpose registers: 32 bits × 32
- Instruction set:
  V850E1 CPU (NU85EFC)
  (compatible with V850 plus additional powerful instructions for reducing code and increasing execution speed)
  Single-precision floating point arithmetic operation
  Signed multiplication (16 bits × 16 bits →32 bits or 32 bits × 32 bits →64 bits): 1 to 2 clocks
  Saturated operation instructions (with overflow/underflow detection function)
  32-bit shift instructions: 1 clock
  Bit manipulation instructions
  Load/store instructions with long/short format
  Signed load instructions
- Memory space:
  64 MB linear address space (common program/data)
  Chip select output function: 4 spaces
  Memory block division function: 2, 4, or 8 MB/block
  Programmable wait function
  Idle state insertion function
- External bus interface:
  32-bit data bus (address/data separated)
  22-bit address bus
  4 programmable chip select areas
  32-/16-/8-bit bus sizing function
  External wait function
- Internal memory:
  Flash ROM: 512 KB
  RAM: 32 KB
- Interrupts/exceptions:

External interrupts:	14 (including NMI)
Internal interrupts:	85 sources
Exceptions:	1 source

  8 programmable interrupt priority levels
- Memory access controller: SRAM controller
- DMA controller:
  8 channels
  Transfer mode: Single transfer
  Transfer units: 8 bits or 16 bits (depending on peripheral)
  Maximum transfer count: 256 (2^8)
  Transfer target: internal RAM ↔ I/O
  Transfer request: On-chip peripheral I/O
  DMA transfer termination interrupt
- I/O lines:

Input ports:	5
I/O ports:	137

- Timer: 16-bit timer for 3-phase PWM inverter control: 2 channels
  16-bit up/down counter for 4-quadrant encoding: 1 channel
  16-bit general purpose timers: 9 channels
  16-bit general purpose timers with encoding capability: 2 channels
- Serial interfaces: Asynchronous serial interface (UARTC): 2 channels
  Clocked serial interface (CSIB): 2 channels
  Queued clocked serial interface (CSI3): 2 channels
  FCAN interface (AFCAN): 2 channels
- A/D converters: 10-bit resolution
  2 × 10 channels
- Random number generator: Automatic seed generation
  Fips/Maurer test passing
- Clock generator: 16 MHz clock oscillator
  4 fold PLL synthesizer for internal system clock
- Power save modes: HALT mode
- Auxiliary frequency output: Programmable by user software
- Supply voltage: 1.5 V (internal power supply, clock generator)
  3.3 V (external I/O pins, A/D converter)
- Package: 208-pin plastic LQFP (fine pitch) (28 × 28)
  256-pin plastic BGA (21 × 21)
- CMOS technology

1.3 Applications

The V850E/PH2 microcontroller is ideally suited for automotive applications, like electrical power steering and electric car control. It is also an excellent choice for other applications where a combination of general-purpose inverter control functions and CAN network support is required.

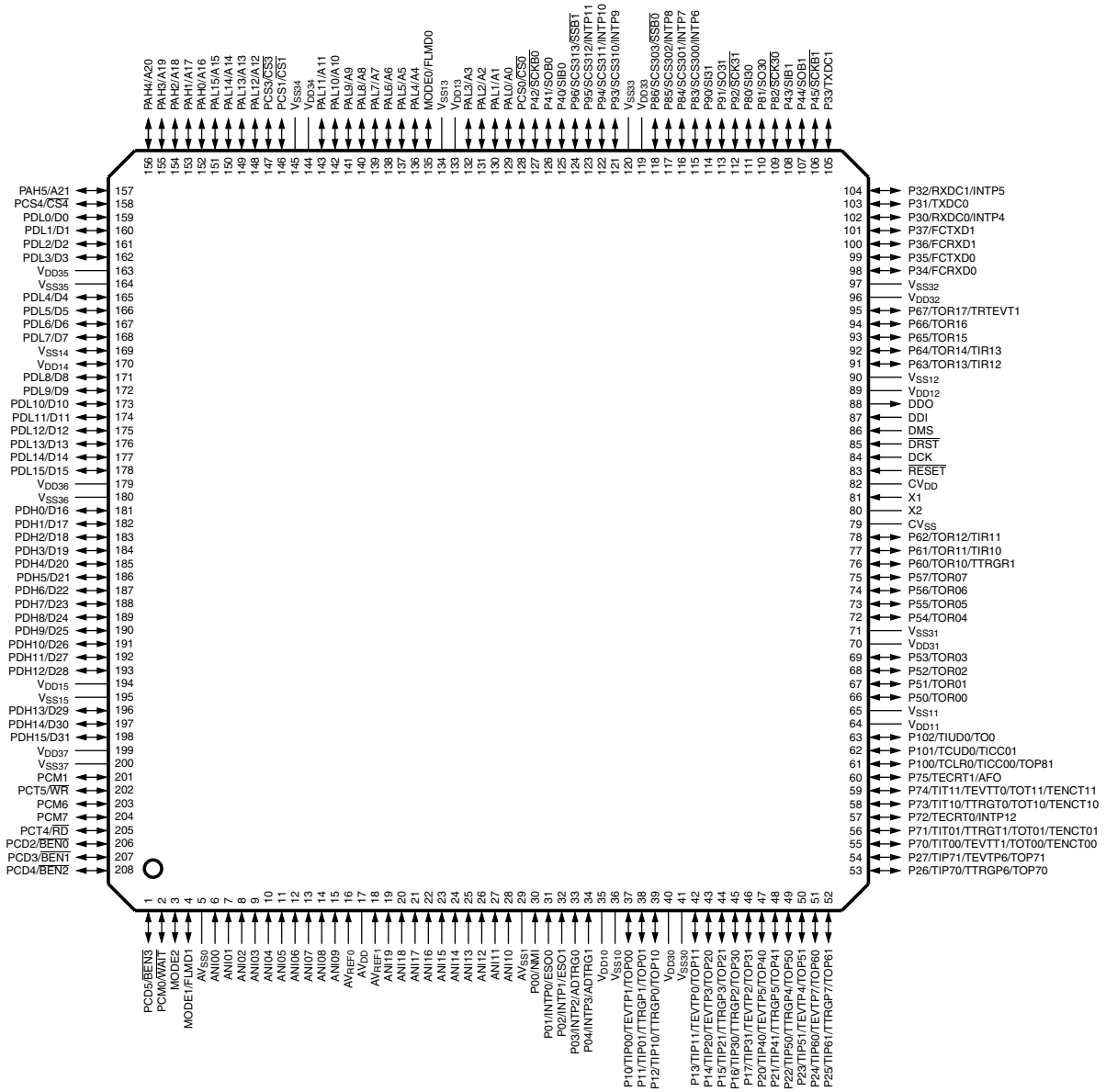
1.4 Ordering Information

Part Number	Package
μPD70F3187GD-64-LML	208-pin plastic LQFP (fine pitch) (28 × 28)
μPD70F3187GD(A1)-64-LML	208-pin plastic LQFP (fine pitch) (28 × 28)
μPD70F3187GD(A2)-64-LML	208-pin plastic LQFP (fine pitch) (28 × 28)
μPD70F3187F1(A2)-64-JN4	256-pin plastic BGA (21 × 21)

1.5 Pin Configuration (Top View)

- **208-pin plastic LQFP (fine pitch) (28 × 28)**
  μPD70F3187GD-64-LML
  μPD70F3187GD(A1)-64-LML
  μPD70F3187GD(A2)-64-LML

Figure 1-1: Pin Configuration 208-pin Plastic LQFP



- **256-pin plastic BGA (21 × 21)**

µPD70F3187F1(A2)-JN4

Figure 1-2: Pin Configuration 256-pin Plastic BGA (21 × 21)

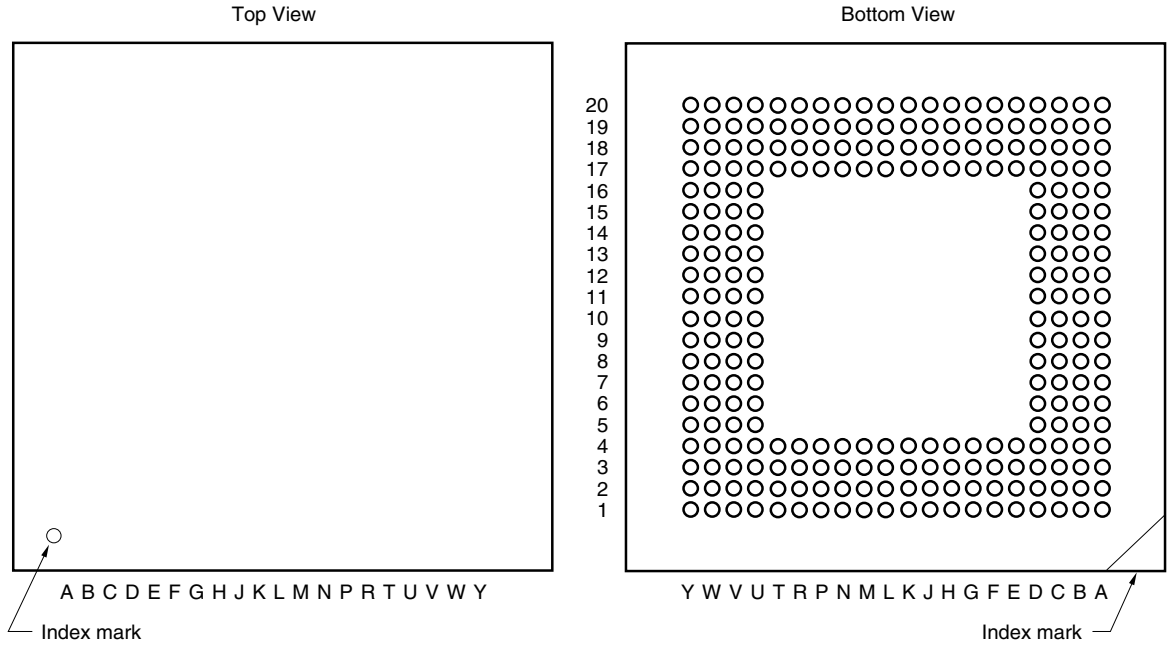


Table 1-1: 256-pin Plastic BGA (1/2)

Pin No	Pin Name	Pin No	Pin Name	Pin No	Pin Name	Pin No	Pin Name
A1	NC	B1	NC	C1	AV_{SS0}	D1	AV_{SS0}
A2	NC	B2	NC	C2	MODE1	D2	AV_{SS0}
A3	PCT4/$\overline{RD}$	B3	PCD5/$\overline{BEN3}$	C3	MODE2	D3	AV_{SS0}
A4	PCT5/$\overline{WR}$	B4	PCD2/$\overline{BEN0}$	C4	PCD4/$\overline{BEN2}$	D4	PCM0/$\overline{WAIT}$
A5	PDH15/D31	B5	PCM6	C5	PCM7	D5	PCD3/$\overline{BEN1}$
A6	PDH13/D29	B6	PCM1	C6	V_{SS37}	D6	V_{DD37}
A7	PDH11/D27	B7	PDH14/D30	C7	V_{SS37}	D7	V_{DD37}
A8	PDH9/D25	B8	PDH12/D28	C8	V_{SS15}	D8	V_{DD15}
A9	PDH8/D24	B9	PDH10/D26	C9	V_{SS15}	D9	V_{DD15}
A10	PDH6/D22	B10	PDH7/D23	C10	PDH4/D20	D10	PDH2/D18
A11	PDH3/D19	B11	PDH5/D21	C11	V_{SS36}	D11	V_{DD36}
A12	PDH0/D16	B12	PDH1/D17	C12	V_{SS36}	D12	V_{DD36}
A13	PDL15/D15	B13	PDL13/D13	C13	PDL11/D11	D13	PDL10/D10
A14	PDL14/D14	B14	PDL12/D12	C14	V_{SS14}	D14	V_{DD14}
A15	PDL9/D9	B15	PDL8/D8	C15	PDL7/D7	D15	PDL6/D6
A16	PDL5/D5	B16	PDL4/D4	C16	V_{SS35}	D16	V_{DD35}
A17	PDL1/D1	B17	PDL3/D3	C17	V_{SS35}	D17	V_{DD35}
A18	PDL0/D0	B18	PCS4/$\overline{CS4}$	C18	PDL2/D2	D18	PAH4/A20
A19	NC	B19	NC	C19	PAH5/A21	D19	PAH3/A19
A20	NC	B20	NC	C20	NC	D20	PAL14/A14
E1	ANI00	F1	ANI03	G1	ANI07	H1	ANI18
E2	ANI02	F2	ANI06	G2	ANI09	H2	ANI19
E3	ANI01	F3	ANI05	G3	ANI08	H3	AV_{DD}
E4	AV_{SS0}	F4	ANI04	G4	AV_{REF0}	H4	AV_{REF1}
E17	PAH0/A16	F17	PAL12/A12	G17	V_{DD34}	H17	V_{DD34}
E18	PAH2/A18	F18	PAL15/A15	G18	V_{SS34}	H18	V_{SS34}
E19	PAH1/A17	F19	PAL13/A13	G19	PCS1/$\overline{CS1}$	H19	PAL10/A10
E20	PCS3/$\overline{CS3}$	F20	PAL11/A11	G20	PAL9/A9	H20	PAL6/A6
J1	ANI17	K1	ANI13	L1	AV_{SS1}	M1	P01/INTP0/ESO0
J2	ANI14	K2	ANI10	L2	AV_{SS1}	M2	P00/NMI
J3	ANI15	K3	ANI11	L3	AV_{SS1}	M3	V_{SS10}
J4	ANI16	K4	ANI12	L4	AV_{SS1}	M4	V_{DD10}
J17	PAL5/A5	K17	V_{DD13}	L17	V_{DD13}	M17	P95/SCS312/INTP11
J18	PAL8/A8	K18	V_{SS13}	L18	V_{SS13}	M18	PAL0/A0
J19	PAL7/A7	K19	PAL4/A4	L19	PAL3/A3	M19	PCS0/$\overline{CS0}$
J20	MODE0	K20	PAL2/A2	L20	PAL1/A1	M20	P42/$\overline{SCKB0}$

Table 1-1: 256-pin Plastic BGA (2/2)

Pin No	Pin Name	Pin No	Pin Name	Pin No	Pin Name	Pin No	Pin Name
N1	P02/INTP1/ESO1	P1	P04/INTP3/ ADTRG1	R1	P11/TIP01/ TTRGP1/TOP01	T1	P13/TIP11/TEVTP0/ TOP11
N2	P03/INTP2/ ADTRG0	P2	P10/TIP00/ TEVTP1/TOP00	R2	P12/TIP10/ TTRGP0/TOP10	T2	P14/TIP20/TEVTP3/ TOP20
N3	V_{SS10}	P3	V_{SS30}	R3	V_{SS30}	T3	P16/TIP30/TTRGP2/ TOP30
N4	V_{DD10}	P4	V_{DD30}	R4	V_{DD30}	T4	P21/TIP41/TTRGP5/ TOP41
N17	V_{DD33}	P17	V_{DD33}	R17	P83/SCS300/INTP6	T17	P80/SI30
N18	V_{SS33}	P18	V_{SS33}	R18	P86/SCS303/$\overline{SSB0}$	T18	P84/SCS301/INTP7
N19	P41/SOB0	P19	P96/SCS313/$\overline{SSB1}$	R19	P93/SCS310/INTP9	T19	P90/SI31
N20	P40/SIB0	P20	P94/SCS311/ INTP10	R20	P85/SCS302/INTP8	T20	P91/SO31
U1	P15/TIP21/ TTRGP3/TOP21	V1	P20/TIP40/ TEVTP5/TOP40	W1	NC	Y1	NC
U2	P17/TIP31/ TEVTP2/TOP31	V2	P23/TIP51/ TEVTP4/TOP51	W2	P26	Y2	NC
U3	P22/TIP50/ TTRGP4/TOP50	V3	P24/TIP60/ TEVTP7/TOP60	W3	P27/TIP71/ TEVTP6/TOP71	Y3	P72/TECRT0/ INTP12
U4	P25/TIP61/ TTRGP7/TOP61	V4	P70/TIT00/TEVTT1/ TOT00/TENCT00	W4	P73/TIT10/TTRGT0/ TOT10/TENCT10	Y4	P100/TCLR1/ TICC10/TOP81
U5	P71/TIT01/TTRGT1/ TOT01/TENCT01	V5	P74/TIT11/TEVTT0/ TOT11/TENCT11	W5	P101/TCUD1/ TICC11	Y5	P50/TOR00
U6	P75	V6	P102/TIUD1/TO1	W6	P51/TOR01	Y6	P52/TOR02
U7	V_{DD11}	V7	V_{SS11}	W7	P53/TOR03	Y7	P54/TOR04
U8	V_{DD31}	V8	V_{SS31}	W8	P55/TOR05	Y8	P56/TOR06
U9	P62/TOR12/TIR11	V9	P61/TOR11/TIR10	W9	P57/TOR07	Y9	P60/TOR10/TTRGR1
U10	V_{SS31}	V10	V_{SS31}	W10	V_{SS31}	Y10	V_{SS31}
U11	DCK	V11	$\overline{RESET}$	W11	X2	Y11	CV_{SS}
U12	V_{DD12}	V12	V_{SS12}	W12	X1	Y12	CV_{DD}
U13	V_{DD12}	V13	V_{SS12}	W13	DMS	Y13	$\overline{DRST}$
U14	V_{DD32}	V14	V_{SS32}	W14	DDO	Y14	DDI
U15	V_{DD32}	V15	V_{SS32}	W15	P65/TOR15	Y15	P63/TOR13/TIR12
U16	P32/RXDC1/INTP5	V16	P67/TOR17/ TEVTR1	W16	P66/TOR16	Y16	P64/TOR14/TIR13
U17	P81/SO30	V17	P31/TXDC0	W17	P34/FCRXD0	Y17	P35/FCTXD0
U18	P82/$\overline{SCK30}$	V18	P30/RXDC0/INTP4	W18	P36/FCRXD1	Y18	P37/FCTXD1
U19	P44/SOB1	V19	P43/SIB1	W19	P33/TXDC1	Y19	NC
U20	P92/$\overline{SCK31}$	V20	P45/$\overline{SCKB1}$	W20	NC	Y20	NC

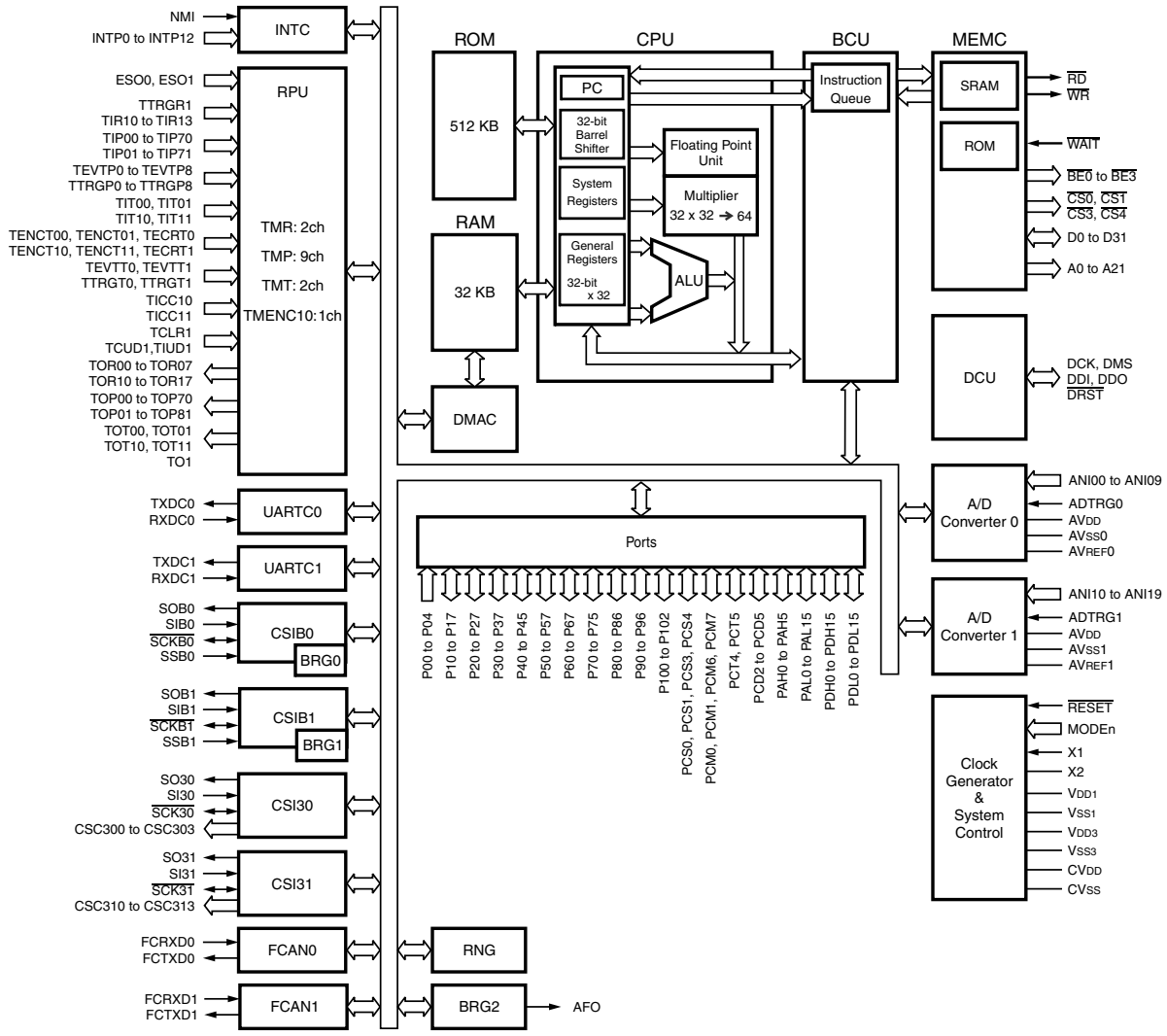
Pin Identification

A0 to A21: Address bus
ADTRG0, ADTRG1: A/D trigger input
AFO: Auxiliary frequency output
ANI00 to ANI09, ANI10 to ANI19: Analog input
AV_{DD}: Analog power supply
AV_{REF0}, AV_{REF1}: Analog reference voltage
AV_{SS0}, AV_{SS1}: Analog ground
$\overline{BEN0}$ to $\overline{BEN3}$: Byte enable
$\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$, $\overline{CS4}$: Chip select
CV_{DD}: Power supply for oscillator
CV_{SS}: Oscillator ground
D0 to D31: Data bus
DCK: Debug clock input
DDI: Debug data input
DDO: Debug data output
DMS: Debug mode select
$\overline{DRST}$: Debug reset
ESO0, ESO1: Emergency shut-off
FCRXD0, FCRXD1: FCAN receive data input
FCTXD0, FCTXD1: FCAN transmit data output
INTP0 to INTP12: External interrupt request
MODE0 to MODE2: Mode
NMI: Non-maskable interrupt request
NC: Not connected
P00 to P04: Port 0
P10 to P17: Port 1
P20 to P27: Port 2
P30 to P37: Port 3
P40 to P45: Port 4
P50 to P57: Port 5
P60 to P67: Port 6
P70 to P75: Port 7
P80 to P86: Port 8
P90 to P96: Port 9
P100 to P102: Port 10
PAL0 to PAL15: Port AL
PAH0 to PAH5: Port AH
PCD2 to PCD5: Port CD
PCM0, PCM1, PCM6, PCM7: Port CM
PCS0, PCS1, PCS3, PCS4: Port CS
PCT4, PCT5: Port CT
PDL0 to PDL15: Port DH
PDH0 to PDH15: Port DL
$\overline{RD}$: Read strobe
$\overline{RESET}$: Reset
RXDC0, RXDC1: Receive data input
$\overline{SCK30}$, $\overline{SCK31}$, $\overline{SCKB0}$, $\overline{SCKB1}$: Serial clock
SCS300 to SCS303, SCS310 to SCS313: Serial chip select
SI30, SI31, SIB0, SIB1: Serial data input
SO30, SO31, SOB0, SOB1: Serial data output
SSB0, SSB1: Serial slave select input
TCLR1: Timer clear
TCUD1: Timer control pulse input
TECRT0, TECRT1: Timer external clear
TENCT00, TENCT01, TENCT10, TENCT11: Timer encoder input
TEVTP0 to TEVTP7, TEVTR1, TEVTT0, TEVTT1: Timer event input
TICC10, TICC11
TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31, TIP40, TIP41, TIP50, TIP51, TIP60, TIP61, TIP70, TIP71, TIR10 to TIR13, TIT00, TIT01, TIT10, TIT11: Timer input
TIUD1: Timer count pulse input
TO1, TOP00, TOP01, TOP10, TOP11, TOP20, TOP21, TOP30, TOP31, TOP40, TOP41, TOP50, TOP51, TOP60, TOP61, TOP70, TOP71, TOP81, TOR00 to TOR07, TOR10 to TOR17, TOT00, TOT01, TOT10, TOT11: Timer output
TTRGP0 to TTRGP7, TTRGR1, TTRGT0, TTRGT1: Timer trigger input
TXDC0, TXDC1: Transmit data output
V_{DD10} to V_{DD15}: Power supply for CPU
V_{DD30} to V_{DD37}: I/O buffers power supply
V_{SS10} to V_{SS15}: CPU Ground
V_{SS30} to V_{SS37}: I/O buffers ground
$\overline{WAIT}$: Wait
$\overline{WR}$: Write strobe
X1, X2: Crystal

1.6 Function Blocks

1.6.1 Internal block diagram

Figure 1-3: Internal Block Diagram



1.6.2 On-chip units

(1) CPU

The CPU uses five-stage pipeline control to enable single-clock execution of address calculations, arithmetic logic operations, data transfers, and almost all other instruction processing.
Other dedicated on-chip hardware, such as a multiplier
(16 bits × 16 bits →32 bits or 32 bits × 32 bits →64 bits) and a barrel shifter (32 bits), help accelerate processing of complex instructions.

(2) Bus control unit (BCU)

The BCU starts the required external bus cycle based on the physical address obtained by the CPU. When an instruction is fetched from external memory area and the CPU does not send a bus cycle start request, the BCU generates a prefetch address and prefetches the instruction code. The prefetched instruction code is stored in an instruction queue in the CPU.
The BCU controls a memory controller (MEMC) and DMA controller (DMAC) and performs external memory access and DMA transfer.

(a) Memory controller (MEMC)

The MEMC controls SRAM, ROM, and various I/O for external memory expansion.

- SRAM, external ROM, external I/O interface

Supports access to SRAM, external ROM, and external I/O.

(b) DMA controller (DMAC)

The DMAC performs data transfers b/w internal on-chip RAM and peripheral I/O. For this purpose eight DMA channels are provided for particular transfer functions of serial I/O interfaces, real-time pulse unit (TMR), and A/D converter.

(3) ROM

There is on-chip flash memory (512 KB) in the V850E/PH2 provided.
On an instruction fetch, the ROM can be accessed by the CPU in one clock.
When single-chip mode 0 or flash memory programming mode is set, ROM is mapped starting from address 00000000H.
When single-chip mode 1 is set, it is mapped starting from address 00100000H.
ROM cannot be accessed if ROM-less mode is set.

(4) RAM

On-chip RAM is mapped starting from address 03FF0000H.
It can be accessed by the CPU in one clock on an instruction fetch or data access.

(5) Interrupt controller (INTC)

The INTC services hardware interrupt requests from on-chip peripheral I/O and external sources (NMI, INTP0 to INTP12). Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple-interrupt servicing control can be performed for interrupt sources

(6) Clock generator (CG)

The CG provides a frequency that is 4 times the input clock (f_X) (using the on-chip PLL) as the internal system clock (f_{CPU}). As the input clock, connect an external crystal or resonator to pins X1 and X2 or input an external clock from the X1 pin.

(7) Real-time pulse unit (RPU)

The RPU incorporates a 2-channel 16-bit timer (TMR) for 3/6-phase sine wave PWM inverter control, an 1-channel 16-bit up/down counter (TMENC10) and a 2-channel 16-bit up/down counter (TMT) that can be used for 2-phase encoder input or as a general-purpose timer, a 9-channel 16-bit general-purpose timer unit (TMP).
The RPU can measure pulse interval or frequency and can output programmable pulses.

(8) Serial interface (SIO)

The serial interfaces consist of 2 channels asynchronous serial interface C (UARTC), 2 channels clocked serial interface B (CSIB), 2 channels clocked serial interface 3 (CSI3), and 2 channels FCAN interface (AFCAN).
The UARTC performs data transfer using pins TXDCn and RXDCn (n = 0, 1).
The CSIB performs data transfer using pins SOBn, SIBn, $\overline{\text{SCKBn}}$, SSIn, and SSOn (n = 0, 1).
The CSI3 performs data transfer using pins SO3n, SI3n, $\overline{\text{SCK3n}}$, SCS3n0 to SCS3 (n = 0, 1).
The AFCAN performs data transfer using pins FCTXDn and FCRXDn (n = 0, 1).

(9) Baud rate generator (BRG)

The baud rate generator comprises 3 channels of 8-bit counters and comparators that can be used for clock supply of serial interfaces (CSIB), auxiliary frequency output (AFO) or interval timer.

(10) A/D converter (ADC)

The two units of high-speed, high-resolution 10-bit A/D converter include 10 analog input pins for each unit. Conversion is performed using the successive approximation method.

(11) Random number generator (RNG)

For encryption purpose a random number generator is provided.

(12) Debug control unit (DCU)

On-chip debugging can be performed via a debug control unit (n-wire interface).

(13) Ports

As shown below, the following ports have general-purpose port functions and control pin functions.

Port	I/O	Control Function
Port 0	5-bit input	NMI input, external interrupt input, A/D converter external trigger input, emergency shut-off input
Port 1	8-bit I/O	Real-time pulse unit I/O
Port 2	8-bit I/O	Real-time pulse unit I/O
Port 3	8-bit I/O	Serial interface I/O, external interrupt input
Port 4	6-bit I/O	Serial interface I/O
Port 5	8-bit I/O	Real-time pulse unit I/O
Port 6	8-bit I/O	Real-time pulse unit I/O
Port 7	6-bit I/O	Real-time pulse unit I/O, external interrupt input
Port 8	7-bit I/O	Serial interface I/O, external interrupt input
Port 9	7-bit I/O	Serial interface I/O, external interrupt input
Port 10	3-bit I/O	Real-time pulse unit I/O
Port AL	16-bit I/O	External address bus
Port AH	6-bit I/O	External address bus
Port DL	16-bit I/O	External data bus
Port DH	16-bit I/O	External data bus
Port CD	4-bit I/O	External bus interface control signal output
Port CM	4-bit I/O	Wait insertion signal input
Port CS	4-bit I/O	External bus interface control signal output
Port CT	2-bit I/O	External bus interface control signal output

Chapter 2 Pin Functions

2.1 List of Pin Functions

The names and functions of the V850E/PH2 microcontroller pins are listed below. These pins can be divided into port pins and non-port pins according to their functions.

(1) Port pins

Table 2-1: Port Pins (1/5)

Pin Name	I/O	Function	Alternate Function
P00	I	Port 0 5-bit input-only port	NMI
P01			INTP0, ESO0
P02			INTP1, ESO1
P03			INTP2, ADTRG0
P04			INTP3, ADTRG1
P10	I/O	Port 1 8-bit I/O port Input or output direction can be specified in 1-bit units	TIP00, TEVTP1, TOP00
P11			TIP01, TTRGP1, TOP01
P12			TIP10, TTRGP0, TOP10
P13			TIP11, TEVTP0, TOP11
P14			TIP20, TEVTP3, TOP20
P15			TIP21, TTRGP3, TOP21
P16			TIP30, TTRGP2, TOP30
P17			TIP31, TEVTP2, TOP31
P20	I/O	Port 2 8-bit I/O port Input or output direction can be specified in 1-bit units	TIP40, TEVTP5, TOP40
P21			TIP41, TTRGP5, TOP41
P22			TIP50, TTRGP4, TOP50
P23			TIP51, TEVTP4, TOP51
P24			TIP60, TEVTP7, TOP60
P25			TIP61, TTRGP7, TOP61
P26			TIP70, TTRGP6, TOP70
P27			TIP71, TEVTP6, TOP71
P30	I/O	Port 3 8-bit I/O port Input or output direction can be specified in 1-bit units	RXDC0, INTP4
P31			TXDC0
P32			RXDC1, INTP5
P33			TXDC1
P34			FCRXD0
P35			FCTXD0
P36			FCRXD1
P37			FCTXD1

Table 2-1: Port Pins (2/5)

Pin Name	I/O	Function	Alternate Function
P40	I/O	Port 4 6-bit I/O port Input or output direction can be specified in 1-bit units	SIB0
P41			SOB0
P42			$\overline{\text{SCKB0}}$
P43			SIB1
P44			SOB1
P45			$\overline{\text{SCKB1}}$
P50	I/O	Port 5 8-bit I/O port Input or output direction can be specified in 1-bit units	TOR00
P51			TOR01
P52			TOR02
P53			TOR03
P54			TOR04
P55			TOR05
P56			TOR06
P57			TOR07
P60	I/O	Port 6 8-bit I/O port Input or output direction can be specified in 1-bit units	TOR10, TTRGR1
P61			TOR11, TIR10
P62			TOR12, TIR11
P63			TOR13, TIR12
P64			TOR14, TIR13
P65			TOR15
P66			TOR16
P67			TOR17, TEVTR1
P70	I/O	Port 7 6-bit I/O port Input or output direction can be specified in 1-bit units	TIT00, TEVTT1, TOT00, TENCT00
P71			TIT01, TTRGT1, TOT01, TENCT01
P72			TECRT0, INTP12
P73			TIT10, TTRGT0, TOT10, TENCT10
P74			TIT11, TEVTT0, TOT11, TENCT11
P75			TECRT1, AFO
P80	I/O	Port 8 7-bit I/O port Input or output direction can be specified in 1-bit units	SI30
P81			SO30
P82			$\overline{\text{SCK30}}$
P83			SCS300, INTP6
P84			SCS301, INTP7
P85			SCS302, INTP8
P86			SCS303, $\overline{\text{SSB0}}$

Table 2-1: Port Pins (3/5)

Pin Name	I/O	Function	Alternate Function
P90	I/O	Port 9 7-bit I/O port Input or output direction can be specified in 1-bit units	SI31
P91			SO31
P92			$\overline{\text{SCK31}}$
P93			SCS310, INTP9
P94			SCS311, INTP10
P95			SCS312, INTP11
P96			SCS313, $\overline{\text{SSB1}}$
P100	I/O	Port 10 3-bit I/O port Input or output direction can be specified in 1-bit units	TCLR1, TICC10, TOP81
P101			TCUD1, TICC11
P102			TIUD1, TO1
PAL0	I/O	Port AL 16-bit I/O port Input or output direction can be specified in 1-bit units	A0
PAL1			A1
PAL2			A2
PAL3			A3
PAL4			A4
PAL5			A5
PAL6			A6
PAL7			A7
PAL8			A8
PAL9			A9
PAL10			A10
PAL11			A11
PAL12			A12
PAL13			A13
PAL14			A14
PAL15			A15
PAH0	I/O	Port AH 6-bit I/O port Input or output direction can be specified in 1-bit units	A16
PAH1			A17
PAH2			A18
PAH3			A19
PAH4			A20
PAH5			A21

Table 2-1: Port Pins (4/5)

Pin Name	I/O	Function	Alternate Function
PDL0	I/O	Port DL 16-bit I/O port Input or output direction can be specified in 1-bit units	D0
PDL1			D1
PDL2			D2
PDL3			D3
PDL4			D4
PDL5			D5
PDL6			D6
PDL7			D7
PDL8			D8
PDL9			D9
PDL10			D10
PDL11			D11
PDL12			D12
PDL13			D13
PDL14			D14
PDL15			D15
PDH0	I/O	Port DH 16-bit I/O port Input or output direction can be specified in 1-bit units	D16
PDH1			D17
PDH2			D18
PDH3			D19
PDH4			D20
PDH5			D21
PDH6			D22
PDH7			D23
PDH8			D24
PDH9			D25
PDH10			D26
PDH11			D27
PDH12			D28
PDH13			D29
PDH14			D30
PDH15			D31
PCD2	I/O	Port CD 4-bit I/O port Input or output direction can be specified in 1-bit units	$\overline{\text{BEN0}}$
PCD3			$\overline{\text{BEN1}}$
PCD4			$\overline{\text{BEN2}}$
PCD5			$\overline{\text{BEN3}}$
PCM0	I/O	Port CM 4-bit I/O port Input or output direction can be specified in 1-bit units	$\overline{\text{WAIT}}$
PCM1			
PCM6			
PCM7			

Table 2-1: Port Pins (5/5)

Pin Name	I/O	Function	Alternate Function
PCS0	I/O	Port CS 4-bit I/O port Input or output direction can be specified in 1-bit units	$\overline{CS0}$
PCS1			$\overline{CS1}$
PCS3			$\overline{CS3}$
PCS4			$\overline{CS4}$
PCT4	I/O	Port CT 2-bit I/O port Input or output direction can be specified in 1-bit units	$\overline{RD}$
PCT5			$\overline{WR}$

(2) Non-port pins

Table 2-2: Non-Port Pins (1/5)

Pin Name	I/O	Function	Alternate Function
A0 to A15	O	22-bit external address bus	PAL0 to PAL15
A16 to A21			PAH0 to PAH5
ADTRG0	I	A/D conversion start trigger (ADC0)	P03, INTP2
ADTRG1	I	A/D conversion start trigger (ADC1)	P04, INTP3
AFO	O	Auxiliary frequency output	P75, TECRT1
ANI00 to ANI09	I	Analog input channels (ADC0)	–
ANI10 to ANI19	I	Analog input channels (ADC1)	–
AV_{DD}	–	Positive power supply (3.3 V) (ADC0, ADC1)	–
AV_{REF0}	I	Reference voltage input (ADC0)	–
AV_{REF1}	I	Reference voltage input (ADC1)	–
AV_{SS0}	–	Power supply ground (ADC0)	–
AV_{SS1}	–	Power supply ground (ADC1)	–
$\overline{BEN0}$	O	External byte enable output	PCD2
$\overline{BEN1}$			PCD3
$\overline{BEN2}$			PCD4
$\overline{BEN3}$			PCD5
$\overline{CS0}$	O	Chip select signal output	PCS0
$\overline{CS1}$			PCS1
$\overline{CS3}$			PCS3
$\overline{CS4}$			PCS4
CV_{DD}	–	Oscillator power supply (1.5 V)	–
CV_{SS}	–	Oscillator power supply ground	–
D0 to D15	I/O	32-bit external data bus	PDL0 to PDL15
D16 to D31			PDH0 to PDH15
DCK	I	N-wire interface clock	–
DDI	I	N-wire data input and reset mode selection	–
DDO	O	N-wire data output	–
DMS	I	N-wire mode select	–
$\overline{DRST}$	I	N-wire interface reset	–
ESO0	I	Emergency shut off input (TMR0)	INTP0, P01
ESO1	I	Emergency shut off input (TMR1)	INTP1, P02
FCRXD0	I	Receive input (AFCAN0)	P34
FCRXD1	I	Receive input (AFCAN1)	P36
FCTXD0	O	Transmit output (AFCAN0)	P35
FCTXD1	O	Transmit output (AFCAN1)	P37
FLMD0	I	Flash programming mode selection	MODE0
FLMD1			MODE1

Table 2-2: Non-Port Pins (2/5)

Pin Name	I/O	Function	Alternate Function
INTP0	I	External maskable interrupt request input	P01, ESO0
INTP1			P02, ESO1
INTP2			P03, ADTRG0
INTP3			P04, ADTRG1
INTP4			P30, RXDC0
INTP5			P32, RXDC1
INTP6			P83, SCS300
INTP7			P84, SCS301
INTP8			P85, SCS302
INTP9			P93, SCS310
INTP10			P94, SCS311
INTP11			P95, SCS312
INTP12			P72, TECRT0
MODE0	I	Device operating mode selection	FLMD0
MODE1			FLMD1
MODE2			–
NMI	I	Non-maskable interrupt request input	P00
$\overline{RD}$	O	Read strobe signal output	PCT4
$\overline{RESET}$	I	System reset input	–
RXDC0	I	Receive input (UARTC0)	P30, INTP4
RXDC1	I	Receive input (UARTC1)	P32, INTP5
$\overline{SCK30}$	I/O	Serial shift clock I/O (CSI30)	P82
$\overline{SCK31}$	I/O	Serial shift clock I/O (CSI31)	P92
$\overline{SCKB0}$	I/O	Serial shift clock I/O (CSIB0)	P42
$\overline{SCKB1}$	I/O	Serial shift clock I/O (CSIB1)	P45
SCS300	O	Serial peripheral chip select (CSI30)	P83, INTP7
SCS301			P84, INTP8
SCS302			P85, INTP9
SCS303			P86, $\overline{SSB0}$
SCS310	O	Serial peripheral chip select (CSI31)	P93, INTP10
SCS311			P94, INTP10
SCS312			P95, INTP11
SCS313			P96, $\overline{SSB1}$
SI30	I	Serial data input (CSI30)	P80
SI31	I	Serial data input (CSI31)	P90
SIB0	I	Serial data input (CSIB0)	P40
SIB1	I	Serial data input (CSIB1)	P43
SO30	O	Serial data output (CSI30)	P81
SO31	O	Serial data output (CSI31)	P91
SOB0	O	Serial data output (CSIB0)	P41
SOB1	O	Serial data output (CSIB1)	P44

Table 2-2: Non-Port Pins (3/5)

Pin Name	I/O	Function	Alternate Function
$\overline{SSB0}$	I	Serial slave select input (CSIB0)	P86, SCS303
$\overline{SSB1}$	I	Serial slave select input (CSIB1)	P96, SCS313
TCLR1	I	Timer clear input (ITENC0)	P100, TICC10, TOP81
TCUD1	I	Count up/down direction control input (ITENC0)	P101, TICC11
TECRT0	I	Timer clear input (TMT0)	P72, INTP12
TECRT1	I	Timer clear input (TMT1)	P75, AFO
TENCT00	I	Timer encoder input (TMT0)	P70, TIT00, TEVTT1, TOT00
TENCT01	I		P71, TIT01, TTRGT1, TOT01
TENCT10	I	Timer encoder input (TMT1)	P73, TIT10, TTRGT0, TOT10
TENCT11	I		P74, TIT11, TEVTT0, TOT11
TEVTP0	I	Timer event input (TMP0)	P13, TIP11, TOP11
TEVTP1	I	Timer event input (TMP1)	P10, TIP00, TOP00
TEVTP2	I	Timer event input (TMP2)	P17, TIP31, TOP31
TEVTP3	I	Timer event input (TMP3)	P14, TIP20, TOP20
TEVTP4	I	Timer event input (TMP4)	P23, TIP51, TOP51
TEVTP5	I	Timer event input (TMP5)	P20, TIP40, TOP40
TEVTP6	I	Timer event input (TMP6)	P27, TIP71, TOP71
TEVTP7	I	Timer event input (TMP7)	P24, TIP60, TOP60
TEVTR1	I	Timer event input (TMR1)	P67, TOR17
TEVTT0	I	Timer event input (TMT0)	P74,TIT11, TOT11, TENCT11
TEVTT1	I	Timer event input (TMT1)	P70, TIT00, TOT00, TENCT00
TICC10	I	ITENC0 capture trigger input	P100, TCLR1, TOP81
TICC11			P101, TCUD1
TIP00	I	Capture trigger input (TMP0)	P10, TEVTP1, TOP00
TIP01			P11, TTRGP1, TOP01
TIP10	I	Capture trigger input (TMP1)	P12, TTRGP0, TOP10
TIP11			P13, TEVTP0, TOP11
TIP20	I	Capture trigger input (TMP2)	P14, TEVTP3, TOP20
TIP21			P15, TTRGP3, TOP21
TIP30	I	Capture trigger input (TMP3)	P16, TTRGP2, TOP30
TIP31			P17, TEVTP2, TOP31
TIP40	I	Capture trigger input (TMP4)	P20, TEVTP5, TOP40
TIP41			P21, TTRGP5, TOP41
TIP50	I	Capture trigger input (TMP5)	P22, TTRGP4, TOP50
TIP51			P23, TEVTP4, TOP51
TIP60	I	Capture trigger input (TMP6)	P24, TEVTP7, TOP60
TIP61			P25, TTRGP7, TOP61
TIP70	I	Capture trigger input (TMP7)	P26, TTRGP6, TOP70
TIP71			P27, TEVTP6, TOP71

Table 2-2: Non-Port Pins (4/5)

Pin Name	I/O	Function	Alternate Function
TIR10	I	Capture trigger input (TMR1)	P61, TOR11
TIR11			P62, TOR12
TIR12			P63, TOR13
TIR13			P64, TOR14
TIT00	I	Capture trigger input (TMT0)	P70, TEVTT1, TOT00, TENCT00
TIT01			P71, TTRGT1, TOT01, TENCT01
TIT10	I	Capture trigger input (TMT0)	P73, TTRGT0, TOT10, TENCT10
TIT11			P74,TEVTT0, TOT11, TENCT11
TIUD1	I	External count clock input (ITENC0)	P102, TO1
TO1	O	Pulse signal output (ITENC0)	P102, TIUD1
TOP00	O	Pulse signal output (TMP0)	P10, TIP00, TEVTP1
TOP01			P11, TIP01, TTRGP1
TOP10	O	Pulse signal output (TMP1)	P12, TIP10, TTRGP0
TOP11			P13, TIP11, TEVTP0
TOP20	O	Pulse signal output (TMP2)	P14, TIP20, TEVTP3
TOP21			P15, TIP21, TTRGP3
TOP30	O	Pulse signal output (TMP3)	P16, TIP30, TTRGP2
TOP31			P17, TIP31, TEVTP2
TOP40	O	Pulse signal output (TMP4)	P20, TIP40, TEVTP5
TOP41			P21, TIP41, TTRGP5
TOP50	O	Pulse signal output (TMP5)	P22, TIP50, TTRGP4
TOP51			P23, TIP51, TEVTP4
TOP60	O	Pulse signal output (TMP6)	P24, TIP60, TEVTP7
TOP61			P25, TIP61, TTRGP7
TOP70	O	Pulse signal output (TMP7)	P26, TIP70, TTRGP6
TOP71			P27, TIP71, TEVTP6
TOP81	O	Pulse signal output (TMP8)	P100, TCLR1, TICC10
TOR00	O	Pulse signal output (TMR0)	P50
TOR01			P51
TOR02			P52
TOR03			P53
TOR04			P54
TOR05			P55
TOR06			P56
TOR07			P57

Table 2-2: Non-Port Pins (5/5)

Pin Name	I/O	Function	Alternate Function
TOR10	O	Pulse signal output (TMR1)	P60, TTRGR1
TOR11			P61, TIR10
TOR12			P62, TIR11
TOR13			P63, TIR12
TOR14			P64, TIR13
TOR15			P65
TOR16			P66
TOR17			P67, TEVTR1
TOT00	O	Pulse signal output (TMT0)	P70, TIT00, TEVTT1, TENCT00
TOT01			P71, TIT01, TTRGT1, TENCT01
TOT10	O	Pulse signal output (TMT1)	P73, TIT10, TTRGT0, TENCT10
TOT11			P74,TIT11, TEVTT0, TENCT11
TTRGP0	I	Timer trigger input (TMP0)	P12, TIP10, TOP10
TTRGP1		Timer trigger input (TMP1)	P11, TIP01, TOP01
TTRGP2		Timer trigger input (TMP2)	P16, TIP30, TOP30
TTRGP3		Timer trigger input (TMP3)	P15, TIP21, TOP21
TTRGP4		Timer trigger input (TMP4)	P22, TIP50, TOP50
TTRGP5		Timer trigger input (TMP5)	P21, TIP41, TOP41
TTRGP6		Timer trigger input (TMP6)	P26, TIP70, TOP70
TTRGP7		Timer trigger input (TMP7)	P25, TIP61, TOP61
TTRGR1	I	Timer trigger input (TMR1)	P60, TOR10
TTRGT0	I	Timer trigger input (TMT0)	P73, TIT10, TOT10, TENCT10
TTRGT1	I	Timer trigger input (TMT1)	P71, TIT01, TOT01, TENCT01
TXDC0	O	Transmit output (UARTC0)	P31
TXDC1	O	Transmit output (UARTC1)	P33
V_{DD10} to V_{DD15}	–	Positive power supply for internal CPU (1.5 V)	–
V_{DD30} to V_{DD37}	–	Positive power supply for peripheral interface (3.3 V)	–
V_{SS10} to V_{SS15}	–	Power supply ground for internal CPU	–
V_{SS30} to V_{SS37}	–	Power supply ground for peripheral interface	–
$\overline{WAIT}$	I	External wait control signal input	PCM0
$\overline{WR}$	O	Write strobe signal output	PCT5
X1	I	Crystal connection	–
X2	–		–

2.2 Pin Status

Table 2-3: Pin Status in Reset and Standby Mode

Pin	Operating Status				
	During reset	After reset release			HALT Mode
		Single-chip Mode 0	Single-chip Mode 1	ROM-less Mode	
A0 to A15 (PAL0 to PAL15)	Hi-Z	Hi-Z	Operating		Operating
A16 to A21 (PAH0 to PAH5)	Hi-Z	Hi-Z	Operating		Operating
D0 to D15 (PDL0 to PDL15)	Hi-Z	Hi-Z	Operating		Operating
D16 to D31 (PDH0 to PDH15)	Hi-Z	Hi-Z	Operating		Operating
$\overline{BEN0}$ to $\overline{BEN3}$ (PCD2 to PCD5)	Hi-Z	Hi-Z	Operating		Operating
$\overline{CS0}$ (PCS0)	Hi-Z	Hi-Z	Operating		Operating
$\overline{CS1}$ (PCS1)	Hi-Z	Hi-Z	Operating		Operating
$\overline{CS3}$ (PCS3)	Hi-Z	Hi-Z	Operating		Operating
$\overline{CS4}$ (PCS4)	Hi-Z	Hi-Z	Operating		Operating
$\overline{RD}$ (PCT4)	Hi-Z	Hi-Z	Operating		Operating
$\overline{WR}$ (PCT5)	Hi-Z	Hi-Z	Operating		Operating
$\overline{WAIT}$ (PCM0)	Hi-Z	Hi-Z	Operating		Operating
BCLK (PCM1)	Hi-Z	Hi-Z	Hi-Z		Operating
STST (PCM6)	Hi-Z	Hi-Z	Hi-Z		Operating
STNXT (PCM7)	Hi-Z	Hi-Z	Hi-Z		Operating
DCK	Operating	Operating	Operating		Operating
DDI	Operating	Operating	Operating		Operating
DDO	Operating	Operating	Operating		Operating
DMS	Operating	Operating	Operating		Operating
$\overline{DRST}$	Operating	Operating	Operating		Operating
INTP0 to INTP3 (P01 to P04)	–	Input	Input		Operating
INTP4 (P30)	–	Input	Input		Operating
INTP5 (P32)	–	Input	Input		Operating
INTP6 to INTP8 (P83 to P85)	–	Input	Input		Operating
INTP9 to INTP11 (P93 to P95)	–	Input	Input		Operating
NMI (P00)	–	Input	Input		Operating
Peripheral input pin other than above	Hi-Z	Hi-Z	Hi-Z		Operating
Peripheral output pin other than above	×	×	×		Operating
Port input pin other than above	Hi-Z	Hi-Z	Hi-Z		–
Port output pin other than above	×	×	×		Hold

Remark: Hi-Z: High Impedance
–: Input data is not sampled
×: No function selected at reset

2.3 Description of Pin Functions

(1) P00 to P04 (Port 0) … Input

Port 0 is an 8-bit input-only port in which all pins are fixed for input.
Besides functioning as a port, in control mode, P00 to P04 operate as NMI input, external interrupt request signal, real-time pulse unit (RPU) emergency shut off signal input, and A/D converter (ADC) external trigger input. Normally, if function pins also serve as ports, one mode or the other is selected using a port mode control register. However, there is no such register for P00 to P04. Therefore, the input port cannot be switched with the NMI input pin, external interrupt request input pin, RPU emergency shut off signal input pin, and A/D converter (ADC) external trigger input pin. Read the status of each pin by reading the port.

(a) Port mode

P00 to P04 are input-only.

(b) Control mode

P00 to P04 also serve as NMI, INTP0 to INTP3, ESO0, ESO1, ADTRG0, and ADTRG1 pins, but the control function cannot be disabled.

(i) NMI (Non-maskable interrupt request) … Input
This is non-maskable interrupt request input.

(ii) INTP0 to INTP3 (Interrupt request from peripherals) … Input
These are external interrupt request input pins.

(iii) ESO0, ESO1 (Emergency shut off) … Input
These pins input timer TMR0 and timer TMR1 emergency shut off signals.

(iv) ADTRG0, ADTRG1 (A/D trigger input) … Input
These are A/D converter external trigger input pins.

(2) P10 to P17 (Port 1) … Input/Output

Port 1 is an 8-bit I/O port in which input or output can be set for each port pin individually. Besides functioning as an I/O port, in control mode, P10 to P17 operate as RPU input or output. The operation mode can be specified by the port 1 mode control register (PMC1) to port or control mode for each port pin individually.

(a) Port mode

P10 to P17 can be set to input or output in 1-bit units using the port 1 mode register (PM1).

(b) Control mode

P10 to P17 can be set to port or control mode in 1-bit units using the PMC1 register.

(i) TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31 (Timer capture input) … Input
These are timer TMP0 to TMP3 capture trigger input pins.

(ii) TEVTP0, TEVTP1, TEVTP2, TEVTP3 (Timer event input) … Input
These are timer TMP0 to TMP3 external event counter input pins.

(iii) TTRGP0, TTRGP1, TTRGP2, TTRGP3 (Timer trigger) … Input
These are timer TMP0 to TMP3 external trigger input pins.

(iv) TOP00, TOP01, TOP10, TOP11, TOP20, TOP21, TOP30, TOP31 (Timer output) … Output
These pins output timer TMP0 to TMP3 pulse signals.

(3) P20 to P27 (Port 2) … Input/Output

Port 2 is an 8-bit I/O port in which input or output can be set for each port pin individually. Besides functioning as an I/O port, in control mode, P20 to P27 operate as RPU input or output. The operation mode can be specified by the port 2 mode control register (PMC2) to port or control mode for each port pin individually.

(a) Port mode

P20 to P27 can be set to input or output in 1-bit units using the port 2 mode register (PM2).

(b) Control mode

P20 to P27 can be set to port or control mode in 1-bit units using the PMC2 register.

(i) TIP40, TIP41, TIP50, TIP51, TIP60, TIP61, TIP70, TIP71 (Timer capture input) … Input
These are timer TMP4 to TMP7 capture trigger input pins.

(ii) TEVTP4, TEVTP5, TEVTP6, TEVTP7 (Timer event input) … Input
These are timer TMP4 to TMP7 external event counter input pins.

(iii) TTRGP4, TTRGP5, TTRGP6, TTRGP7 (Timer trigger) … Input
These are timer TMP4 to TMP7 external trigger input pins.

(iv) TOP40, TOP41, TOP50, TOP51, TOP60, TOP61, TOP70, TOIP71 (Timer output) … Output
These pins output timer TMP4 to TMP7 pulse signals.

(4) P30 to P37 (Port 3) … Input/Output

Port 3 is an 8-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as an I/O port, in control mode, P30 to P37 operate as serial interface (UARTC0, UARTC1, AFCAN0, AFCAN1). Additionally external interrupt request signal inputs are available in port input mode.
The operation mode can be specified by the port 3 mode control register (PMC3) to port or control mode for each port pin individually.

(a) Port mode

P30 to P37 can be set to input or output in 1-bit units using the port 3 mode register (PM3).

(i) INTP4, INTP5 (Interrupt request from peripherals) … Input
These are external interrupt request input pins, which are simultaneously enabled in port input mode.

(b) Control mode

P30 to P37 can be set to port or control mode in 1-bit units using the PMC3 register.

(i) TXDC0, TXDC1 (Transmit data) … Output
These pins output serial transmit data of UARTC0 and UARTC1.

(ii) RXDC0, RXDC1 (Receive data) … Input
These pins input serial receive data of UARTC0 and UARTC1.

(iii) FCTXD0, FCTXD1 (Transmit data for controller area network) … Output
These pins output AFCAN0 and AFCAN1 serial transmit data.

(iv) FCRXD 0, FCRXD1 (Receive data for controller area network) … Input
These pins input AFCAN0 and AFCAN1 serial receive data.

(5) P40 to P45 (Port 10) … Input/Output

Port 4 is a 6-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as an I/O port, in control mode, P40 to P45 operate as serial interface (CSIB0, CSIB1).
The operation mode can be specified by the port 4 mode control register (PMC4) to port or control mode for each port pin individually.

(a) Port mode

P40 to P45 can be set to input or output in 1-bit units using the port 4 mode register (PM4).

(b) Control mode

P40 to P45 can be set to port or control mode in 1-bit units using the PMC4 register.

(i) SOB0, SOB1 (Serial output) … Output
These pins output CSIB0 and CSIB1 serial transmit data.

(ii) SIB0, SIB1 (Serial input) … Input
These pins input CSIB0 and CSIB1 serial receive data.

(iii) $\overline{\text{SCKB0}}$, $\overline{\text{SCKB1}}$ (Serial clock) … I/O
These are the CSIB0 and CSIB1 serial clock I/O pins.

(6) P50 to P57 (Port 5) … Input/Output

Port 5 is an 8-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as an I/O port, in control mode, P50 to P57 operate as RPU input or output.
The operation mode can be specified by the port 5 mode control register (PMC5) to port or control mode for each port pin individually.

(a) Port mode

P50 to P57 can be set to input or output in 1-bit units using the port 5 mode register (PM5).

(b) Control mode

P50 to P57 can be set to port or control mode in 1-bit units using the PMC5 register.

(i) TOR00, TOR01, TOR02, TOR03, TOR04 (Timer output) … Output
These pins output timer TMR0 pulse signals.

(7) P60 to P67 (Port 6) … Input/Output

Port 6 is an 8-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as an I/O port, in control mode, P60 to P67 operate as RPU input or output.
The operation mode can be specified by the port 6 mode control register (PMC6) to port or control mode for each port pin individually.

(a) Port mode

P60 to P67 can be set to input or output in 1-bit units using the port 6 mode register (PM6).

(b) Control mode

P60 to P67 can be set to port or control mode in 1-bit units using the PMC6 register.

(i) TIR10, TIR11, TIR12, TIR13 (Timer capture input) … Input
These are timer TMR1 capture trigger input pins.

(ii) TEVTR1 (Timer event input) … Input
This is a timer TMR1 external event counter input pin.

(iii) TTRGR1 (Timer trigger) … Input
This is a timer TMR1 external trigger input pin.

(iv) TOR10, TOR11, TOR12, TOR13, TOR14 (Timer output) … Output
These pins output timer TMR1 pulse signals.

(8) P70 to P75 (Port 7) … Input/Output

Port 7 is a 6-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as an I/O port, in control mode, P70 to P75 operate as RPU input or output, and auxiliary frequency output. Additionally an external interrupt request signal input is available in port input mode.
The operation mode can be specified by the port 7 mode control register (PMC7) to port or control mode for each port pin individually.

(a) Port mode

P70 to P75 can be set to input or output in 1-bit units using the port 7 mode register (PM7).

(i) INTP12 (Interrupt request from peripherals) … Input
This is an external interrupt request input pin, which is simultaneously enabled in port input mode.

(b) Control mode

P70 to P75 can be set to port or control mode in 1-bit units using the PMC7 register.

(i) TIT00, TIT01, TIT10, TIT11 (Timer capture input) … Input
These are timer TMT0 and TMT1 capture trigger input pins.

(ii) TEVTT0, TEVTT1 (Timer event input) … Input
These are timer TMT0 and TMT1 external event counter input pins.

(iii) TTRGT0, TTRGT1 (Timer trigger) … Input
These are timer TMT0 and TMT1 external trigger input pins.

(iv) TECRT0, TECRT1 (Timer clear) … Input
These are timer TMT0 and TMT1 external clear input pins.

(v) TENCT00, TENCT01, TENCT10, TENCT11 (Timer encoder input … Input
These are timer TMT0 and TMT1 encoder input pins.

(vi) TOT00, TOT01, TOT10, TOT11 (Timer output) … Output
These pins output timer TMT0 and TMT1 pulse signals.

(vii)AFO (Auxiliary frequency) … Output
This is an auxiliary frequency output signal pin of baudrate generator BGR2.

(9) P80 to P86 (Port 8) … Input/Output

Port 8 is a 7-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as an I/O port, in control mode, P80 to P86 operate as serial interface (CSI30, CSIB0). Additionally external interrupt request signal inputs are available in port input mode.
The operation mode can be specified by the port 8 mode control register (PMC8) to port or control mode for each port pin individually.

(a) Port mode

P80 to P86 can be set to input or output in 1-bit units using the port 8 mode register (PM8).

(i) INTP6, INTP7, INTP8 (Interrupt request from peripherals) … Input
These are external interrupt request input pins, which are simultaneously enabled in port input mode.

(b) Control mode

P80 to P86 can be set to port or control mode in 1-bit units using the PMC8 register.

(i) SO30 (Serial output) … Output
This pin outputs CSI30 serial transmit data.

(ii) SI30 (Serial input) … Input
This pin inputs CSI30 serial receive data.

(iii) $\overline{\text{SCK30}}$ (Serial clock) … I/O
This is the CSI30 serial clock I/O pin.

(iv) SCS300 to SCS303 (Serial chip select) … Output
These pins output CSI30 serial chip select signals.

(v) $\overline{\text{SSB0}}$ (Serial slave select signal) … Input
This pin inputs CSIB0 slave select signal.

(10) P90 to P96 (Port 9) … Input/Output

Port 9 is a 7-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as an I/O port, in control mode, P90 to P96 operate as serial interface (CSI31, CSIB1). Additionally external interrupt request signal inputs are available in port input mode.
The operation mode can be specified by the port 9 mode control register (PMC9) to port or control mode for each port pin individually.

(a) Port mode

P90 to P96 can be set to input or output in 1-bit units using the port 9 mode register (PM9).

(i) INTP9, INTP10, INTP11 (Interrupt request from peripherals) … Input
These are external interrupt request input pins, which are simultaneously enabled in port input mode.

(b) Control mode

P90 to P96 can be set to port or control mode in 1-bit units using the PMC9 register.

(i) SO31 (Serial output) … Output
This pin outputs CSI31 serial transmit data.

(ii) SI31 (Serial input) … Input
This pin inputs CSI31 serial receive data.

(iii) $\overline{\text{SCK31}}$ (Serial clock) … I/O
This is the CSI31 serial clock I/O pin.

(iv) SCS310 to SCS313 (Serial chip select) … Output
These pins output CSI31 serial chip select signals.

(v) $\overline{\text{SSB1}}$ (Serial slave select input) … Input
This pin inputs CSIB1 slave select signal.

(11) P100 to P102 (Port 10) … Input/Output

Port 10 is a 3-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as an I/O port, in control mode, P100 to P102 operate as RPU input or output
The operation mode can be specified by the port 10 mode control register (PMC10) to port or control mode for each port pin individually.

(a) Port mode

P100 to P102 can be set to input or output in 1-bit units using the port 10 mode register (PM10).

(b) Control mode

P100 to P102 can be set to port or control mode in 1-bit units using the PMC4 register.

(i) TIUD1 (Timer count pulse input) … Input
This is an external count clock input pin to the up/down counter (TMENC10).

(ii) TCUD1 (Timer control pulse input) … Input
This is an input count operation switching signal to the up/down counter (TMENC10).

(iii) TCLR1 (Timer clear) … Input
This is a clear signal input pin to the up/down counter (TMENC10).

(iv) TICC10, TICC11 (Timer capture input) … Input
These are timer TMENC10 external capture trigger input pins.

(v) TO1 (Timer output) … Output
This pin outputs timer TMENC10 pulse signals.

(vi) TOP80 (Timer output) … Output
This pin outputs timer TMP8 pulse signals.

(12) PAL0 to PAL15 (Port AL) … I/O

Port AL is an 8-bit or a 16-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as a port, in control mode, these pins operate as the address bus (A0 to A15) when memory is expanded externally.
The operation mode can be specified by the port AL mode control register (PMCAL) to port or control mode for each port pin individually.

(a) Port mode

PAL0 to PAL15 can be set to input or output in 1-bit units using the port AL mode register (PMAL).

(b) Control mode

PAL0 to PAL15 can be set to port or control mode in 1-bit units using the PMCAL register.

(i) A0 to A15 (Address bus) … 3-state output
These are the address output pins of the lower 16 bits of the 22-bit address bus when the external memory is accessed.

(13) PAH0 to PAH5 (Port AH) … I/O

Port AH is a 6-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as a port, in control mode, these pins operate as the address bus (A16 to A21) when memory is expanded externally.
The operation mode can be specified by the port AH mode control register (PMCAH) to port or control mode for each port pin individually.

(a) Port mode

PAH0 to PAH5 can be set to input or output in 1-bit units using the port AH mode register (PMAH).

(b) Control mode

PAH0 to PAH6 can be set to port or control mode in 1-bit units using the PMCAH register.

(i) A16 to A21 (Address bus) … 3-state output
These are the address output pins of the higher 6 bits of the 22-bit address bus when the external memory is accessed.

(14) PDL0 to PDL15 (Port DL) … I/O

Port DL is an 8-bit or a 16-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as a port, in control mode, these pins operate as the data bus (D0 to D15) when memory is expanded externally.
The operation mode can be specified by the port DL mode control register (PMCDL) to port or control mode for each port pin individually.

(a) Port mode

PDL0 to PDL15 can be set to input or output in 1-bit units using the port DL mode register (PMDL).

(b) Control mode

PDL0 to PDL15 can be set to port or control mode in 1-bit units using the PMCDL register.

(i) D0 to D15 (Address bus) … 3-state I/O
These are the data I/O pins of the lower 16 bits of the 32-bit data bus when the external memory is accessed.

(15) PDH0 to PDH15 (Port DH) … I/O

Port DH is an 8-bit or a 16-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as a port, in control mode, these pins operate as the data bus (D16 to D31) when memory is expanded externally.
The operation mode can be specified by the port DH mode control register (PMCDH) to port or control mode for each port pin individually.

(a) Port mode

PDH0 to PDH15 can be set to input or output in 1-bit units using the port DH mode register (PMDH).

(b) Control mode

PDH0 to PDH15 can be set to port or control mode in 1-bit units using the PMCDH register.

(i) D16 to D31 (Address bus) … 3-state I/O
These are the data I/O pins of the higher 16 bits of the 32-bit data bus when the external memory is accessed.

(16) PCD2 to PCD5 (Port CD) … I/O

Port CD is a 4-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as a port, in control mode, these pins operate as control signal outputs when memory is expanded externally.
The operation mode can be specified by the port CD mode control register (PMCCD) to port or control mode for each port pin individually.

(a) Port mode

PCD2 to PCD5 can be set to input or output in 1-bit units using the port CD mode register (PMCD).

(b) Control mode

PCD2 to PCD5 can be set to port or control mode in 1-bit units using the PMCCD register.

(i) $\overline{\text{BEN0}}$ to $\overline{\text{BEN3}}$ (Byte enable) … 3-state output
These are the byte enable control signal pins, which indicate the validity of the corresponding byte on the 32-bit data bus.

(17) PCM0, PCM1, PCM6, PCM7 (Port CM) … I/O

Port CM is a 4-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as a port, in control mode, these pins operate as control signal input when memory is expanded externally.
The operation mode can be specified by the port CM mode control register (PMCCM) to port or control mode for each port pin individually.

(a) Port mode

PCM0, PCM1, PCM6, and PCM7 can be set to input or output in 1-bit units using the port CM mode register (PMCM).

(b) Control mode

PCM0 can be set to port or control mode in 1-bit units using the PMCCM register.

(i) $\overline{\text{WAIT}}$ (Wait) … Input
This is the control signal input pin at which an external data wait is inserted into the bus cycle.
The $\overline{\text{WAIT}}$ signal can be input asynchronously, and is sampled at the falling edge of the $\overline{\text{BCLK}}$ signal. When the setup or hold time is terminated within the sampling timing, wait insertion may not be executed.

(18) PCS0, PCS1, PCS3, PCS4 (Port CS) … I/O

Port CS is a 4-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as a port, in control mode, these pins operate as control signal outputs when memory is expanded externally.
The operation mode can be specified by the port CS mode control register (PMCCS) to port or control mode for each port pin individually.

(a) Port mode

PCS0, PCS1, PCS3, and PCS4 can be set to input or output in 1-bit units using the port CS mode register (PMCS).

(b) Control mode

PCS0, PCS1, PCS3, and PCS4 can be set to port or control mode in 1-bit units using the PMCCS register.

(i) $\overline{\text{CS0}}$, $\overline{\text{CS1}}$, $\overline{\text{CS3}}$, $\overline{\text{CS4}}$ (Chip select) … 3-state output
These are the chip select signal output pins for the external memory or peripheral I/O extension areas.
The $\overline{\text{CSn}}$ signal is assigned to the memory block n (n = 0, 1, 3, 4).
It becomes active while the bus cycle that accesses the corresponding memory block is activated. In the idle state (TI), it becomes inactive.

(19) PCT4, PCT5 (Port CT) … I/O

Port CT is a 2-bit I/O port in which input or output can be set for each port pin individually.
Besides functioning as a port, in control mode, these pins operate as control signal outputs when memory is expanded externally.
The operation mode can be specified by the port CT mode control register (PMCCT) to port or control mode for each port pin individually.

(a) Port mode

PCT4 and PCT5 can be set to input or output in 1-bit units using the port CT mode register (PMCT).

(b) Control mode

PCT4 and PCT5 can be set to port or control mode in 1-bit units using the PMCCT register.

(i) $\overline{\text{RD}}$ (Read strobe) … 3-state output
This is a strobe signal output pin that shows whether the bus cycle currently being executed is a read cycle for the external memory or peripheral I/O extension area. In the idle state (TI), it becomes inactive.

(ii) $\overline{\text{WR}}$ (Write strobe) … 3-state output
This is a strobe signal output pin that shows whether the bus cycle currently being executed is a write cycle for the external memory or peripheral I/O extension area.

(20) DCK (Debug clock) … Input

This pin inputs a debug clock. At the rising edge of the DCK signal, the DMS and DDI signals are sampled, and data is output from the DDO pin at the falling edge of the DCK signal. Keep this pin high when the debug function is not used.

(21) DDI (Debug data input) … Input

This pin inputs debug data, which is sampled at the rising edge of the DCK signal when the debug serial interface is in the shift state. Data is input with the LSB first. Keep this pin high when the debug function is not used.

(22) DDO (Debug data output) … Output

This pin outputs debug data at the falling edge of the DCK signal when the debug serial interface is in the shift state. Data is output with the LSB first.

(23) DMS (Debug mode select) … Input

This input pin selects a debug mode. Depending on the level of the DMS signal, the state machine of the debug serial interface changes. This pin is sampled at the rising edge of the DCK signal. Keep this pin high when the debug function is not used.

(24) $\overline{\text{DRST}}$ (Debug reset) … Input

This pin inputs a debug reset signal that is a negative-logic signal to initialize the DCU asynchronously.
When this signal goes low, the DCU is reset/invalidated. Keep this pin low when the debug function is not used.

(25) MODE0 to MODE2 (Mode) … Input

These are input pins used to specify the operating mode.

(26) FLMD0, FLMD1 (flash programming mode)

These are input pins used to specify the flash programming mode.

(27) $\overline{\text{RESET}}$ (Reset) … Input

$\overline{\text{RESET}}$ is a signal that is input asynchronously and that has a constant low level width regardless of the operating clock's status. When this signal is input, a system reset is executed as the first priority ahead of all other operations.
In addition to being used for ordinary initialization/start operations, this pin can also be used to release a standby mode (HALT).

(28) X1, X2 (Crystal)

These pins are used to connect the resonator that generates the system clock.

(29) ANI00 to ANI09, ANI10 to ANI19 (Analog input) … Input

These are analog input pins of the corresponding A/D converter (ADC0, ADC1).

(30) AV_{REF0}, AV_{REF1} (Analog reference voltage) … Input

These are reference voltage supply pins for the corresponding A/D converter (ADC0, ADC1).

(31) AV_{DD} (Analog power supply)

This is the positive power supply pin for the A/D converters.

(32) AV_{SS} (Analog ground)

This is the analog ground pin for the A/D converters.

(33) CV_{DD} (Power supply for clock generator)

This is the positive power supply pin for the clock generator.

(34) CV_{SS} (Ground for clock oscillator)

This is the ground pin for the clock generator.

(35) V_{DD10} to V_{DD15} (Power supply)

These are the positive power supply pins for the internal CPU.

(36) V_{DD30} to V_{DD37} (Power supply)

These are the positive power supply pins for the peripheral interface.

(37) V_{SS10} to V_{SS15} (Ground)

These are the ground pins for the internal CPU.

(38) V_{SS30} to V_{SS37} (Ground)

These are the ground pins for the peripheral interface.

2.4 Pin I/O Circuits and Recommended Connection of Unused Pins

Table 2-4: I/O Circuit Types (1/3)

Terminal	I/O circuit type	Recommended termination
P00/NMI	2	Connect independently to V_{SS3} via a resistor
P01/INTP0/ESO0		
P02/INTP1/ESO1		
P03INTP2/ADTRG0		
P04INTP3/ADTRG1		
P10/TIP00/TEVTP1/TOP00	5-K	Input: Connect independently to V_{DD3} or V_{SS3} via a resistor Output: leave open
P11/TIP01/TTRGP1/TOP01		
P12/TIP10/TTRGP0/TOP10		
P13/TIP11/TEVTP0/TOP11		
P14/TIP20/TEVTP3/TOP20		
P15/TIP21/TTRGP3/TOP21		
P16/TIP30/TTRGP2/TOP30		
P17/TIP31/TEVTP2/TOP31		
P20/TIP40/TEVTP5/TOP40		
P21/TIP41/TTRGP5/TOP41		
P22/TIP50/TTRGP4/TOP50		
P23/TIP51/TEVTP4/TOP51		
P24/TIP60/TEVTP7/TOP60		
P25/TIP61/TTRGP7/TOP61		
P26/TIP70/TTRGP6/TOP70		
P27/TIP71/TEVTP6/TOP71		
P30/RXDC0/INTP4		
P31/TXDC0		
P32/RXDC1/INTP5		
P33/TXDC1		
P34/FCRXD0		
P35/FCTXD0		
P36/FCRXD1		
P37/FCTXD1		
P40/SIB0		
P41/SOB0		
P42/$\overline{\text{SCKB0}}$		
P43/SIB1		
P44/SOB1		
P45/$\overline{\text{SCKB1}}$		
P50/TOR00 to P57/TOR07		

Table 2-4: I/O Circuit Types (2/3)

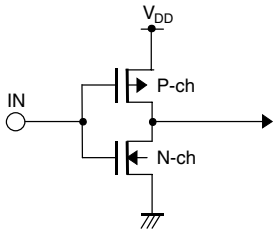
Terminal	I/O circuit type	Recommended termination
P60/TOR10/TTRGR1	5-K	Input: Connect independently to V_{DD3} or V_{SS3} via a resistor Output: leave open
P61/TOR11/TIR10		
P62/TOR12/TIR11		
P63/TOR13/TIR12		
P64/TOR14/TIR13		
P65/TOR15		
P66/TOR16		
P67/TOR17/TEVTR1		
P70/TIT00/TEVTT1/TOT00/TENCT00		
P71/TIT01/TTRGT1/TOT01/TENCT01		
P72/TECRT0/INTP12		
P73/TIT10/TTRGT0/TOT10/TENCT10		
P74/TIT11/TEVTT0/TOT11/TENCT11		
P75/TECRT1/AFO		
P80/SI30		
P81/SO30		
P82/$\overline{\text{SCK30}}$		
P83/SCS300/INTP6		
P84/SCS301/INTP7		
P85/SCS302/INTP8		
P86/SCS303/$\overline{\text{SSB0}}$		
P90/SI31		
P91/SO31		
P92/$\overline{\text{SCK31}}$		
P93/SCS310/INTP9		
P94/SCS311/INTP10		
P95/SCS312/INTP11		
P96/SCS313/$\overline{\text{SSB1}}$		
P100/TCLR1/TICC10/TOP80		
P101/TCUD1/TICC11		
P102/TIUD1/TO1		
PAH0/A16 to PAH5/A21	5	Input: Connect independently to V_{DD3} or V_{SS3} via a resistor Output: leave open
PAL0/A0 to PAL15/A15		
PDH0/D16 to PDH15/D31		
PDL0/D0 to PDL15/D15		
PCS0/$\overline{\text{CS0}}$		
PCS1/$\overline{\text{CS1}}$		
PCS3/$\overline{\text{CS3}}$		
PCS4/$\overline{\text{CS4}}$		
PCD2/$\overline{\text{BEN0}}$ to PCD5/$\overline{\text{BEN3}}$		

Table 2-4: I/O Circuit Types (3/3)

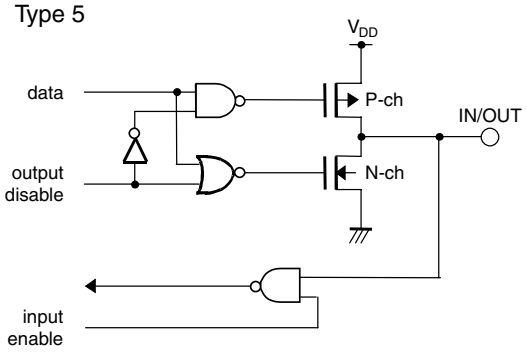
Terminal	I/O circuit type	Recommended termination
PCT4/$\overline{RD}$	5	Input: Connect independently to V_{DD3} or V_{SS3} via a resistor Output: leave open
PCT5/$\overline{WR}$		
PCM0/$\overline{WAIT}$		
PCM1		
PCM6		
PCM7		
$\overline{RESET}$	2	Pin must be used in the intended way
X1	–	
X2	–	
MODE0/FLMD0	2	
MODE1/FLMD1	2	
MODE2	2	
DCK	1	Connect independently to V_{DD3} via a resistor
$\overline{DRST}$	2-I	Leave open (on-chip pull-down resistor
DMS	1	Connect independently to V_{DD3} via a resistor
DDI		
DDO	3	Leave open (always level output during reset)
ANI00 to ANI09	7	Connect independently to AV_{DD} or AV_{SS} via a resistor
ANI10 to ANI19		
AV_{REF0}	–	Connect independently to AV_{SS} via a resistor
AV_{REF1}		
AV_{DD}	–	Pin must be used in the intended way
AV_{SS0}		
AV_{SS1}		
V_{DD10} to V_{DD15}		
V_{SS10} to V_{SS15}		
V_{DD30} to V_{DD37}		
V_{SS30} to V_{SS37}		
CV_{DD}		
CV_{SS}		

Figure 2-1: Pin I/O Circuits

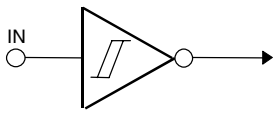
Type 1



Type 5

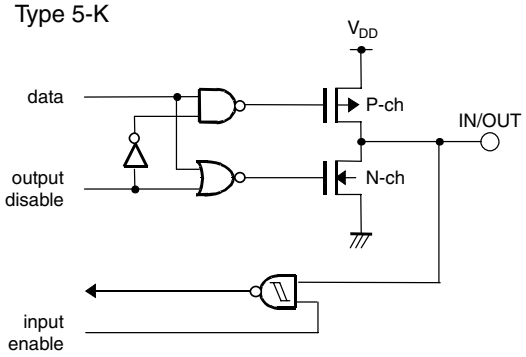


Type 2

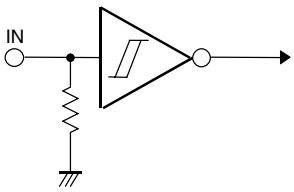


Schmitt trigger input with hysteresis characteristics

Type 5-K

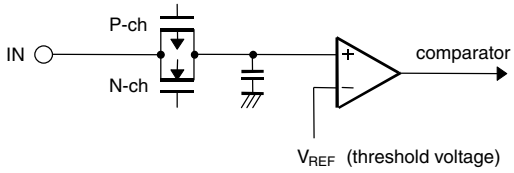


Type 2-I

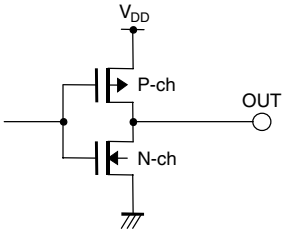


Schmitt trigger input with hysteresis characteristics

Type 7



Type 3



2.5 Noise Suppression

The V850E/PH2 has a digital or analog delay circuits for noise suppression on all edge sensitive inputs. The digital delay circuit suppresses input pulses shorter than the internally generated edge detection signal to assure the hold time for these signals. The noise suppression is only effective on alternate pin functions, and it is not effective when the port input function is selected.

Table 2-5: Noise Suppression Timing

Pin Function	Noise removal time	Clock Source
NMI	4 to 5 clocks	$f_{XX}/16$ or $f_{XX}/64$ (set by NCR0 bit of NRC register)
INTP0, INTP1, ESO0, ESO1	Analog delay (60ns to 200ns)	
INTP2 to INTP11, ADTRG0, ADTRG1	4 to 5 clocks	$f_{XX}/16$ or $f_{XX}/64$ (set by NCR1 bit of NRC register)
INTP12, TICC00, TICC01, TCLR0, TCUD0, TIUD0, TIT00, TIT01, TIT10, TIT11, TECRT0, TECRT1, TEVTT0, TEVTT1, TTRGT0, TTRGT1, TENCT00, TENCT01, TENCT10, TENCT11	4 to 5 clocks	$f_{XX}/16$ or $f_{XX}/64$ (set by NCR2 bit of NRC register)
TIP00, TIP01, TIP10, TIP11, TEVTP0, TEVTP1, TTRGP0, TTRGP1	4 to 5 clocks	$f_{XX}/16$ or $f_{XX}/64$ (set by NCR3 bit of NRC register)
TIP20, TIP21, TIP30, TIP31, TEVTP2, TEVTP3, TTRGP2, TTRGP3	4 to 5 clocks	$f_{XX}/16$ or $f_{XX}/64$ (set by NCR4 bit of NRC register)
TIP40, TIP41, TIP50, TIP51, TEVTP4, TEVTP5, TTRGP4, TTRGP5	4 to 5 clocks	$f_{XX}/16$ or $f_{XX}/64$ (set by NCR5 bit of NRC register)
TIP60, TIP61, TIP70, TIP71, TEVTP6, TEVTP7, TTRGP6, TTRGP7	4 to 5 clocks	$f_{XX}/16$ or $f_{XX}/64$ (set by NCR6 bit of NRC register)
TIR10 to TIR13, TEVTR1, TTRGR1	4 to 5 clocks	$f_{XX}/16$ or $f_{XX}/64$ (set by NCR7 bit of NRC register)

(1) Noise removal time control register (NRC)

The NRC register specifies the noise removal clock setting for different edge sensitive inputs.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Figure 2-2: Noise Removal Time Control Register (1/2)

After reset: 00H R/W Address: FFFFF7A0H

	7	6	5	4	3	2	1	0
NRC	NCR7	NCR6	NCR5	NCR4	NCR3	NCR2	NCR1	NCR0

NCR7	Noise removal clock setting for input pins TIR10 to TIR13, TEVTR1, TTRGR1
0	$f_{XX}/16$
1	$f_{XX}/64$

NCR6	Noise removal clock setting for input pins TIP60, TIP61, TIP70, TIP71, TEVTP6, TEVTP7, TTRGP6, TTRGP7
0	$f_{XX}/16$
1	$f_{XX}/64$

NCR5	Noise removal clock setting for input pins TIP40, TIP41, TIP50, TIP51, TEVTP4, TEVTP5, TTRGP4, TTRGP5
0	$f_{XX}/16$
1	$f_{XX}/64$

NCR4	Noise removal clock setting for input pins TIP20, TIP21, TIP30, TIP31, TEVTP2, TEVTP3, TTRGP2, TTRGP3
0	$f_{XX}/16$
1	$f_{XX}/64$

NCR3	Noise removal clock setting for input pins TIP00, TIP01, TIP10, TIP11, TEVTP0, TEVTP1, TTRGP0, TTRGP1
0	$f_{XX}/16$
1	$f_{XX}/64$

NCR2	Noise removal clock setting for input pins INTP12, TICC00, TICC01, TCLR0, TCUD0, TIUD0, TIT00, TIT01, TIT10, TIT11, TECRT0, TECRT1, TEVTT0, TEVTT1, TTRGT0, TTRGT1, TENCT00, TENCT01, TENCT10, TENCT11
0	$f_{XX}/16$
1	$f_{XX}/64$

Figure 2-2: Noise Removal Time Control Register (2/2)

NCR1	Noise removal clock setting for input pins INTP2 to INTP11, ADTRG0, ADTRG1
0	$f_{XX}/16$
1	$f_{XX}/64$

NCR0	Noise removal clock setting for NMI input pin
0	$f_{XX}/16$
1	$f_{XX}/64$

Chapter 3 CPU Functions

The CPU of the V850E/PH2 microcontroller is based on the RISC architecture and executes most instructions in one clock cycle by using a 5-stage pipeline control.

3.1 Features

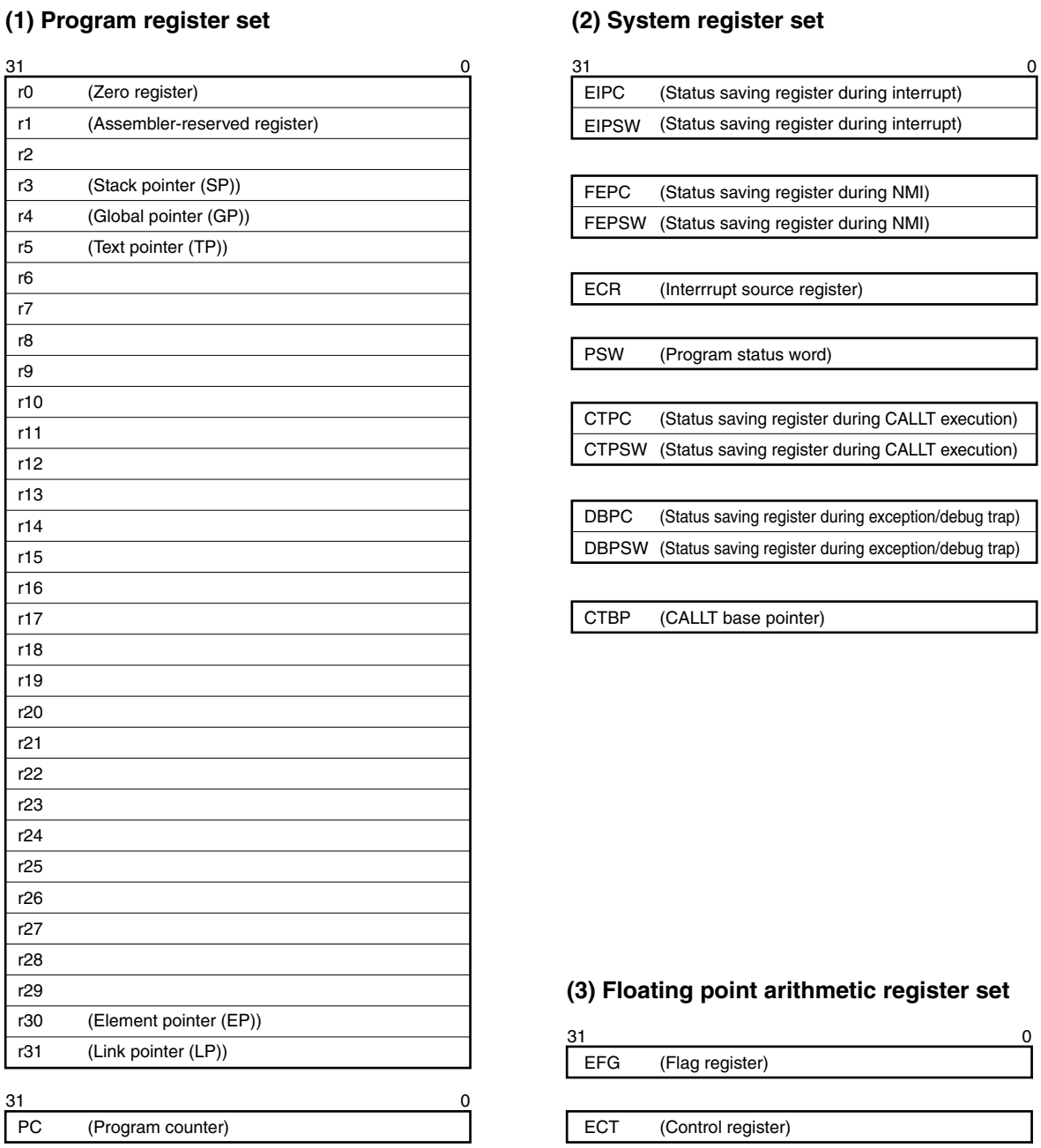
- Number of instructions: 96
- Minimum instruction execution time: 15.6 ns (@ 64 MHz operation)
- Memory space
  Program space: 64 MB linear
  Data space: 4 GB linear
- General-purpose registers: 32 bits × 32
- Internal 32-bit architecture
- 5-stage pipeline control
- Multiply/divide instructions (32 bits × 32 bits →64 bits in 1 to 2 clocks)
- Saturated operation instructions
- Floating point arithmetic unit (single precision, 32 bits, IEEE754-85 standard)
- 32-bit shift instruction: 1 clock
- Load/store instruction with long/short format
- Four types of bit manipulation instructions
  - SET1
  - CLR1
  - NOT1
  - TST1

3.2 CPU Register Set

The CPU registers of the V850E/PH2 can be classified into three categories: a general-purpose program register set, a dedicated system register set and a dedicated floating point arithmetic register set. All the registers have 32-bit width.
In addition, the V850E/PH2 contains special system control registers that should be initialized before CPU operation, and a specific register controlling its clock.

For detailed description of V850E1 core, refer to V850E1 Core Architecture Manual and the addendum for floating point arithmetic.

Figure 3-1: CPU Register Set



3.2.1 Program register set

The program register set includes general-purpose registers and a program counter.

(1) General-purpose registers (r0 to r31)

Thirty-two general-purpose registers, r0 to r31, are available. All of these registers can be used as a data variable or address variable.

However, r0 and r30 are implicitly used by instructions and care must be exercised when using these registers. r0 always holds 0 and is used for operations that use 0 or offset 0 addressing. r30 is used as a base pointer when performing memory access with the SLD and SST short instructions.

Also, r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. Therefore, before using these registers, their contents must be saved so that they are not lost, and they must be restored to the registers after use. There are cases when r2 is used by the real-time OS. If r2 is not used by the real-time OS, r2 can be used as a variable register.

Table 3-1: Program Registers

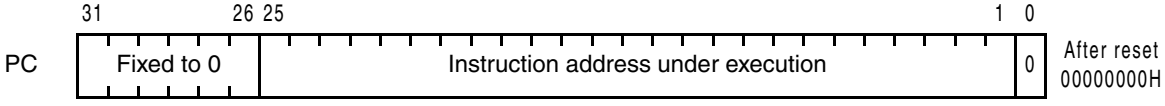
Name	Usage	Operation
r0	Zero register	Always holds 0
r1	Assembler-reserved register	Working register for generating 32-bit immediate
r2	Address/data variable register (when r2 is not used by the real-time OS to be used)	
r3	Stack pointer	Used to generate stack frame when function is called
r4	Global pointer	Used to access global variable in data area
r5	Text pointer	Register to indicate the start of the text area (area for placing program code)
r6 to r29	Address/data variable register	
r30	Element pointer	Base pointer when memory is accessed
r31	Link pointer	Used by compiler when calling function

(2) Program counter (PC)

This register holds the address of the instruction under execution. The lower 26 bits of this register are valid, and bits 31 to 26 are fixed to 0. If a carry occurs from bit 25 to bit 26, it is ignored.

Bit 0 is fixed to 0, and branching to an odd address cannot be performed.

Figure 3-2: Program Counter (PC)



3.2.2 System register set

System registers control the status of the CPU and hold interrupt information.
Read from and write to system registers are performed by setting the system register numbers shown below with the system register load/store instructions (LDSR, STSR instructions).

Table 3-2: System Register Numbers

System Register			Operand Specification Enabled for instruction	
No.	Name	Function	LDSR	STSR
0	EIPC	PC value at Interrupt handler entry **Note 1**	Yes	Yes
1	EIPSW	PSW value at Interrupt handler entry **Note 1**	Yes	Yes
2	FEPC	PC value at NMI handler entry	Yes	Yes
3	FEPSW	PSW value at NMI handler entry	Yes	Yes
4	ECR	Exception Cause Register	No	Yes
5	PSW	Program status word	Yes	Yes
6 to 15	-	Reserved numbers for future function expansion (The operation is not guaranteed if accessed.)	No	No
16	CTPC	PC value at CALLT subroutine entry **Note 2**	Yes	Yes
17	CTPSW	PSW value at CALLT subroutine entry **Note 2**	Yes	Yes
18	DBPC	PC value at exception/debug trap entry	Yes	Yes
19	DBPSW	PSW value at exception/debug trap entry	Yes	Yes
20	CTBP	CALLT base pointer	Yes	Yes
21 to 31	-	Reserved numbers for future function expansion (The operation is not guaranteed if accessed.)	No	No

Notes: 1. Since only one set of registers is available, the contents of these registers must be saved by the program when multiple interrupt servicing is enabled.

2. Since only one set of registers is available, the contents of these registers must be saved by the program when CALLT instructions nesting is used.

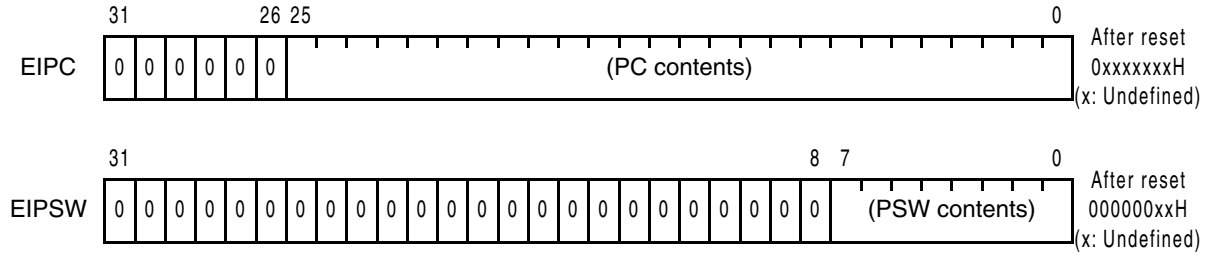
Caution: Even if bit 0 of EIPC, FEPC, or CTPC is set to (1) by the LDSR instruction, bit 0 is ignored during return with the RETI instruction following interrupt servicing (because bit 0 of PC is fixed to 0). If setting a value to EIPC, FEPC, and CTPC, set an even number (bit 0 = 0).

(1) Interrupt status saving registers (EIPC, EIPSW)

There are two context saving registers, EIPC and EIPSW.
Upon occurrence of a software exception or a maskable interrupt, the content of the program counter (PC) is saved to EIPC and the content of the program status word (PSW) is saved to EIPSW (upon occurrence of a non-maskable interrupt (NMI), the contents are saved to the NMI status saving registers (FEPC, FEPSW)).
The address of the next instruction following the instruction executed when a software exception or maskable interrupt occurs is saved to EIPC, except for the DIVH instruction (see Chapter 7 ”Interrupt/Exception Processing Function” on page 207).

Since there is only one set of interrupt status saving registers, the contents of these registers must be saved by the program when multiple interrupt servicing is enabled.
Bits 31 to 26 of EIPC and bits 31 to 8 of EIPSW are reserved (fixed to 0) for future function expansion.

Figure 3-3: Interrupt Status Saving Registers (EIPC, EIPSW)

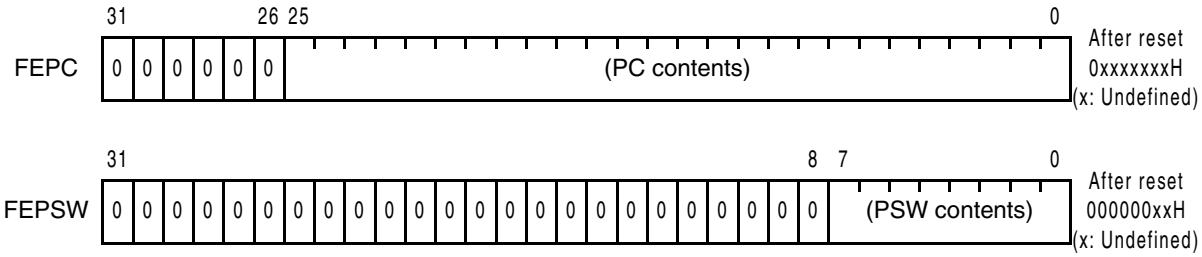


The values of EIPC and EIPSW are restored to PC and PSW during execution of a RETI instruction.

(2) NMI status saving registers (FEPC, FEPSW)

There are two NMI status saving registers, FEPC and FEPSW.
Upon occurrence of a non-maskable interrupt (NMI), the content of the program counter (PC) is saved to FEPC and the content of the program status word (PSW) is saved to FEPSW.
The address of the next instruction following the instruction executed when a non-maskable interrupt occurs is saved to FEPC, except for the DIVH instruction.
Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved (fixed to 0) for future function expansion.

Figure 3-4: NMI Status Saving Registers (FEPC, FEPSW)

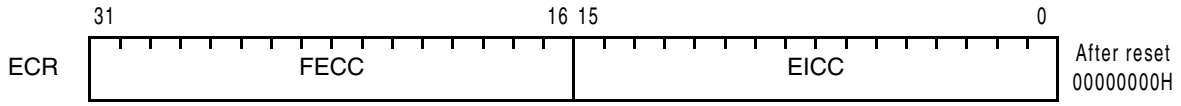


The values of FEPC and FEPSW are restored to PC and PSW during execution of a RETI instruction.

(3) Exception cause register (ECR)

Upon occurrence of an interrupt or an exception, the Exception Cause Register (ECR) holds the source of the interrupt or the exception. The value held by ECR is an exception code, coded for each interrupt source. This register is a read-only register, and thus data cannot be written to it using the LDSR instruction.

Figure 3-5: Interrupt Source Register (ECR)



Bit position	Bit name	Description
31 to 16	FECC	Non-maskable interrupt (NMI) exception code
15 to 0	EICC	Exception, maskable interrupt exception code

The list of exception codes is tabulated in **Table 7-1, "Interrupt/Exception Source List," on page 207**.

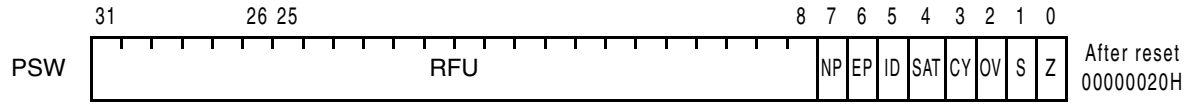
(4) Program status word (PSW)

The program status word (PSW) is a collection of flags that indicate the program status (instruction execution result) and the CPU status.

When the contents of this register are changed using the LDSR instruction, the new contents become valid immediately following completion of the LDSR instruction execution. However, if the ID flag is set to 1, interrupt request acknowledgement during LDSR instruction execution is prohibited.

Bits 31 to 8 are reserved (fixed to 0) for future function expansion.

Figure 3-6: Program Status Word (PSW)



Bit position	Bit name	Description
31 to 8	RFU	Reserved field. Fixed to 0.
7	NP	Indicates that non-maskable interrupt (NMI) servicing is in progress. This flag is set to 1 when a NMI request is acknowledged, and disables multiple interrupts. 0: NMI servicing not in progress 1: NMI servicing in progress
6	EP	Indicates that exception processing is in progress. This flag is set to 1 when an exception occurs. Moreover, interrupt requests can be acknowledged even when this bit is set. 0: Exception processing not in progress 1: Exception processing in progress
5	ID	Indicates whether maskable interrupt request acknowledgment is enabled. 0: Interrupt enabled 1: Interrupt disabled
4	SAT**Note**	Indicates that the result of executing a saturated operation instruction has overflowed and that the calculation result is saturated. Since this is a cumulative flag, it is set to 1 when the result of a saturated operation instruction becomes saturated, and it is not cleared to 0 even if the operation results of successive instructions do not become saturated. This flag is neither set nor cleared when arithmetic operation instructions are executed. 0: Not saturated 1: Saturated
3	CY	Indicates whether carry or borrow occurred as the result of an operation. 0: No carry or borrow occurred 1: Carry or borrow occurred
2	OV**Note**	Indicates whether overflow occurred during an operation. 0: No overflow occurred 1: Overflow occurred.
1	S**Note**	Indicates whether the result of an operation is negative. 0: Operation result is positive or 0. 1: Operation result is negative.
0	Z	Indicates whether operation result is 0. 0: Operation result is not 0. 1: Operation result is 0.

Note: During saturated operation, the saturated operation results are determined by the contents of the OV flag and S flag. The SAT flag is set to 1 only when the OV flag is set to 1 during saturated operation. This is explained on the following table.

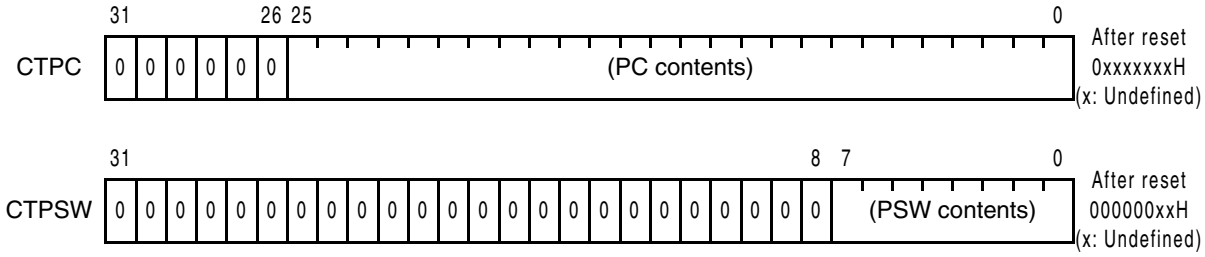
Table 3-3: Saturated Operation Results

Operation result status	Flag status			Saturated operation result
	SAT	OV	S	
Maximum positive value exceeded	1	1	0	7FFFFFFFH
Maximum negative value exceeded	1	1	1	80000000H
Positive (maximum value not exceeded)	Holds value before operation	0	0	Actual operation result
Negative (maximum value not exceeded)			1	

(5) CALLT execution status saving registers (CTPC, CTPSW)

There are two CALLT execution status saving registers, CTPC and CTPSW.
When the CALLT instruction is executed, the contents of the program counter (PC) are saved to CTPC, and the program status word (PSW) contents are saved to CTPSW.
The contents saved to CTPC consist of the address of the next instruction after the CALLT instruction.
Bits 31 to 26 CTPC and bits 31 to 8 of CTPSW are reserved (fixed to 0) for future function expansion.

Figure 3-7: CALLT Execution Status Saving Registers (CTPC, CTPSW)

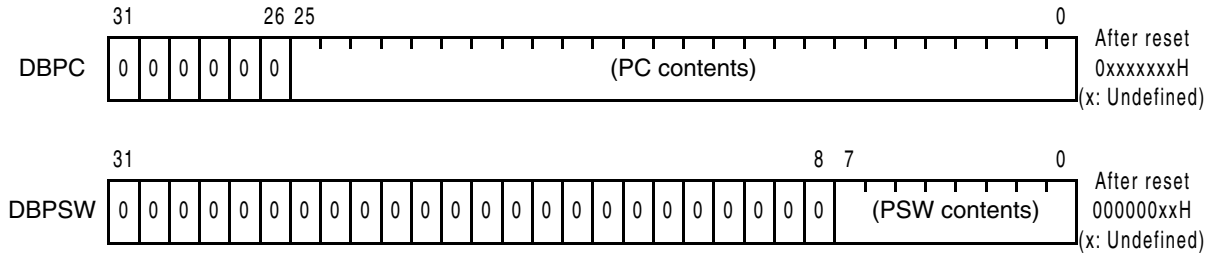


The values of CTPC and CTPSW are restored to PC and PSW during execution of the CTRET instruction.

(6) Exception/debug trap status saving registers (DBPC, DBPSW)

There are two exception/debug trap status saving registers, DBPC and DBPSW.
Upon occurrence of an exception trap or debug trap, the contents of the program counter (PC) are saved to DBPC, and the program status word (PSW) contents are saved to DBPSW.
The contents saved to DBPC consist of the address of the next instruction after the instruction executed when an exception trap or debug trap occurs.
Bits 31 to 26 of DBPC and bits 31 to 8 of DBPSW are reserved (fixed to 0) for future function expansion.

Figure 3-8: Exception/Debug Trap Status Saving Registers (DBPC, DBPSW)

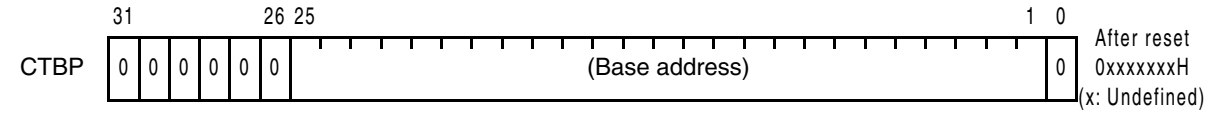


The values of DBPC and DBPSW are restored to PC and PSW during execution of the DBRET instruction.

(7) CALLT base pointer (CTBP)

The CALLT base pointer (CTBP) is used to specify CALLT table start address and generate target addresses (bit 0 is fixed to 0).
Bits 31 to 26 are reserved (fixed to 0) for future function expansion.

Figure 3-9: CALLT Base Pointer (CTBP)



3.2.3 Floating point arithmetic unit register set

The floating point arithmetic unit is provided with one flag register and one control register.

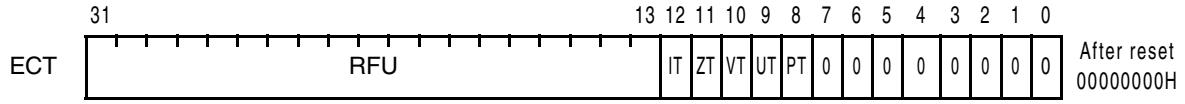
Table 3-4: Floating Point Arithmetic Unit Registers

Name	Usage	Operation
ECT	Control register	Sets the operation of the EFG register
EFG	Flag register	Holds the status of the FPU

(1) Floating point arithmetic control register (ECT)

This register is used for controlling the setting conditions of the TR flag:
TR is a logical OR between all the invalid operations the FPU can detect and each bit of ECT is a mask bit for each condition.

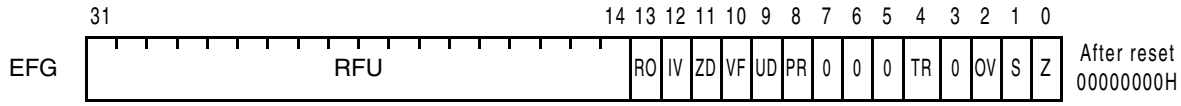
Figure 3-10: Floating Point Arithmetic Control Register (ECT)



Bit position	Bit name	Description
31 to 13	RFU	Reserved field. Fixed to 0.
12	IT	Enables invalid operation detection in the TR value calculation 0: IV is set when an invalid operation is detected 1: IV and TR are set when an invalid operation is detected
11	ZT	Enables zero divide operation detection in the TR value calculation 0: ZD is set when a zero divide operation is detected 1: ZD and TR are set when a zero divide operation is detected
10	VT	Enables overflow detection in the TR value calculation 0: VF is set when an overflow is detected 1: VF and TR are set when an overflow is detected
9	UT	Enables underflow detection in the TR value calculation 0: UD is set when an underflow is detected 1: UD and TR are set when an underflow is detected
8	PT	Enables accuracy fail detection in the TR value calculation 0: PR is set when an accuracy fail is detected 1: PR and TR are set when an accuracy fail is detected
7 to 0	0	Reserved field. Fixed to 0.

(2) Floating point arithmetic status register (EFG)

Figure 3-11: Floating Point Arithmetic Status Register (EFG)



Bit position	Bit name	Description
31 to 14	RFU	Reserved field. Fixed to 0.
13	RO	Running Operation: indicates whether the floating point arithmetic unit is running 0: operation in progress 1: FPU idle
12	IV	InValid operation: Indicates that an invalid operation has been requested. 0: normal operation 1: invalid operation detected
11	ZD	Zero Divide: Indicates whether a division by 0 has been detected. 0: normal operation 1: division by 0 detected
10	VF	oVerFlow: indicates that the result of executing a floating point operation has overflowed. 0: no overflow generated 1: overflow generated
9	UD	Undervalue: indicates that the result of executing a floating point operation has underflowed. 0: no underflow generated 1: underflow generated
8	PR	PRecision error: indicates that an accuracy failure occurred. 0: no accuracy failure occurred 1: accuracy failure occurred
7 to 5	0	Reserved field. Fixed to 0.
4	TR	This flag summarizes the state of the FPU: 0: normal state 1: abnormal condition detected: one of the bits 13 to 8 is set. The setting conditions of this flag depends on the ECT register value.
3	0	Reserved field. Fixed to 0.
2	OV	Indicates whether an overflow occurred during floating point to integer conversion 0: no overflow generated 1: overflow generated
1	S	Indicates whether floating point operation result is negative. 0: Operation result is not negative. 1: Operation result is negative.
0	Z	Indicates whether floating point operation result is 0. 0: Operation result is not 0. 1: Operation result is 0.

3.3 Operating Modes

The V850E/PH2 has the following operating modes.

3.3.1 Operating modes outline

(1) Normal operating mode

(a) Single-chip modes 0, 1

Access to the internal ROM is enabled.
In single-chip mode 0, after the system reset is released, each pin related to the bus interface enters the port mode, program execution branches to the reset entry address of the internal ROM, and instruction processing starts. By setting the PMCDH, PMCDL, PMCCS, PMCCT, and PMCCM registers to control mode by instruction, an external device can be connected to the external memory area.
In single-chip mode 1, after the system reset is released, each pin related to the bus interface enters the control mode, program execution branches to the external device's (memory) reset entry address, and instruction processing starts. The internal ROM area is mapped from address 100000H.

(b) ROM-less mode

After the system reset is released, each pin related to the bus interface enters the control mode, program execution branches to the external device's (memory) reset entry address, and instruction processing starts. Fetching of instructions and data access for internal ROM becomes impossible.
In ROM-less mode the data bus width is 32 bits.

(2) Flash memory programming mode

In this mode the internal flash memory can be written or erased with an external flash writer, using the CSIB0 or UARTC0 as serial interface.

3.3.2 Operation mode specification

The operation mode is specified according to the status of pins MODE0 to MODE2. In an application system fix the specification of these pins and do not change them during operation. Operation is not guaranteed if these pins are changed during operation.

MODE2	MODE1	MODE0	Mode	Remark
L	L	L	Single chip mode 0	Internal ROM area is allocated from address 00000000H.
L	L	H	Flash memory programming mode	CSIB0/IUARTC0 selected by MODE0 pin toggling.
L	H	L	ROM-less mode	External 32-bit data bus
L	H	H	Single chip mode 1	Internal ROM area is allocated from address 00100000H. External 32-bit data bus
other value than above			Setting prohibited	

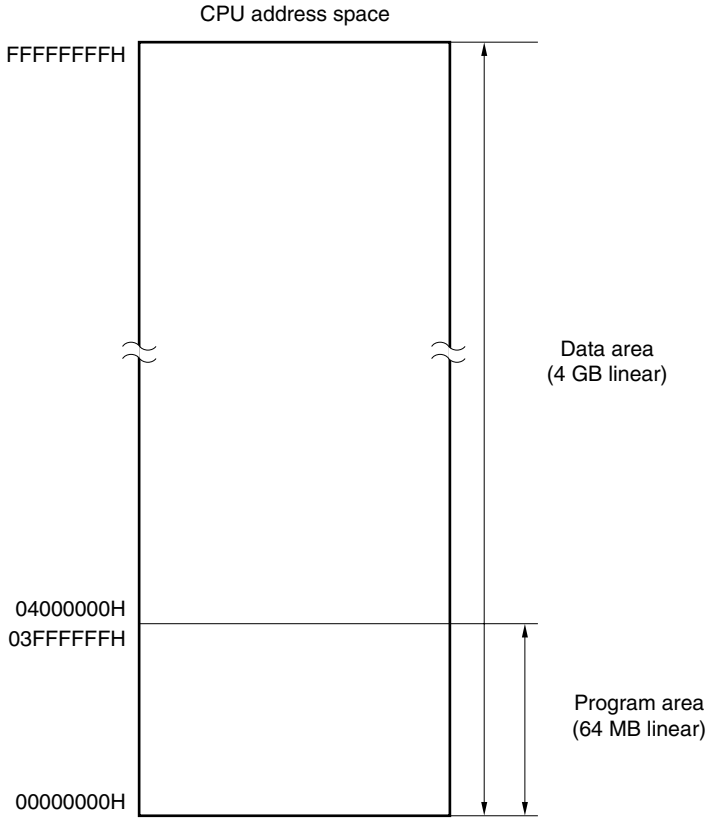
Remark: L: Low-level input
H: High-level input

3.4 Address Space

3.4.1 CPU address space

The CPU of the V850E/PH2 uses a 32-bit architecture and supports up to 4 GB of linear address space (data space) during operand addressing (data access). When addressing instructions, a linear address space (program space) of up to 64 MB is supported. However, both the program and data spaces include areas whose use is prohibited.
For details, refer to Figure 3-13, “Address Space Image,” on page 90.
Figure 3-12 shows the CPU address space.

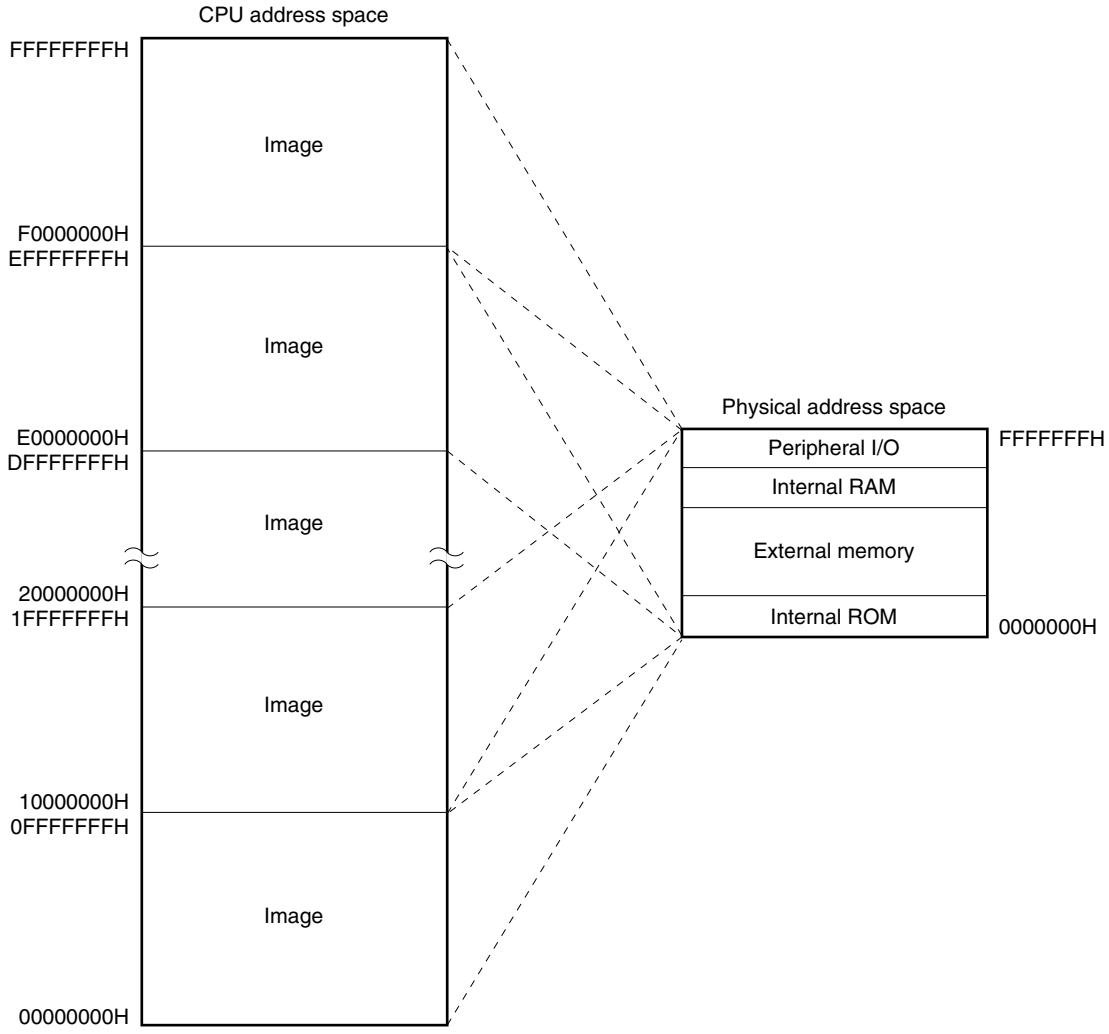
Figure 3-12: CPU Address Space



3.4.2 Images

When addressing an instruction address, up to 64 MB of linear address space (program space) and Internal RAM area are supported.
For operand addressing (data access), up to 4 GB of linear address space (data area) is supported. On this 4 GB address space, however, 256 MB physical address spaces can be seen as an image.
Therefore, whatever the values of bits 31 to 29 of an address may be, a physical address space of the same 256 MB is accessed.

Figure 3-13: Address Space Image



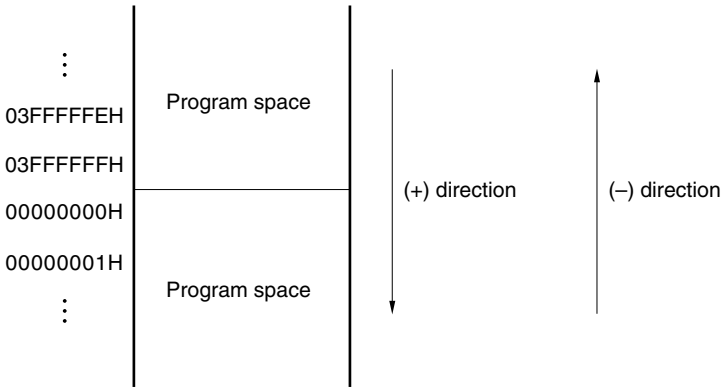
3.4.3 Wrap-around of CPU address space

(1) Program space

Of the 32 bits of the program counter (PC), the higher 6 bits are fixed to 0 and only the lower 26 bits are valid. Even if a carry or borrow occurs from bit 25 to bit 26 as a result of branch address calculation, the higher 6 bits ignore this and remain 0.
Therefore, the lower-limit address of the program space, 00000000H, and the upper-limit address, 03FFFFFFH, are contiguous addresses, and the program space is wrapped around at the boundary of these addresses.

Caution: No instructions can be fetched from the 4 KB area of 03FFF000H to 03FFFFFFH because this area is a peripheral I/O area. Therefore, do not execute any branch operation instructions in which the destination address will reside in any part of this area.

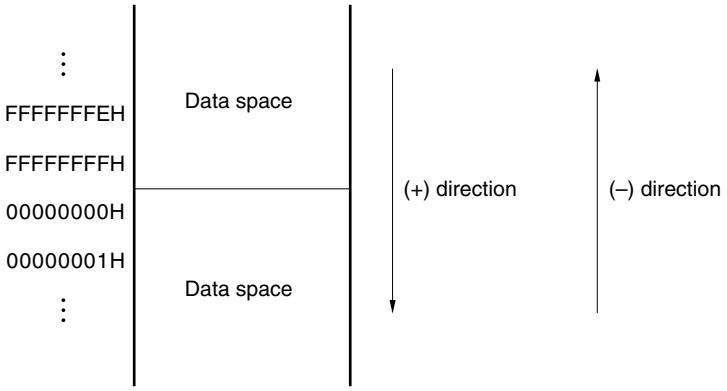
Figure 3-14: Program Space



(2) Data space

The result of an operand address calculation that exceeds 32 bits is truncated to 32 bits.
Therefore, the lower-limit address of the data space, address 00000000H, and the upper-limit address, FFFFFFFFH, are contiguous addresses, and the data space is wrapped around at the boundary of these addresses.

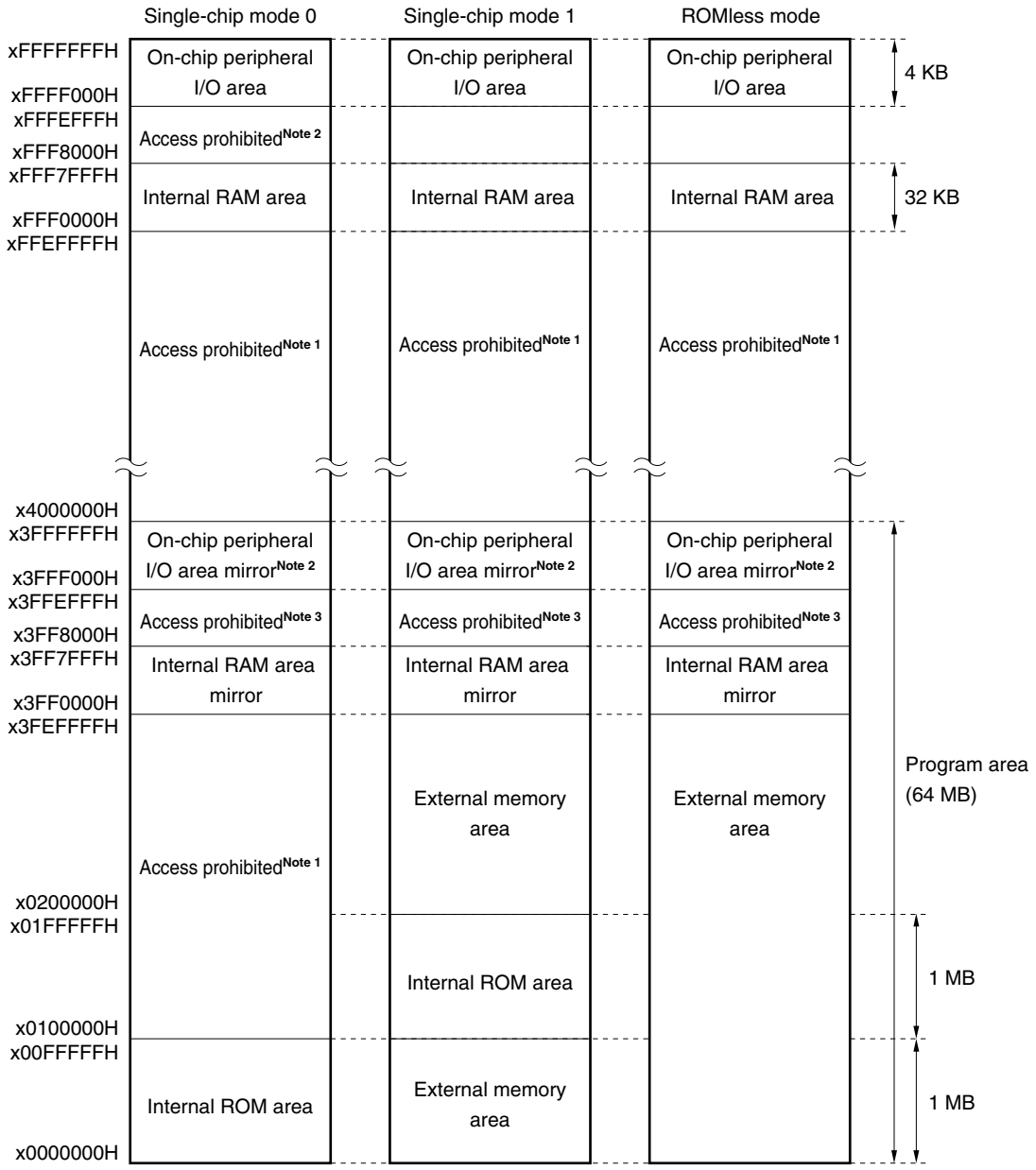
Figure 3-15: Data Space



3.4.4 Memory map

Areas are reserved in V850E/PH2 as shown in Figure 3-16. Each mode is specified by the MODE0 to MODE2 pins.

Figure 3-16: Memory Map



Notes: 1. By setting the PMCAL, PMCAH, PMCDL, PMCDH, PMCCS, PMCCT, and PMCCD port mode control registers to control mode, this area can be used as external memory area.

2. Accessing addresses 3FFF000H to 3FFFFFFH is prohibited. Specify addresses FFFF000H to FFFFFFFH to access the on-chip peripheral I/O.

3. The operation is not guaranteed if an access-prohibited area is accessed.

3.4.5 Areas

(1) Internal ROM area

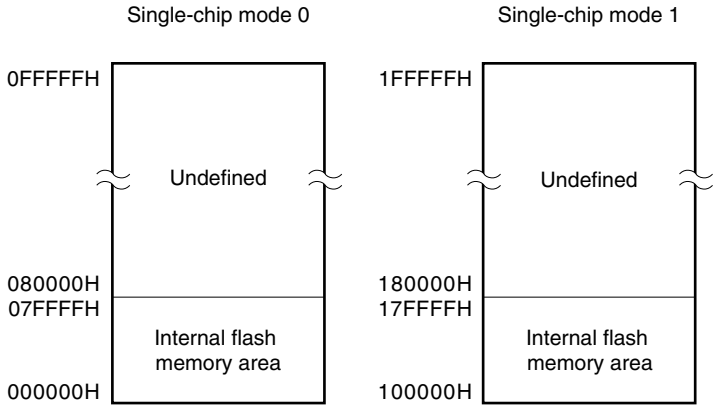
(a) Memory map

1 MB of internal ROM area, addresses 00000H to FFFFFH, is reserved. 512 KB are provided at addresses 000000H to 07FFFFH as physical internal ROM (flash memory).

Up to 1 MB of internal ROM/internal flash memory area is reserved.
512 KB are provided in the following addresses as physical internal ROM (flash memory).

- In single-chip mode 0: Addresses 000000H to 07FFFFH
  (addresses 080000H to 0FFFFFH are undefined)
- In single-chip mode 1: Addresses 0100000H to 017FFFFH
  (addresses 0180000H to 01FFFFFH are undefined)

Figure 3-17: Internal ROM / Internal Flash Memory Area



(b) Interrupt/exception table

The V850E/PH2 increases the interrupt response speed by assigning handler addresses corresponding to each interrupt/exception.
This group of handler addresses is called an interrupt/exception table. This table is located in the internal ROM area. When an interrupt/exception request is acknowledged, execution jumps to the handler address and the program written in that memory is executed.
For detailed list of the interrupt/exception sources and the corresponding handler addresses, please refer to **Table 7-1, "Interrupt/Exception Source List," on page 207**.

(2) Internal RAM area

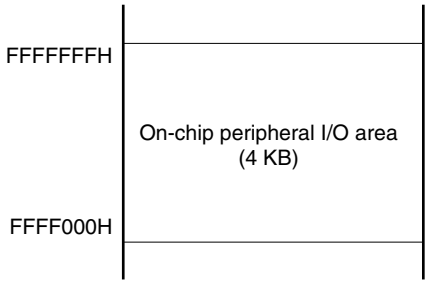
An area of 60 KB from FFF0000H to FFFEFFFH is reserved for the internal RAM area.
32 KB are provided at addresses FFF0000H to FFF7FFFH as physical internal RAM. The 32 KB area of 3FF0000H to 3FF7FFFH can be seen as an image of FFF0000H to FFF7FFFH.

(3) On-chip peripheral I/O area (SFR area)

A 4 KB area from FFFF000H to FFFFFFFH is provided as the 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 3-18: On-Chip Peripheral I/O Area



Peripheral I/O registers assigned with functions such as on-chip peripheral I/O operation mode specification and state monitoring are mapped to the on-chip peripheral I/O area. Program fetches are not allowed in this area.

Cautions: 1. For registers in which byte access is possible, if half-word access is executed, the higher 8 bits become undefined during a read operation, and the lower 8 bits of data are written to the register during a write operation. Do not access an 8-bit register in half-word units.

2. Addresses that are not defined as registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.

3.4.6 Peripheral I/O registers list

.

Table 3-5: Peripheral I/O Registers (1/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFF000H	PAL	Port register AL	R/W			×		0000H
FFFFF000H	PALL	Port register ALL	R/W	×	×			00H
FFFFF001H	PALH	Port register ALH	R/W	×	×			00H
FFFFF002H	PAH	Port register AH	R/W	×	×			00H
FFFFF004H	PDL	Port register DL	R/W			×		0000H
FFFFF004H	PDLL	Port register DLL	R/W	×	×			00H
FFFFF005H	PDLH	Port register DLH	R/W	×	×			00H
FFFFF006H	PDH	Port register DH	R/W			×		0000H
FFFFF006H	PDHL	Port register DHL	R/W	×	×			00H
FFFFF007H	PDHH	Port register DHH	R/W	×	×			00H
FFFFF008H	PCS	Port register CS	R/W	×	×			00H
FFFFF00AH	PCT	Port register CT	R/W	×	×			00H
FFFFF00CH	PCM	Port register CM	R/W	×	×			00H
FFFFF00EH	PCD	Port register CD	R/W	×	×			00H
FFFFF020H	PMAL	Port mode register AL	R/W			×		FFFFH
FFFFF020H	PMALL	Port mode register ALL	R/W	×	×			FFH
FFFFF021H	PMALH	Port mode register ALH	R/W	×	×			FFH
FFFFF022H	PMAH	Port mode register AH	R/W	×	×			FFH
FFFFF024H	PMDL	Port mode register DL	R/W			×		FFFFH
FFFFF024H	PMDLL	Port mode register DLL	R/W	×	×			FFH
FFFFF025H	PMDLH	Port mode register DLH	R/W	×	×			FFH
FFFFF026H	PMDH	Port mode register DH	R/W			×		FFFFH
FFFFF026H	PMDHL	Port mode register DHL	R/W	×	×			FFH
FFFFF027H	PMDHH	Port mode register DHH	R/W	×	×			FFH
FFFFF028H	PMCS	Port mode register CS	R/W	×	×			FFH
FFFFF02AH	PMCT	Port mode register CT	R/W	×	×			FFH
FFFFF02CH	PMCM	Port mode register CM	R/W	×	×			FFH
FFFFF02EH	PMCD	Port mode register CD	R/W	×	×			FFH
FFFFF040H	PMCAL	Port mode control register AL	R/W			×		0000H
FFFFF040H	PMCALL	Port mode control register ALL	R/W	×	×			00H
FFFFF041H	PMCALH	Port mode control register ALH	R/W	×	×			00H
FFFFF042H	PMCAH	Port mode control register AH	R/W	×	×			00H
FFFFF044H	PMCDL	Port mode control register DL	R/W			×		0000H
FFFFF044H	PMCDLL	Port mode control register DLL	R/W	×	×			00H
FFFFF045H	PMCDLH	Port mode control register DLH	R/W	×	×			00H
FFFFF046H	PMCDH	Port mode control register DH	R/W			×		0000H
FFFFF046H	PMCDHL	Port mode control register DHL	R/W	×	×			00H
FFFFF047H	PMCDHH	Port mode control register DHH	R/W	×	×			00H

Table 3-5: Peripheral I/O Registers (2/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFF048H	PMCCS	Port mode control register CS	R/W	×	×			00H
FFFFF04AH	PMCCT	Port mode control register CT	R/W	×	×			00H
FFFFF04CH	PMCCM	Port mode control register CM	R/W	×	×			00H
FFFFF04EH	PMCCD	Port mode control register CD	R/W	×	×			00H
FFFFF060H	CSC0	Chip area select control register 0	R/W			×		2C11H
FFFFF062H	CSC1	Chip area select control register 1	R/W			×		2C11H
FFFFF064H	BPC	Peripheral area select control register	R/W			×		0FFFH
FFFFF066H	BSC	Bus size configuration register	R/W			×		AAAAH
FFFFF068H	BEC	Endian configuration register	R/W			×		0000H
FFFFF06EH	VSWC	System wait control register	R/W	×	×			77H
FFFFF100H	IMR0	Interrupt mask register 0	R/W			×		FFFFH
FFFFF100H	IMR0L	Interrupt mask register 0L	R/W	×	×			FFH
FFFFF101H	IMR0H	Interrupt mask register 0H	R/W	×	×			FFH
FFFFF102H	IMR1	Interrupt mask register 1	R/W			×		FFFFH
FFFFF102H	IMR1L	Interrupt mask register 1L	R/W	×	×			FFH
FFFFF103H	IMR1H	Interrupt mask register 1H	R/W	×	×			FFH
FFFFF104H	IMR2	Interrupt mask register 2	R/W			×		FFFFH
FFFFF104H	IMR2L	Interrupt mask register 2L	R/W	×	×			FFH
FFFFF105H	IMR2H	Interrupt mask register 2H	R/W	×	×			FFH
FFFFF106H	IMR3	Interrupt mask register 3	R/W			×		FFFFH
FFFFF106H	IMR3L	Interrupt mask register 3L	R/W	×	×			FFH
FFFFF107H	IMR3H	Interrupt mask register 3H	R/W	×	×			FFH
FFFFF108H	IMR4	Interrupt mask register 4	R/W			×		FFFFH
FFFFF108H	IMR4L	Interrupt mask register 4L	R/W	×	×			FFH
FFFFF109H	IMR4H	Interrupt mask register 4H	R/W	×	×			FFH
FFFFF10AH	IMR5	Interrupt mask register 5	R/W			×		FFFFH
FFFFF10AH	IMR5L	Interrupt mask register 5L	R/W	×	×			FFH
FFFFF10BH	IMR5H	Interrupt mask register 5H	R/W	×	×			FFH
FFFFF10CH	IMR6	Interrupt mask register 6	R/W			×		FFFFH
FFFFF10CH	IMR6L	Interrupt mask register 6L	R/W	×	×			FFH
FFFFF10DH	IMR6H	Interrupt mask register 6H	R/W	×	×			FFH
FFFFF10EH	IMR7	Interrupt mask register 7	R/W			×		FFFFH
FFFFF10EH	IMR7L	Interrupt mask register 7L	R/W	×	×			FFH
FFFFF10FH	IMR7H	Interrupt mask register 7H	R/W	×	×			FFH
FFFFF110H	PIC0	Interrupt control register 0	R/W	×	×			47H
FFFFF112H	PIC1	Interrupt control register 1	R/W	×	×			47H
FFFFF114H	PIC2	Interrupt control register 2	R/W	×	×			47H
FFFFF116H	PIC3	Interrupt control register 3	R/W	×	×			47H
FFFFF118H	PIC4	Interrupt control register 4	R/W	×	×			47H
FFFFF11AH	PIC5	Interrupt control register 5	R/W	×	×			47H

Table 3-5: Peripheral I/O Registers (3/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFF11CH	PIC6	Interrupt control register 6	R/W	×	×			47H
FFFFF11EH	PIC7	Interrupt control register 7	R/W	×	×			47H
FFFFF120H	PIC8	Interrupt control register 8	R/W	×	×			47H
FFFFF122H	PIC9	Interrupt control register 9	R/W	×	×			47H
FFFFF124H	PIC10	Interrupt control register 10	R/W	×	×			47H
FFFFF126H	PIC11	Interrupt control register 11	R/W	×	×			47H
FFFFF128H	PIC12	Interrupt control register 12	R/W	×	×			47H
FFFFF12AH	PIC13	Interrupt control register 13	R/W	×	×			47H
FFFFF12CH	PIC14	Interrupt control register 14	R/W	×	×			47H
FFFFF12EH	PIC15	Interrupt control register 15	R/W	×	×			47H
FFFFF130H	PIC16	Interrupt control register 16	R/W	×	×			47H
FFFFF132H	PIC17	Interrupt control register 17	R/W	×	×			47H
FFFFF134H	PIC18	Interrupt control register 18	R/W	×	×			47H
FFFFF136H	PIC19	Interrupt control register 19	R/W	×	×			47H
FFFFF138H	PIC20	Interrupt control register 20	R/W	×	×			47H
FFFFF13AH	PIC21	Interrupt control register 21	R/W	×	×			47H
FFFFF13CH	PIC22	Interrupt control register 22	R/W	×	×			47H
FFFFF13EH	PIC23	Interrupt control register 23	R/W	×	×			47H
FFFFF140H	PIC24	Interrupt control register 24	R/W	×	×			47H
FFFFF142H	PIC25	Interrupt control register 25	R/W	×	×			47H
FFFFF144H	PIC26	Interrupt control register 26	R/W	×	×			47H
FFFFF146H	PIC27	Interrupt control register 27	R/W	×	×			47H
FFFFF148H	PIC28	Interrupt control register 28	R/W	×	×			47H
FFFFF14AH	PIC29	Interrupt control register 29	R/W	×	×			47H
FFFFF14CH	PIC30	Interrupt control register 30	R/W	×	×			47H
FFFFF14EH	PIC31	Interrupt control register 31	R/W	×	×			47H
FFFFF150H	PIC32	Interrupt control register 32	R/W	×	×			47H
FFFFF152H	PIC33	Interrupt control register 33	R/W	×	×			47H
FFFFF154H	PIC34	Interrupt control register 34	R/W	×	×			47H
FFFFF156H	PIC35	Interrupt control register 35	R/W	×	×			47H
FFFFF158H	PIC36	Interrupt control register 36	R/W	×	×			47H
FFFFF15AH	PIC37	Interrupt control register 37	R/W	×	×			47H
FFFFF15CH	PIC38	Interrupt control register 38	R/W	×	×			47H
FFFFF15EH	PIC39	Interrupt control register 39	R/W	×	×			47H
FFFFF160H	PIC40	Interrupt control register 40	R/W	×	×			47H
FFFFF162H	PIC41	Interrupt control register 41	R/W	×	×			47H
FFFFF164H	PIC42	Interrupt control register 42	R/W	×	×			47H
FFFFF166H	PIC43	Interrupt control register 43	R/W	×	×			47H
FFFFF168H	PIC44	Interrupt control register 44	R/W	×	×			47H
FFFFF16AH	PIC45	Interrupt control register 45	R/W	×	×			47H

Table 3-5: Peripheral I/O Registers (4/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFF16CH	PIC46	Interrupt control register 46	R/W	×	×			47H
FFFFF16EH	PIC47	Interrupt control register 47	R/W	×	×			47H
FFFFF170H	PIC48	Interrupt control register 48	R/W	×	×			47H
FFFFF172H	PIC49	Interrupt control register 49	R/W	×	×			47H
FFFFF174H	PIC50	Interrupt control register 50	R/W	×	×			47H
FFFFF176H	PIC51	Interrupt control register 51	R/W	×	×			47H
FFFFF178H	PIC52	Interrupt control register 52	R/W	×	×			47H
FFFFF17AH	PIC53	Interrupt control register 53	R/W	×	×			47H
FFFFF17CH	PIC54	Interrupt control register 54	R/W	×	×			47H
FFFFF17EH	PIC55	Interrupt control register 55	R/W	×	×			47H
FFFFF180H	PIC56	Interrupt control register 56	R/W	×	×			47H
FFFFF182H	PIC57	Interrupt control register 57	R/W	×	×			47H
FFFFF184H	PIC58	Interrupt control register 58	R/W	×	×			47H
FFFFF186H	PIC59	Interrupt control register 59	R/W	×	×			47H
FFFFF188H	PIC60	Interrupt control register 60	R/W	×	×			47H
FFFFF18AH	PIC61	Interrupt control register 61	R/W	×	×			47H
FFFFF18CH	PIC62	Interrupt control register 62	R/W	×	×			47H
FFFFF18EH	PIC63	Interrupt control register 63	R/W	×	×			47H
FFFFF190H	PIC64	Interrupt control register 64	R/W	×	×			47H
FFFFF192H	PIC65	Interrupt control register 65	R/W	×	×			47H
FFFFF194H	PIC66	Interrupt control register 66	R/W	×	×			47H
FFFFF196H	PIC67	Interrupt control register 67	R/W	×	×			47H
FFFFF198H	PIC68	Interrupt control register 68	R/W	×	×			47H
FFFFF19AH	PIC69	Interrupt control register 69	R/W	×	×			47H
FFFFF19CH	PIC70	Interrupt control register 70	R/W	×	×			47H
FFFFF19EH	PIC71	Interrupt control register 71	R/W	×	×			47H
FFFFF1A0H	PIC72	Interrupt control register 72	R/W	×	×			47H
FFFFF1A2H	PIC73	Interrupt control register 73	R/W	×	×			47H
FFFFF1A4H	PIC74	Interrupt control register 74	R/W	×	×			47H
FFFFF1A6H	PIC75	Interrupt control register 75	R/W	×	×			47H
FFFFF1A8H	PIC76	Interrupt control register 76	R/W	×	×			47H
FFFFF1AAH	PIC77	Interrupt control register 77	R/W	×	×			47H
FFFFF1ACH	PIC78	Interrupt control register 78	R/W	×	×			47H
FFFFF1AEH	PIC79	Interrupt control register 79	R/W	×	×			47H
FFFFF1B0H	PIC80	Interrupt control register 80	R/W	×	×			47H
FFFFF1B2H	PIC81	Interrupt control register 81	R/W	×	×			47H
FFFFF1B4H	PIC82	Interrupt control register 82	R/W	×	×			47H
FFFFF1B6H	PIC83	Interrupt control register 83	R/W	×	×			47H
FFFFF1B8H	PIC84	Interrupt control register 84	R/W	×	×			47H
FFFFF1BAH	PIC85	Interrupt control register 85	R/W	×	×			47H

Table 3-5: Peripheral I/O Registers (5/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFF1BCH	PIC86	Interrupt control register 86	R/W	×	×			47H
FFFFF1BEH	PIC87	Interrupt control register 87	R/W	×	×			47H
FFFFF1C0H	PIC88	Interrupt control register 88	R/W	×	×			47H
FFFFF1C2H	PIC89	Interrupt control register 89	R/W	×	×			47H
FFFFF1C4H	PIC90	Interrupt control register 90	R/W	×	×			47H
FFFFF1C6H	PIC91	Interrupt control register 91	R/W	×	×			47H
FFFFF1C8H	PIC92	Interrupt control register 92	R/W	×	×			47H
FFFFF1CAH	PIC93	Interrupt control register 93	R/W	×	×			47H
FFFFF1CCH	PIC94	Interrupt control register 94	R/W	×	×			47H
FFFFF1CEH	PIC95	Interrupt control register 95	R/W	×	×			47H
FFFFF1D0H	PIC96	Interrupt control register 96	R/W	×	×			47H
FFFFF1D2H	PIC97	Interrupt control register 97	R/W	×	×			47H
FFFFF1D4H	PIC98	Interrupt control register 98	R/W	×	×			47H
FFFFF1D6H	PIC99	Interrupt control register 99	R/W	×	×			47H
FFFFF1D8H	PIC100	Interrupt control register 100	R/W	×	×			47H
FFFFF1DAH	PIC101	Interrupt control register 101	R/W	×	×			47H
FFFFF1DCH	PIC102	Interrupt control register 102	R/W	×	×			47H
FFFFF1DEH	PIC103	Interrupt control register 103	R/W	×	×			47H
FFFFF1E0H	PIC104	Interrupt control register 104	R/W	×	×			47H
FFFFF1E2H	PIC105	Interrupt control register 105	R/W	×	×			47H
FFFFF1FAH	ISPR	Interrupt service priority register	R	×	×			00H
FFFFF1FCH	PRCMD	Command register	W		×			undefined
FFFFF200H	ADM00	A/D converter 0 mode register 0	R/W	×	×			00H
FFFFF201H	ADM01	A/D converter 0 mode register 1	R/W	×	×			00H
FFFFF202H	ADM02	A/D converter 0 mode register 2	R/W	×	×			00H
FFFFF210H	ADCR00	A/D conversion result register 00	R			×		undefined
FFFFF211H	ADCR00H	A/D conversion result register 00H	R		×			undefined
FFFFF212H	ADCR01	A/D conversion result register 01	R			×		undefined
FFFFF213H	ADCR01H	A/D conversion result register 01H	R		×			undefined
FFFFF214H	ADCR02	A/D conversion result register 02	R			×		undefined
FFFFF215H	ADCR02H	A/D conversion result register 02H	R		×			undefined
FFFFF216H	ADCR03	A/D conversion result register 03	R			×		undefined
FFFFF217H	ADCR03H	A/D conversion result register 03H	R		×			undefined
FFFFF218H	ADCR04	A/D conversion result register 04	R			×		undefined
FFFFF219H	ADCR04H	A/D conversion result register 04H	R		×			undefined
FFFFF21AH	ADCR05	A/D conversion result register 05	R			×		undefined
FFFFF21BH	ADCR05H	A/D conversion result register 05H	R		×			undefined
FFFFF21CH	ADCR06	A/D conversion result register 06	R			×		undefined
FFFFF21DH	ADCR06H	A/D conversion result register 06H	R		×			undefined

Table 3-5: Peripheral I/O Registers (6/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFF21EH	ADCR07	A/D conversion result register 07	R			×		undefined
FFFFF21FH	ADCR07H	A/D conversion result register 07H	R		×			undefined
FFFFF220H	ADCR08	A/D conversion result register 08	R			×		undefined
FFFFF221H	ADCR08H	A/D conversion result register 08H	R		×			undefined
FFFFF222H	ADCR09	A/D conversion result register 09	R			×		undefined
FFFFF223H	ADCR09H	A/D conversion result register 09H	R		×			undefined
FFFFF22EH	ADDMA0	A/D conversion result register 0 for DMA	R			×		undefined
FFFFF240H	ADM10	A/D converter 1 mode register 0	R/W	×	×			00H
FFFFF241H	ADM11	A/D converter 1 mode register 1	R/W	×	×			00H
FFFFF242H	ADM12	A/D converter 1 mode register 2	R/W	×	×			00H
FFFFF250H	ADCR10	A/D conversion result register 10	R			×		undefined
FFFFF251H	ADCR10H	A/D conversion result register 10H	R		×			undefined
FFFFF252H	ADCR11	A/D conversion result register 11	R			×		undefined
FFFFF253H	ADCR11H	A/D conversion result register 11H	R		×			undefined
FFFFF254H	ADCR112	A/D conversion result register 12	R			×		undefined
FFFFF255H	ADCR12H	A/D conversion result register 12H	R		×			undefined
FFFFF256H	ADCR13	A/D conversion result register 13	R			×		undefined
FFFFF257H	ADCR13H	A/D conversion result register 13H	R		×			undefined
FFFFF258H	ADCR14	A/D conversion result register 14	R			×		undefined
FFFFF259H	ADCR14H	A/D conversion result register 14H	R		×			undefined
FFFFF25AH	ADCR15	A/D conversion result register 15	R			×		undefined
FFFFF25BH	ADCR15H	A/D conversion result register 15H	R		×			undefined
FFFFF25CH	ADCR16	A/D conversion result register 16	R			×		undefined
FFFFF25DH	ADCR16H	A/D conversion result register 16H	R		×			undefined
FFFFF25EH	ADCR17	A/D conversion result register 17	R			×		undefined
FFFFF25FH	ADCR17H	A/D conversion result register 17H	R		×			undefined
FFFFF260H	ADCR18	A/D conversion result register 18	R			×		undefined
FFFFF261H	ADCR18H	A/D conversion result register 18H	R		×			undefined
FFFFF262H	ADCR19	A/D conversion result register 19	R			×		undefined
FFFFF263H	ADCR19H	A/D conversion result register 19H	R		×			undefined
FFFFF26EH	ADDMA1	A/D conversion result register 1 for DMA	R			×		undefined
FFFFF270H	ADTRSEL0	A/D trigger select register 0	R/W	×	×			00H
FFFFF272H	ADTRSEL1	A/D trigger select register 1	R/W	×	×			00H
FFFFF300H	MAR0	Memory transfer start address register 0	R/W			×		undefined
FFFFF302H	MAR1	Memory transfer start address register 1	R/W			×		undefined
FFFFF304H	MAR2	Memory transfer start address register 2	R/W			×		undefined
FFFFF306H	MAR3	Memory transfer start address register 3	R/W			×		undefined
FFFFF308H	MAR4	Memory transfer start address register 4	R/W			×		undefined
FFFFF30AH	MAR5	Memory transfer start address register 5	R/W			×		undefined
FFFFF30CH	MAR6	Memory transfer start address register 6	R/W			×		undefined

Table 3-5: Peripheral I/O Registers (7/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFF30EH	MAR7	Memory transfer start address register 7	R/W			×		undefined
FFFFF314H	SAR2	SFR transfer start address register 2	R/W	×	×			undefined
FFFFF316H	SAR3	SFR transfer start address register 3	R/W	×	×			undefined
FFFFF320H	DTCR0	DMA transfer count register 0	R/W	×	×			undefined
FFFFF322H	DTCR1	DMA transfer count register 1	R/W	×	×			undefined
FFFFF324H	DTCR2	DMA transfer count register 2	R/W	×	×			undefined
FFFFF326H	DTCR3	DMA transfer count register 3	R/W	×	×			undefined
FFFFF328H	DTCR4	DMA transfer count register 4	R/W	×	×			undefined
FFFFF32AH	DTCR5	DMA transfer count register 5	R/W	×	×			undefined
FFFFF32CH	DTCR6	DMA transfer count register 6	R/W	×	×			undefined
FFFFF32EH	DTCR7	DMA transfer count register 7	R/W	×	×			undefined
FFFFF330H	DMAMC	DMA mode control register	R/W	×	×			00H
FFFFF332H	DMAS	DMA status register	R/W	×	×			00H
FFFFF334H	DMADSC	DMA data size control register	R/W	×	×			00H
FFFFF348H	DTFR4	DMA trigger factor register 4	R/W	×	×			00H
FFFFF34AH	DTFR5	DMA trigger factor register 5	R/W	×	×			00H
FFFFF34CH	DTFR6	DMA trigger factor register 6	R/W	×	×			00H
FFFFF34EH	DTFR7	DMA trigger factor register 7	R/W	×	×			00H
FFFFF400H	P0	Port register 0	R	×	×			undefined
FFFFF402H	P1	Port register 1	R/W	×	×			undefined
FFFFF404H	P2	Port register 2	R/W	×	×			undefined
FFFFF406H	P3	Port register 3	R/W	×	×			undefined
FFFFF408H	P4	Port register 4	R/W	×	×			undefined
FFFFF40AH	P5	Port register 5	R/W	×	×			undefined
FFFFF40CH	P6	Port register 6	R/W	×	×			undefined
FFFFF40EH	P7	Port register 7	R/W	×	×			undefined
FFFFF410H	P8	Port register 8	R/W	×	×			undefined
FFFFF412H	P9	Port register 9	R/W	×	×			undefined
FFFFF414H	P10	Port register 10	R/W	×	×			undefined
FFFFF422H	PM1	Port mode register 1	R/W	×	×			FFH
FFFFF424H	PM2	Port mode register 2	R/W	×	×			FFH
FFFFF426H	PM3	Port mode register 3	R/W	×	×			FFH
FFFFF428H	PM4	Port mode register 4	R/W	×	×			FFH
FFFFF42AH	PM5	Port mode register 5	R/W	×	×			FFH
FFFFF42CH	PM6	Port mode register 6	R/W	×	×			FFH
FFFFF42EH	PM7	Port mode register 7	R/W	×	×			FFH
FFFFF430H	PM8	Port mode register 8	R/W	×	×			FFH
FFFFF432H	PM9	Port mode register 9	R/W	×	×			FFH
FFFFF434H	PM10	Port mode register 10	R/W	×	×			FFH
FFFFF442H	PMC1	Port mode control register 1	R/W	×	×			00H

Table 3-5: Peripheral I/O Registers (8/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFF444H	PMC2	Port mode control register 2	R/W	×	×			00H
FFFFF446H	PMC3	Port mode control register 3	R/W	×	×			00H
FFFFF448H	PMC4	Port mode control register 4	R/W	×	×			00H
FFFFF44AH	PMC5	Port mode control register 5	R/W	×	×			00H
FFFFF44CH	PMC6	Port mode control register 6	R/W	×	×			00H
FFFFF44EH	PMC7	Port mode control register 7	R/W	×	×			00H
FFFFF450H	PMC8	Port mode control register 8	R/W	×	×			00H
FFFFF452H	PMC9	Port mode control register 9	R/W	×	×			00H
FFFFF454H	PMC10	Port mode control register 10	R/W	×	×			00H
FFFFF480H	BCT0	Bus cycle type configuration register 0	R/W			×		CCCCH
FFFFF482H	BCT1	Bus cycle type configuration register 1	R/W			×		CCCCH
FFFFF484H	DWC0	Data wait control register 0	R/W			×		7777H
FFFFF486H	DWC1	Data wait control register 1	R/W			×		7777H
FFFFF488H	AWC	Address wait control register	R/W			×		0000H
FFFFF48AH	BCC	Bus and cycle control register	R/W			×		AAAAH
FFFFF48EH	DVC	Bus clock dividing control register	R/W		×			01H
FFFFF4C0H	RAMERR	iRAM parity error flag register	R/W	×	×			00H
FFFFF4C2H	RAMPADD	iRAM parity error address register	R/W			×		8000H
FFFFF580H	TR0CTL0	TMR0 control register 0	R/W	×	×			00H
FFFFF581H	TR0CTL1	TMR0 control register 1	R/W	×	×			00H
FFFFF582H	TR0IOC0	TMR0 I/O control register 0	R/W	×	×			00H
FFFFF585H	TR0IOC3	TMR0 I/O control register 3	R/W	×	×			00H
FFFFF586H	TR0IOC4	TMR0 I/O control register 4	R/W	×	×			00H
FFFFF587H	TR0OPT0	TMR0 option register 0	R/W	×	×			00H
FFFFF588H	TR0OPT2	TMR0 option register 2	R/W	×	×			00H
FFFFF589H	TR0OPT3	TMR0 option register 3	R/W	×	×			00H
FFFFF58CH	TR0OPT6	TMR0 option register 6	R/W	×	×			00H
FFFFF58DH	TR0OPT7	TMR0 option register 7	R/W	×	×			00H
FFFFF58EH	TR0OPT1	TMR0 option register 1	R/W	×	×			0000H
FFFFF590H	TR0CCR5	TMR0 capture/compare register 5	R/W			×		0000H
FFFFF592H	TR0CCR4	TMR0 capture/compare register 4	R/W			×		0000H
FFFFF598H	TR0CCR0	TMR0 capture/compare register 0	R/W			×		0000H
FFFFF59AH	TR0CCR3	TMR0 capture/compare register 3	R/W			×		0000H
FFFFF59CH	TR0CCR2	TMR0 capture/compare register 2	R/W			×		0000H
FFFFF59EH	TR0CCR1	TMR0 capture/compare register 1	R/W			×		0000H
FFFFF5A0H	TR0DTC0	TMR0 dead time set register 0	R/W			×		0000H
FFFFF5A2H	TR0DTC1	TMR0 dead time set register 1	R/W			×		0000H
FFFFF5A4H	TR0CNT	TMR0 timer counter read register	R/W			×		0000H
FFFFF5A6H	TR0SBC	TMR0 timer sub-counter read register	R/W			×		0000H
FFFFF5C0H	TR1CTL0	TMR1 control register 0	R/W	×	×			00H

Table 3-5: Peripheral I/O Registers (9/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFF5C1H	TR1CTL1	TMR1 control register 1	R/W	×	×			00H
FFFFF5C2H	TR1IOC0	TMR1 I/O control register 0	R/W	×	×			00H
FFFFF5C3H	TR1IOC1	TMR1 I/O control register 1	R/W	×	×			00H
FFFFF5C4H	TR1IOC2	TMR1 I/O control register 2	R/W	×	×			00H
FFFFF5C5H	TR1IOC3	TMR1 I/O control register 3	R/W	×	×			00H
FFFFF5C6H	TR1IOC4	TMR1 I/O control register 4	R/W	×	×			00H
FFFFF5C7H	TR1OPT0	TMR1 option register 0	R/W	×	×			00H
FFFFF5C8H	TR1OPT2	TMR1 option register 2	R/W	×	×			00H
FFFFF5C9H	TR1OPT3	TMR1 option register 3	R/W	×	×			00H
FFFFF5CCH	TR1OPT6	TMR1 option register 6	R/W	×	×			00H
FFFFF5CDH	TR1OPT7	TMR1 option register 7	R/W	×	×			00H
FFFFF5CEH	TR1OPT1	TMR1 option register 1	R/W	×	×			0000H
FFFFF5D0H	TR1CCR5	TMR1 capture/compare register 5	R/W			×		0000H
FFFFF5D2H	TR1CCR4	TMR1 capture/compare register 4	R/W			×		0000H
FFFFF5D8H	TR1CCR0	TMR1 capture/compare register 0	R/W			×		0000H
FFFFF5DAH	TR1CCR3	TMR1 capture/compare register 3	R/W			×		0000H
FFFFF5DCH	TR1CCR2	TMR1 capture/compare register 2	R/W			×		0000H
FFFFF5DEH	TR1CCR1	TMR1 capture/compare register 1	R/W			×		0000H
FFFFF5E0H	TR1DTC0	TMR1 dead time set register 0	R/W			×		0000H
FFFFF5E2H	TR1DTC1	TMR1 dead time set register 1	R/W			×		0000H
FFFFF5E4H	TR1CNT	TMR1 timer counter read register	R			×		0000H
FFFFF5E6H	TR1SBC	TMR1 timer sub-counter read register	R			×		0000H
FFFFF600H	TP0CTL0	TMP0 timer control register 0	R/W	×	×			00H
FFFFF601H	TP0CTL1	TMP0 timer control register 1	R/W	×	×			00H
FFFFF602H	TP0IOC0	TMP0 I/O control register 0	R/W	×	×			00H
FFFFF603H	TP0IOC1	TMP0 I/O control register 1	R/W	×	×			00H
FFFFF604H	TP0IOC2	TMP0 I/O control register 2	R/W	×	×			00H
FFFFF605H	TP0OPT0	TMP0 option register	R/W	×	×			00H
FFFFF606H	TP0CCR0	TMP0 capture/compare register 0	R/W			×		0000H
FFFFF608H	TP0CCR1	TMP0 capture/compare register 1	R/W			×		0000H
FFFFF60AH	TP0CNT	TMP0 count register	R			×		0000H
FFFFF610H	TP1CTL0	TMP1 timer control register 0	R/W	×	×			00H
FFFFF611H	TP1CTL1	TMP1 timer control register 1	R/W	×	×			00H
FFFFF612H	TP1IOC0	TMP1 I/O control register 0	R/W	×	×			00H
FFFFF613H	TP1IOC1	TMP1 I/O control register 1	R/W	×	×			00H
FFFFF614H	TP1IOC2	TMP1 I/O control register 2	R/W	×	×			00H
FFFFF615H	TP1OPT0	TMP1 option register	R/W	×	×			00H
FFFFF616H	TP1CCR0	TMP1 capture/compare register 0	R/W			×		0000H
FFFFF618H	TP1CCR1	TMP1 capture/compare register 1	R/W			×		0000H
FFFFF61AH	TP1CNT	TMP1 count register	R			×		0000H

Table 3-5: Peripheral I/O Registers (10/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFF620H	TP2CTL0	TMP2 timer control register 0	R/W	×	×			00H
FFFFF621H	TP2CTL1	TMP2 timer control register 1	R/W	×	×			00H
FFFFF622H	TP2IOC0	TMP2 I/O control register 0	R/W	×	×			00H
FFFFF623H	TP2IOC1	TMP2 I/O control register 1	R/W	×	×			00H
FFFFF624H	TP2IOC2	TMP2 I/O control register 2	R/W	×	×			00H
FFFFF625H	TP2OPT0	TMP2 option register	R/W	×	×			00H
FFFFF626H	TP2CCR0	TMP2 capture/compare register 0	R/W			×		0000H
FFFFF628H	TP2CCR1	TMP2 capture/compare register 1	R/W			×		0000H
FFFFF62AH	TP2CNT	TMP2 count register	R			×		0000H
FFFFF630H	TP3CTL0	TMP3 timer control register 0	R/W	×	×			00H
FFFFF631H	TP3CTL1	TMP3 timer control register 1	R/W	×	×			00H
FFFFF632H	TP3IOC0	TMP3 I/O control register 0	R/W	×	×			00H
FFFFF633H	TP3IOC1	TMP3 I/O control register 1	R/W	×	×			00H
FFFFF634H	TP3IOC2	TMP3 I/O control register 2	R/W	×	×			00H
FFFFF635H	TP3OPT0	TMP3 option register	R/W	×	×			00H
FFFFF636H	TP3CCR0	TMP3 capture/compare register 0	R/W			×		0000H
FFFFF638H	TP3CCR1	TMP3 capture/compare register 1	R/W			×		0000H
FFFFF63AH	TP3CNT	TMP3 count register	R			×		0000H
FFFFF640H	TP4CTL0	TMP4 timer control register 0	R/W	×	×			00H
FFFFF641H	TP4CTL1	TMP4 timer control register 1	R/W	×	×			00H
FFFFF642H	TP4IOC0	TMP4 I/O control register 0	R/W	×	×			00H
FFFFF643H	TP4IOC1	TMP4 I/O control register 1	R/W	×	×			00H
FFFFF644H	TP4IOC2	TMP4 I/O control register 2	R/W	×	×			00H
FFFFF645H	TP4OPT0	TMP4 option register	R/W	×	×			00H
FFFFF646H	TP4CCR0	TMP4 capture/compare register 0	R/W			×		0000H
FFFFF648H	TP4CCR1	TMP4 capture/compare register 1	R/W			×		0000H
FFFFF64AH	TP4CNT	TMP4 count register	R			×		0000H
FFFFF650H	TP5CTL0	TMP5 timer control register 0	R/W	×	×			00H
FFFFF651H	TP5CTL1	TMP5 timer control register 1	R/W	×	×			00H
FFFFF652H	TP5IOC0	TMP5 I/O control register 0	R/W	×	×			00H
FFFFF653H	TP5IOC1	TMP5 I/O control register 1	R/W	×	×			00H
FFFFF654H	TP5IOC2	TMP5 I/O control register 2	R/W	×	×			00H
FFFFF655H	TP5OPT0	TMP5 option register	R/W	×	×			00H
FFFFF656H	TP5CCR0	TMP5 capture/compare register 0	R/W			×		0000H
FFFFF658H	TP5CCR1	TMP5 capture/compare register 1	R/W			×		0000H
FFFFF65AH	TP5CNT	TMP5 count register	R			×		0000H
FFFFF660H	TP6CTL0	TMP6 timer control register 0	R/W	×	×			00H
FFFFF661H	TP6CTL1	TMP6 timer control register 1	R/W	×	×			00H
FFFFF662H	TP6IOC0	TMP6 I/O control register 0	R/W	×	×			00H
FFFFF663H	TP6IOC1	TMP6 I/O control register 1	R/W	×	×			00H

Table 3-5: Peripheral I/O Registers (11/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFF664H	TP6IOC2	TMP6 I/O control register 2	R/W	×	×			00H
FFFFF665H	TP6OPT0	TMP6 option register	R/W	×	×			00H
FFFFF666H	TP6CCR0	TMP6 capture/compare register 0	R/W			×		0000H
FFFFF668H	TP6CCR1	TMP6 capture/compare register 1	R/W			×		0000H
FFFFF66AH	TP6CNT	TMP6 count register	R			×		0000H
FFFFF670H	TP7CTL0	TMP7 timer control register 0	R/W	×	×			00H
FFFFF671H	TP7CTL1	TMP7 timer control register 1	R/W	×	×			00H
FFFFF672H	TP7IOC0	TMP7 I/O control register 0	R/W	×	×			00H
FFFFF673H	TP7IOC1	TMP7 I/O control register 1	R/W	×	×			00H
FFFFF674H	TP7IOC2	TMP7 I/O control register 2	R/W	×	×			00H
FFFFF675H	TP7OPT0	TMP7 option register	R/W	×	×			00H
FFFFF676H	TP7CCR0	TMP7 capture/compare register 0	R/W			×		0000H
FFFFF678H	TP7CCR1	TMP7 capture/compare register 1	R/W			×		0000H
FFFFF67AH	TP7CNT	TMP7 count register	R			×		0000H
FFFFF680H	TP8CTL0	TMP8 timer control register 0	R/W	×	×			00H
FFFFF681H	TP8CTL1	TMP8 timer control register 1	R/W	×	×			00H
FFFFF682H	TP8IOC0	TMP8 I/O control register 0	R/W	×	×			00H
FFFFF683H	TP8IOC1	TMP8 I/O control register 1	R/W	×	×			00H
FFFFF684H	TP8IOC2	TMP8 I/O control register 2	R/W	×	×			00H
FFFFF685H	TP8OPT0	TMP8 option register	R/W	×	×			00H
FFFFF686H	TP8CCR0	TMP8 capture/compare register 0	R/W			×		0000H
FFFFF688H	TP8CCR1	TMP8 capture/compare register 1	R/W			×		0000H
FFFFF68AH	TP8CNT	TMP8 count register	R			×		0000H
FFFFF690H	TT0CTL0	TMT0 timer control register 0	R/W	×	×			00H
FFFFF691H	TT0CTL1	TMT0 timer control register 1	R/W	×	×			00H
FFFFF692H	TT0CTL2	TMT0 timer control register 2	R/W	×	×			00H
FFFFF693H	TT0IOC0	TMT0 I/O control register 0	R/W	×	×			00H
FFFFF694H	TT0IOC1	TMT0 I/O control register 1	R/W	×	×			00H
FFFFF695H	TT0IOC2	TMT0 I/O control register 2	R/W	×	×			00H
FFFFF696H	TT0IOC3	TMT0 I/O control register 3	R/W	×	×			00H
FFFFF697H	TT0OPT0	TMT0 option register 0	R/W	×	×			00H
FFFFF698H	TT0OPT1	TMT0 option register 1	R/W	×	×			00H
FFFFF699H	TT0OPT2	TMT0 option register 2	R/W	×	×			00H
FFFFF69AH	TT0CCR0	TMT0 capture/compare register 0	R/W			×		0000H
FFFFF69CH	TT0CCR1	TMT0 capture/compare register 1	R/W			×		0000H
FFFFF69EH	TT0CNT	TMT0 counter read register	R			×		0000H
FFFFF6A0H	TT1CTL0	TMT1 timer control register 0	R/W	×	×			00H
FFFFF6A1H	TT1CTL1	TMT1 timer control register 1	R/W	×	×			00H
FFFFF6A2H	TT1CTL2	TMT1 timer control register 2	R/W	×	×			00H
FFFFF6A3H	TT1IOC0	TMT1 I/O control register 0	R/W	×	×			00H

Table 3-5: Peripheral I/O Registers (12/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFF6A4H	TT1IOC1	TMT1 I/O control register 1	R/W	×	×			00H
FFFFF6A5H	TT1IOC2	TMT1 I/O control register 2	R/W	×	×			00H
FFFFF6A6H	TT1IOC3	TMT1 I/O control register 3	R/W	×	×			00H
FFFFF6A7H	TT1OPT0	TMT1 option register 0	R/W	×	×			00H
FFFFF6A8H	TT1OPT1	TMT1 option register 1	R/W	×	×			00H
FFFFF6A9H	TT1OPT2	TMT1 option register 2	R/W	×	×			00H
FFFFF6AAH	TT1CCR0	TMT1 capture/compare register 0	R/W			×		0000H
FFFFF6ACH	TT1CCR1	TMT1 capture/compare register 1	R/W			×		0000H
FFFFF6AEH	TT1CNT	TMT1 counter read register	R			×		0000H
FFFFF6B0H	TMENC10	Timer ENC10 count register	R/W			×		0000H
FFFFF6B2H	CM100	Compare register 100	R/W			×		0000H
FFFFF6B4H	CM101	Compare register 101	R/W			×		0000H
FFFFF6B6H	CC100	Capture/Compare register 100	R/W			×		0000H
FFFFF6B8H	CC101	Capture/Compare register 101	R/W			×		0000H
FFFFF6BAH	CCR10	Capture/Compare control register 10	R/W	×	×			00H
FFFFF6BBH	TUM10	Timer unit mode register 10	R/W	×	×			00H
FFFFF6BCH	TMC10	Timer control register 10	R/W	×	×			00H
FFFFF6BDH	SESA10	Signal edge selection register 10	R/W	×	×			00H
FFFFF6BEH	PRM10	Prescaler mode register 10	R/W	×	×			07H
FFFFF6BFH	STATUS10	Status register 10	R	×	×			00H
FFFFF6F0H	TPIC0	TMP input source control register 0	R/W	×	×			00H
FFFFF6F2H	TPIC1	TMP input source control register 1	R/W	×	×			00H
FFFFF6F4H	TPIC2	TMP input source control register 2	R/W	×	×			00H
FFFFF700H	RNG	Random number register	R			×		undefined
FFFFF7A0H	NRC	Noise removal time control register	R/W	×	×			00H
FFFFF802H	PHS	Peripheral status register	R/W	×	×			00H
FFFFF880H	INTM0	Interrupt mode register 0	R/W	×	×			00H
FFFFF882H	INTM1	Interrupt mode register 1	R/W	×	×			00H
FFFFF884H	INTM2	Interrupt mode register 2	R/W	×	×			00H
FFFFF886H	INTM3	Interrupt mode register 3	R/W	×	×			00H
FFFFF888H	PESC5	Port emergency shut off control register 5	R/W	×	×			00H
FFFFF88AH	ESOST5	Port emergency shut off status register 5	R/W	×	×			00H
FFFFF88CH	PESC6	Port emergency shut off control register 6	R/W	×	×			00H
FFFFF88EH	ESOST6	Port emergency shut off status register 6	R/W	×	×			00H
FFFFF990H	TT0TCW	Timer T0 counter write buffer register	R/W			×		0000H
FFFFF9A0H	TT1TCW	Timer T1 counter write buffer register	R/W			×		0000H
FFFFFA00H	UC0CTL0	UARTC0 control register 0	R/W	×	×			10H
FFFFFA01H	UC0CTL1	UARTC0 control register 1	R/W		×			00H
FFFFFA02H	UC0CTL2	UARTC0 control register 2	R/W		×			00H
FFFFFA03H	UC0OPT0	UARTC0 option control register 0	R/W	×	×			14H

Table 3-5: Peripheral I/O Registers (13/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFFA04H	UC0STR	UARTC0 status register	R/W	×	×			00H
FFFFFA06H	UC0RX	UARTC0 receive data register	R			×		01FFH
FFFFFA06H	UC0RXL	UARTC0 receive data register L	R		×			FFH
FFFFFA08H	UC0TX	UARTC0 transmit data register	R/W			×		01FFH
FFFFFA08H	UC0TXL	UARTC0 transmit data register L	R/W		×			FFH
FFFFFA0AH	UC0OPT1	UARTC0 option control register 1	R/W	×	×			00H
FFFFFA0BH	UC0STR1	UARTC0 status register 1	R	×	×			00H
FFFFFA20H	UC1CTL0	UARTC1 control register 0	R/W	×	×			10H
FFFFFA21H	UC1CTL1	UARTC1 control register 1	R/W		×			00H
FFFFFA22H	UC1CTL2	UARTC1 control register 2	R/W		×			00H
FFFFFA23H	UC1OPT0	UARTC1 option control register 0	R/W	×	×			14H
FFFFFA24H	UC1STR	UARTC1 status register	R/W	×	×			00H
FFFFFA26H	UC1RX	UARTC1 receive data register	R			×		01FFH
FFFFFA26H	UC1RXL	UARTC1 receive data register L	R		×			FFH
FFFFFA28H	UC1TX	UARTC1 transmit data register	R/W			×		01FFH
FFFFFA28H	UC1TXL	UARTC1 transmit data register L	R/W		×			FFH
FFFFFA2AH	UC1OPT1	UARTC1 option control register 1	R/W	×	×			00H
FFFFFA2BH	UC1STR1	UARTC1 status register 1	R	×	×			00H
FFFFFD00H	CB0CTL0	CSIB0 control register 0	R/W	×	×			01H
FFFFFD01H	CB0CTL1	CSIB0 control register 1	R/W	×	×			00H
FFFFFD02H	CB0CTL2	CSIB0 control register 2	R/W		×			00H
FFFFFD03H	CB0STR	CSIB0 state register	R/W	×	×			00H
FFFFFD04H	CB0RX0	CSIB0 receive data register	R			×		0000H
FFFFFD04H	CB0RX0L	CSIB0 receive data register L	R		×			00H
FFFFFD06H	CB0TX0L	CSIB0 transmit data register L	R/W		×			00H
FFFFFD06H	CB0TX0	CSIB0 transmit data register	R/W			×		0000H
FFFFFD20H	CB1CTL0	CSIB1 control register 0	R/W	×	×			01H
FFFFFD21H	CB1CTL1	CSIB1 control register 1	R/W	×	×			00H
FFFFFD22H	CB1CTL2	CSIB1 control register 2	R/W		×			00H
FFFFFD23H	CB1STR	CSIB1 state register	R/W	×	×			00H
FFFFFD24H	CB1RX0	CSIB1 receive data register	R			×		0000H
FFFFFD24H	CB1RX0L	CSIB1 receive data register L	R		×			00H
FFFFFD26H	CB1TX0L	CSIB1 transmit data register L	R/W		×			00H
FFFFFD26H	CB1TX0	CSIB1 transmit data register	R/W			×		0000H
FFFFFD40H	CSIM30	CSI30 operation mode register	R/W	×	×			00H
FFFFFD41H	CSIC30	CSI30 clock selection register	R/W	×	×			07H
FFFFFD42H	SIRB30	CSI30 receive data buffer register	R			×		0000H
FFFFFD42H	SIRB30L	CSI30 receive data buffer register L	R		×			00H
FFFFFD43H	SIRB30H	CSI30 receive data buffer register H	R		×			00H
FFFFFD44H	SFCS30L	CSI30 chip selection CSI buffer register L	R/W	×	×			FFH

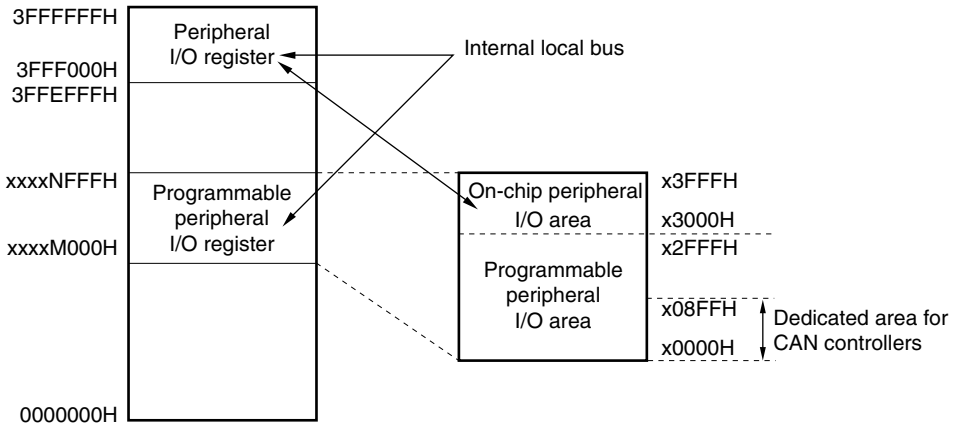
Table 3-5: Peripheral I/O Registers (14/14)

Address	Symbol	Function Register Name	R/W	Bit Units for Manipulation				Reset
				1	8	16	32	
FFFFFD44H	SFCS30	CSI30 chip selection CSI buffer register	R/W			×		FFFFH
FFFFFD45H	SFCS30H	CSI30 chip selection CSI buffer register H	R	×	×			FFH
FFFFFD46H	SFDB30L	CSI30 transmit data CSI buffer register L	R/W		×			00H
FFFFFD46H	SFDB30	CSI30 transmit data CSI buffer register	R/W			×		0000H
FFFFFD47H	SFDB30H	CSI30 transmit data CSI buffer register H	R/W		×			00H
FFFFFD48H	SFA30	CSI30 SIBUF state register	R/W	×	×			20H
FFFFFD49H	CSIL30	CSI30 transfer data length select register	R/W	×	×			00H
FFFFFD4CH	SFN30	CSI30 transfer data number specification register	R/W	×	×			00H
FFFFFD60H	CSIM31	CSI31 operation mode register	R/W	×	×			00H
FFFFFD61H	CSIC31	CSI31 clock selection register	R/W	×	×			07H
FFFFFD62H	SIRB31	CSI31 receive data buffer register	R			×		0000H
FFFFFD62H	SIRB31L	CSI31 receive data buffer register L	R		×			00H
FFFFFD63H	SIRB31H	CSI31 receive data buffer register H	R		×			00H
FFFFFD64H	SFCS31L	CSI31 chip selection CSI buffer register L	R/W	×	×			FFH
FFFFFD64H	SFCS31	CSI31 chip selection CSI buffer register	R/W			×		FFFFH
FFFFFD65H	SFCS31H	CSI31 chip selection CSI buffer register H	R	×	×			FFH
FFFFFD66H	SFDB31L	CSI31 transmit data CSI buffer register L	R/W		×			00H
FFFFFD66H	SFDB31	CSI31 transmit data CSI buffer register	R/W			×		0000H
FFFFFD67H	SFDB31H	CSI31 transmit data CSI buffer register H	R/W		×			00H
FFFFFD68H	SFA31	CSI31 SIBUF state register	R/W	×	×			20H
FFFFFD69H	CSIL31	CSI31 transfer data length select register	R/W	×	×			00H
FFFFFD6CH	SFN31	CSI31 transfer data number specification register	R/W	×	×			00H
FFFFFDC0H	PRSM0	Prescaler mode register 0	R/W	×	×			00H
FFFFFDC1H	PRSCM0	Prescaler compare register 0	R/W	×	×			00H
FFFFFDD0H	PRSM1	Prescaler mode register 1	R/W	×	×			00H
FFFFFDD1H	PRSCM1	Prescaler compare register 1	R/W	×	×			00H
FFFFFDE0H	PRSM2	Prescaler mode register 2	R/W	×	×			00H
FFFFFDE1H	PRSCM2	Prescaler compare register 2	R/W	×	×			00H
FFFFFE00H	DMAWC0	DMA wait control register 0	R/W	×	×			37H
FFFFFE02H	DMAWC1	DMA wait control register 1	R/W	×	×			07H

3.4.7 Programmable peripheral I/O area

In the V850E/PH2, the 16 KB area of x0000H to x3FFFH is provided as a programmable peripheral I/O area. In this area, the area between x0000H and x08FFH is used exclusively for the CAN controllers (CAN0, CAN1).
The internal bus of the V850E/PH2 becomes active when the on-chip peripheral I/O register area (FFFF000H to FFFFFFFH) or the programmable peripheral I/O register area (xxxxm000H to xxxxnFFFH) is accessed (m = xx00B, n= xx11B). However, the on-chip peripheral I/O area is allocated to the last 4 KB of the programmable peripheral I/O register area. Note that when data is written to this area, the written contents are reflected on the on-chip peripheral I/O area. Therefore, access to this area is prohibited. To access the on-chip peripheral I/O area, be sure to specify addresses FFFF000H to FFFFFFFH.

Figure 3-19: Programmable Peripheral I/O Area (Outline)



Remark: M = xx00B, N = M + 11B, P= M + 10B

Cautions: 1. It is recommended to locate the programmable peripheral area in the first 32 Mbyte of the physical memory.

2. The programmable peripheral area is not allowed to overlap the ROM or RAM areas: BPC must be initialized with a value in the range 0040H to 0FFBH.

(1) Peripheral area selection control register (BPC)

The peripheral area selection control register (BPC) is used to select a programmable peripheral I/O register area where the registers of the CAN controller are allocated.
This register can be read/written in 16-bit units.

Figure 3-20: Programmable Peripheral Area Control Register BPC

After reset: 0000H R/W Address: FFFFF064H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BPC	PA15	0	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

(n = 0, 1)

PA15	Usage of Programmable Peripheral I/O Area
0	Disables usage of programmable peripheral I/O area
1	Enables usage of programmable peripheral I/O area

PA13 to PA0	Base address of Programmable Peripheral I/O Area
0000H to 3FFFH	Specifies the base address of the programmable peripheral I/O area (PA13 to PA0 corresponds to A27 to A14, respectively).

Remark: The recommended value of the BPC register to enable the programmable peripheral I/O area is 87FFH. This setting assigns the programmable peripheral I/O area to addresses from 1FFC000H to 1FFFFFFH.

(2) Registers in the programmable peripheral I/O area

In the following Table 3-6 the addresses shown are offsets in the programmable peripheral I/O area, which have to be added to base address set by the BPC register.

Table 3-6: Programmable Peripheral I/O Registers (1/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
0000000H	C0GMCTRL	CAN0 global macro control register	R/W			×	0000H
0000000H	C0GMCTRLL	CAN0 global macro control register L	R/W	×	×		00H
0000001H	C0GMCTRLH	CAN0 global macro control register H	R/W	×	×		00H
0000002H	C0GMCS	CAN0 global macro clock selection register	R/W	×	×		00H
0000006H	C0GMABT	CAN0 global macro automatic block transmission register	R/W			×	0000H
0000006H	C0GMABTL	CAN0 global macro automatic block transmission register L	R/W	×	×		00H
0000007H	C0GMABTH	CAN0 global macro automatic block transmission register H	R/W	×	×		00H
0000008H	C0GMABTD	CAN0 global macro automatic block transmission delay register	R/W	×	×		00H
0000040H	C0MASK1L	CAN0 module mask 1 register L	R/W			×	Undefined
0000042H	C0MASK1H	CAN0 module mask 1 register H	R/W			×	Undefined
0000044H	C0MASK2L	CAN0 module mask 2 register L	R/W			×	Undefined
0000046H	C0MASK2H	CAN0 module mask 2 register H	R/W			×	Undefined
0000048H	C0MASK3L	CAN0 module mask 3 register L	R/W			×	Undefined
000004AH	C0MASK3H	CAN0 module mask 3 register H	R/W			×	Undefined
000004CH	C0MASK4L	CAN0 module mask 4 register L	R/W			×	Undefined
000004EH	C0MASK4H	CAN0 module mask 4 register H	R/W			×	Undefined
0000050H	C0CTRL	CAN0 module control register	R/W			×	0000H
0000052H	C0LEC	CAN0 module last error code register	R/W	×	×		00H
0000053H	C0INFO	CAN0 module information register	R	×	×		00H
0000054H	C0ERC	CAN0 module error counter	R/W			×	0000H
0000056H	C0IE	CAN0 module interrupt enable register	R/W			×	0000H
0000056H	C0IEL	CAN0 module interrupt enable register L	R/W	×	×		00H
0000057H	C0IEH	CAN0 module interrupt enable register H	R/W	×	×		00H
0000058H	C0INTS	CAN0 module interrupt status register	R/W			×	0000H
0000058H	C0INTSL	CAN0 module interrupt status register L	R/W	×	×		00H
000005AH	C0BRP	CAN0 module bit-rate prescaler register	R/W	×	×		FFH
000005CH	C0BTR	CAN0 bit-rate register	R/W			×	370FH
000005EH	C0LIPT	CAN0 module last in-pointer register	R/W		×		Undefined
0000060H	C0RGPT	CAN0 module receive history list get pointer register	R/W			×	Undefined
0000060H	C0RGPTL	CAN0 module receive history list get pointer register L	R/W	×	×		01H
0000062H	C0LOPT	CAN0 module last out-pointer register	R		×		Undefined

Table 3-6: Programmable Peripheral I/O Registers (2/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
0000064H	C0TGPT	CAN0 module transmit history list get pointer register	R/W			×	Undefined
0000064H	C0TGPTL	CAN0 module transmit history list get pointer register L	R/W	×	×		01H
0000066H	C0TS	CAN0 module time stamp register	R/W			×	0000H
0000066H	C0TSL	CAN0 module time stamp register L	R/W	×	×		00H
0000067H	C0TSH	CAN0 module time stamp register H	R/W	×	×		00H
0000100H	C0MDATA0100	CAN0 message data byte 0 and 1 register 00	R/W			×	Undefined
0000100H	C0MDATA000	CAN0 message data byte 0 register 00	R/W	×	×		Undefined
0000101H	C0MDATA100	CAN0 message data byte 1 register 00	R/W	×	×		Undefined
0000102H	C0MDATA2300	CAN0 message data byte 2 and 3 register 00	R/W			×	Undefined
0000102H	C0MDATA200	CAN0 message data byte 2 register 00	R/W	×	×		Undefined
0000103H	C0MDATA300	CAN0 message data byte 3 register 00	R/W	×	×		Undefined
0000104H	C0MDATA4500	CAN0 message data byte 4 and 5 register 00	R/W			×	Undefined
0000104H	C0MDATA400	CAN0 message data byte 2 register 00	R/W	×	×		Undefined
0000105H	C0MDATA500	CAN0 message data byte 3 register 00	R/W	×	×		Undefined
0000106H	C0MDATA6700	CAN0 message data byte 6 and 7 register 00	R/W			×	Undefined
0000106H	C0MDATA600	CAN0 message data byte 6 register 00	R/W	×	×		Undefined
0000107H	C0MDATA700	CAN0 message data byte 7 register 00	R/W	×	×		Undefined
0000108H	C0MDLC00	CAN0 message data length code register 00	R/W	×	×		Undefined
0000109H	C0MCONF00	CAN0 message configuration register 00	R/W	×	×		Undefined
000010AH	C0MIDL00	CAN0 message identifier L register 00	R/W			×	Undefined
000010CH	C0MIDH00	CAN0 message identifier H register 00	R/W			×	Undefined
000010EH	C0MCTRL00	CAN0 message control register 00	R/W			×	Undefined
0000120H	C0MDATA0101	CAN0 message data byte 0 and 1 register 01	R/W			×	Undefined
0000120H	C0MDATA001	CAN0 message data byte 0 register 01	R/W	×	×		Undefined
0000121H	C0MDATA101	CAN0 message data byte 1 register 01	R/W	×	×		Undefined
0000122H	C0MDATA2301	CAN0 message data byte 2 and 3 register 01	R/W			×	Undefined
0000122H	C0MDATA201	CAN0 message data byte 2 register 01	R/W	×	×		Undefined
0000123H	C0MDATA301	CAN0 message data byte 3 register 01	R/W	×	×		Undefined
0000124H	C0MDATA4501	CAN0 message data byte 4 and 5 register 01	R/W			×	Undefined
0000124H	C0MDATA401	CAN0 message data byte 2 register 01	R/W	×	×		Undefined
0000125H	C0MDATA501	CAN0 message data byte 3 register 01	R/W	×	×		Undefined
0000126H	C0MDATA6701	CAN0 message data byte 6 and 7 register 01	R/W			×	Undefined
0000126H	C0MDATA601	CAN0 message data byte 6 register 01	R/W	×	×		Undefined
0000127H	C0MDATA701	CAN0 message data byte 7 register 01	R/W	×	×		Undefined
0000128H	C0MDLC01	CAN0 message data length code register 01	R/W	×	×		Undefined
0000129H	C0MCONF01	CAN0 message configuration register 01	R/W	×	×		Undefined
000012AH	C0MIDL01	CAN0 message identifier L register 01	R/W			×	Undefined
000012CH	C0MIDH01	CAN0 message identifier H register 01	R/W			×	Undefined
000012EH	C0MCTRL01	CAN0 message control register 01	R/W			×	Undefined

Table 3-6: Programmable Peripheral I/O Registers (3/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
0000140H	C0MDATA0102	CAN0 message data byte 0 and 1 register 02	R/W			×	Undefined
0000140H	C0MDATA002	CAN0 message data byte 0 register 02	R/W	×	×		Undefined
0000141H	C0MDATA102	CAN0 message data byte 1 register 02	R/W	×	×		Undefined
0000142H	C0MDATA2302	CAN0 message data byte 2 and 3 register 02	R/W			×	Undefined
0000142H	C0MDATA202	CAN0 message data byte 2 register 02	R/W	×	×		Undefined
0000143H	C0MDATA302	CAN0 message data byte 3 register 02	R/W	×	×		Undefined
0000144H	C0MDATA4502	CAN0 message data byte 4 and 5 register 02	R/W			×	Undefined
0000144H	C0MDATA402	CAN0 message data byte 2 register 02	R/W	×	×		Undefined
0000145H	C0MDATA502	CAN0 message data byte 3 register 02	R/W	×	×		Undefined
0000146H	C0MDATA6702	CAN0 message data byte 6 and 7 register 02	R/W			×	Undefined
0000146H	C0MDATA602	CAN0 message data byte 6 register 02	R/W	×	×		Undefined
0000147H	C0MDATA702	CAN0 message data byte 7 register 02	R/W	×	×		Undefined
0000148H	C0MDLC02	CAN0 message data length code register 02	R/W	×	×		Undefined
0000149H	C0MCONF02	CAN0 message configuration register 02	R/W	×	×		Undefined
000014AH	C0MIDL02	CAN0 message identifier L register 02	R/W			×	Undefined
000014CH	C0MIDH02	CAN0 message identifier H register 02	R/W			×	Undefined
000014EH	C0MCTRL02	CAN0 message control register 02	R/W			×	Undefined
0000160H	C0MDATA0103	CAN0 message data byte 0 and 1 register 03	R/W			×	Undefined
0000160H	C0MDATA003	CAN0 message data byte 0 register 03	R/W	×	×		Undefined
0000161H	C0MDATA103	CAN0 message data byte 1 register 03	R/W	×	×		Undefined
0000162H	C0MDATA2303	CAN0 message data byte 2 and 3 register 03	R/W			×	Undefined
0000162H	C0MDATA203	CAN0 message data byte 2 register 03	R/W	×	×		Undefined
0000163H	C0MDATA303	CAN0 message data byte 3 register 03	R/W	×	×		Undefined
0000164H	C0MDATA4503	CAN0 message data byte 4 and 5 register 03	R/W			×	Undefined
0000164H	C0MDATA403	CAN0 message data byte 2 register 03	R/W	×	×		Undefined
0000165H	C0MDATA503	CAN0 message data byte 3 register 03	R/W	×	×		Undefined
0000166H	C0MDATA6703	CAN0 message data byte 6 and 7 register 03	R/W			×	Undefined
0000166H	C0MDATA603	CAN0 message data byte 6 register 03	R/W	×	×		Undefined
0000167H	C0MDATA703	CAN0 message data byte 7 register 03	R/W	×	×		Undefined
0000168H	C0MDLC03	CAN0 message data length code register 03	R/W	×	×		Undefined
0000169H	C0MCONF03	CAN0 message configuration register 03	R/W	×	×		Undefined
000016AH	C0MIDL03	CAN0 message identifier L register 03	R/W			×	Undefined
000016CH	C0MIDH03	CAN0 message identifier H register 03	R/W			×	Undefined
000016EH	C0MCTRL03	CAN0 message control register 03	R/W			×	Undefined
0000180H	C0MDATA0104	CAN0 message data byte 0 and 1 register 04	R/W			×	Undefined
0000180H	C0MDATA004	CAN0 message data byte 0 register 04	R/W	×	×		Undefined
0000181H	C0MDATA104	CAN0 message data byte 1 register 04	R/W	×	×		Undefined
0000182H	C0MDATA2304	CAN0 message data byte 2 and 3 register 04	R/W			×	Undefined
0000182H	C0MDATA204	CAN0 message data byte 2 register 04	R/W	×	×		Undefined
0000183H	C0MDATA304	CAN0 message data byte 3 register 04	R/W	×	×		Undefined

Table 3-6: Programmable Peripheral I/O Registers (4/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
0000184H	C0MDATA4504	CAN0 message data byte 4 and 5 register 04	R/W			×	Undefined
0000184H	C0MDATA404	CAN0 message data byte 2 register 04	R/W	×	×		Undefined
0000185H	C0MDATA504	CAN0 message data byte 3 register 04	R/W	×	×		Undefined
0000186H	C0MDATA6704	CAN0 message data byte 6 and 7 register 04	R/W			×	Undefined
0000186H	C0MDATA604	CAN0 message data byte 6 register 04	R/W	×	×		Undefined
0000187H	C0MDATA704	CAN0 message data byte 7 register 04	R/W	×	×		Undefined
0000188H	C0MDLC04	CAN0 message data length code register 04	R/W	×	×		Undefined
0000189H	C0MCONF04	CAN0 message configuration register 04	R/W	×	×		Undefined
000018AH	C0MIDL04	CAN0 message identifier L register 04	R/W			×	Undefined
000018CH	C0MIDH04	CAN0 message identifier H register 04	R/W			×	Undefined
000018EH	C0MCTRL04	CAN0 message control register 04	R/W			×	Undefined
00001A0H	C0MDATA0105	CAN0 message data byte 0 and 1 register 05	R/W			×	Undefined
00001A0H	C0MDATA005	CAN0 message data byte 0 register 05	R/W	×	×		Undefined
00001A1H	C0MDATA105	CAN0 message data byte 1 register 05	R/W	×	×		Undefined
00001A2H	C0MDATA2305	CAN0 message data byte 2 and 3 register 05	R/W			×	Undefined
00001A2H	C0MDATA205	CAN0 message data byte 2 register 05	R/W	×	×		Undefined
00001A3H	C0MDATA305	CAN0 message data byte 3 register 05	R/W	×	×		Undefined
00001A4H	C0MDATA4505	CAN0 message data byte 4 and 5 register 05	R/W			×	Undefined
00001A4H	C0MDATA405	CAN0 message data byte 2 register 05	R/W	×	×		Undefined
00001A5H	C0MDATA505	CAN0 message data byte 3 register 05	R/W	×	×		Undefined
00001A6H	C0MDATA6705	CAN0 message data byte 6 and 7 register 05	R/W			×	Undefined
00001A6H	C0MDATA605	CAN0 message data byte 6 register 05	R/W	×	×		Undefined
00001A7H	C0MDATA705	CAN0 message data byte 7 register 05	R/W	×	×		Undefined
00001A8H	C0MDLC05	CAN0 message data length code register 05	R/W	×	×		Undefined
00001A9H	C0MCONF05	CAN0 message configuration register 05	R/W	×	×		Undefined
00001AAH	C0MIDL05	CAN0 message identifier L register 05	R/W			×	Undefined
00001ACH	C0MIDH05	CAN0 message identifier H register 05	R/W			×	Undefined
00001AEH	C0MCTRL05	CAN0 message control register 05	R/W			×	Undefined
00001C0H	C0MDATA0106	CAN0 message data byte 0 and 1 register 06	R/W			×	Undefined
00001C0H	C0MDATA006	CAN0 message data byte 0 register 06	R/W	×	×		Undefined
00001C1H	C0MDATA106	CAN0 message data byte 1 register 06	R/W	×	×		Undefined
00001C2H	C0MDATA2306	CAN0 message data byte 2 and 3 register 06	R/W			×	Undefined
00001C2H	C0MDATA206	CAN0 message data byte 2 register 06	R/W	×	×		Undefined
00001C3H	C0MDATA306	CAN0 message data byte 3 register 06	R/W	×	×		Undefined
00001C4H	C0MDATA4506	CAN0 message data byte 4 and 5 register 06	R/W			×	Undefined
00001C4H	C0MDATA406	CAN0 message data byte 2 register 06	R/W	×	×		Undefined
00001C5H	C0MDATA506	CAN0 message data byte 3 register 06	R/W	×	×		Undefined
00001C6H	C0MDATA6706	CAN0 message data byte 6 and 7 register 06	R/W			×	Undefined
00001C6H	C0MDATA606	CAN0 message data byte 6 register 06	R/W	×	×		Undefined
00001C7H	C0MDATA706	CAN0 message data byte 7 register 06	R/W	×	×		Undefined

Table 3-6: Programmable Peripheral I/O Registers (5/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
00001C8H	C0MDLC06	CAN0 message data length code register 06	R/W	×	×		Undefined
00001C9H	C0MCONF06	CAN0 message configuration register 06	R/W	×	×		Undefined
00001CAH	C0MIDL06	CAN0 message identifier L register 06	R/W			×	Undefined
00001CCH	C0MIDH06	CAN0 message identifier H register 06	R/W			×	Undefined
00001CEH	C0MCTRL06	CAN0 message control register 06	R/W			×	Undefined
00001E0H	C0MDATA0107	CAN0 message data byte 0 and 1 register 07	R/W			×	Undefined
00001E0H	C0MDATA007	CAN0 message data byte 0 register 07	R/W	×	×		Undefined
00001E1H	C0MDATA107	CAN0 message data byte 1 register 07	R/W	×	×		Undefined
00001E2H	C0MDATA2307	CAN0 message data byte 2 and 3 register 07	R/W			×	Undefined
00001E2H	C0MDATA207	CAN0 message data byte 2 register 07	R/W	×	×		Undefined
00001E3H	C0MDATA307	CAN0 message data byte 3 register 07	R/W	×	×		Undefined
00001E4H	C0MDATA4507	CAN0 message data byte 4 and 5 register 07	R/W			×	Undefined
00001E4H	C0MDATA407	CAN0 message data byte 2 register 07	R/W	×	×		Undefined
00001E5H	C0MDATA507	CAN0 message data byte 3 register 07	R/W	×	×		Undefined
00001E6H	C0MDATA6707	CAN0 message data byte 6 and 7 register 07	R/W			×	Undefined
00001E6H	C0MDATA607	CAN0 message data byte 6 register 07	R/W	×	×		Undefined
00001E7H	C0MDATA707	CAN0 message data byte 7 register 07	R/W	×	×		Undefined
00001E8H	C0MDLC07	CAN0 message data length code register 07	R/W	×	×		Undefined
00001E9H	C0MCONF07	CAN0 message configuration register 07	R/W	×	×		Undefined
00001EAH	C0MIDL07	CAN0 message identifier L register 07	R/W			×	Undefined
00001ECH	C0MIDH07	CAN0 message identifier H register 07	R/W			×	Undefined
00001EEH	C0MCTRL07	CAN0 message control register 07	R/W			×	Undefined
0000200H	C0MDATA0108	CAN0 message data byte 0 and 1 register 08	R/W			×	Undefined
0000200H	C0MDATA008	CAN0 message data byte 0 register 08	R/W	×	×		Undefined
0000201H	C0MDATA108	CAN0 message data byte 1 register 08	R/W	×	×		Undefined
0000202H	C0MDATA2308	CAN0 message data byte 2 and 3 register 08	R/W			×	Undefined
0000202H	C0MDATA208	CAN0 message data byte 2 register 08	R/W	×	×		Undefined
0000203H	C0MDATA308	CAN0 message data byte 3 register 08	R/W	×	×		Undefined
0000204H	C0MDATA4508	CAN0 message data byte 4 and 5 register 08	R/W			×	Undefined
0000204H	C0MDATA408	CAN0 message data byte 2 register 08	R/W	×	×		Undefined
0000205H	C0MDATA508	CAN0 message data byte 3 register 08	R/W	×	×		Undefined
0000206H	C0MDATA6708	CAN0 message data byte 6 and 7 register 08	R/W			×	Undefined
0000206H	C0MDATA608	CAN0 message data byte 6 register 08	R/W	×	×		Undefined
0000207H	C0MDATA708	CAN0 message data byte 7 register 08	R/W	×	×		Undefined
0000208H	C0MDLC08	CAN0 message data length code register 08	R/W	×	×		Undefined
0000209H	C0MCONF08	CAN0 message configuration register 08	R/W	×	×		Undefined
000020AH	C0MIDL08	CAN0 message identifier L register 08	R/W			×	Undefined
000020CH	C0MIDH08	CAN0 message identifier H register 08	R/W			×	Undefined
000020EH	C0MCTRL08	CAN0 message control register 08	R/W			×	Undefined

Table 3-6: Programmable Peripheral I/O Registers (6/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
0000220H	C0MDATA0109	CAN0 message data byte 0 and 1 register 09	R/W			×	Undefined
0000220H	C0MDATA009	CAN0 message data byte 0 register 09	R/W	×	×		Undefined
0000221H	C0MDATA109	CAN0 message data byte 1 register 09	R/W	×	×		Undefined
0000222H	C0MDATA2309	CAN0 message data byte 2 and 3 register 09	R/W			×	Undefined
0000222H	C0MDATA209	CAN0 message data byte 2 register 09	R/W	×	×		Undefined
0000223H	C0MDATA309	CAN0 message data byte 3 register 09	R/W	×	×		Undefined
0000224H	C0MDATA4509	CAN0 message data byte 4 and 5 register 09	R/W			×	Undefined
0000224H	C0MDATA409	CAN0 message data byte 2 register 09	R/W	×	×		Undefined
0000225H	C0MDATA509	CAN0 message data byte 3 register 09	R/W	×	×		Undefined
0000226H	C0MDATA6709	CAN0 message data byte 6 and 7 register 09	R/W			×	Undefined
0000226H	C0MDATA609	CAN0 message data byte 6 register 09	R/W	×	×		Undefined
0000227H	C0MDATA709	CAN0 message data byte 7 register 09	R/W	×	×		Undefined
0000228H	C0MDLC09	CAN0 message data length code register 09	R/W	×	×		Undefined
0000229H	C0MCONF09	CAN0 message configuration register 09	R/W	×	×		Undefined
000022AH	C0MIDL09	CAN0 message identifier L register 09	R/W			×	Undefined
000022CH	C0MIDH09	CAN0 message identifier H register 09	R/W			×	Undefined
000022EH	C0MCTRL09	CAN0 message control register 09	R/W			×	Undefined
0000240H	C0MDATA0110	CAN0 message data byte 0 and 1 register 10	R/W			×	Undefined
0000240H	C0MDATA010	CAN0 message data byte 0 register 10	R/W	×	×		Undefined
0000241H	C0MDATA110	CAN0 message data byte 1 register 10	R/W	×	×		Undefined
0000242H	C0MDATA2310	CAN0 message data byte 2 and 3 register 10	R/W			×	Undefined
0000242H	C0MDATA210	CAN0 message data byte 2 register 10	R/W	×	×		Undefined
0000243H	C0MDATA310	CAN0 message data byte 3 register 10	R/W	×	×		Undefined
0000244H	C0MDATA4510	CAN0 message data byte 4 and 5 register 10	R/W			×	Undefined
0000244H	C0MDATA410	CAN0 message data byte 2 register 10	R/W	×	×		Undefined
0000245H	C0MDATA510	CAN0 message data byte 3 register 10	R/W	×	×		Undefined
0000246H	C0MDATA6710	CAN0 message data byte 6 and 7 register 10	R/W			×	Undefined
0000246H	C0MDATA610	CAN0 message data byte 6 register 10	R/W	×	×		Undefined
0000247H	C0MDATA710	CAN0 message data byte 7 register 10	R/W	×	×		Undefined
0000248H	C0MDLC10	CAN0 message data length code register 10	R/W	×	×		Undefined
0000249H	C0MCONF10	CAN0 message configuration register 10	R/W	×	×		Undefined
000024AH	C0MIDL10	CAN0 message identifier L register 10	R/W			×	Undefined
000024CH	C0MIDH10	CAN0 message identifier H register 10	R/W			×	Undefined
000024EH	C0MCTRL10	CAN0 message control register 10	R/W			×	Undefined
0000260H	C0MDATA0111	CAN0 message data byte 0 and 1 register 11	R/W			×	Undefined
0000260H	C0MDATA011	CAN0 message data byte 0 register 11	R/W	×	×		Undefined
0000261H	C0MDATA111	CAN0 message data byte 1 register 11	R/W	×	×		Undefined
0000262H	C0MDATA2311	CAN0 message data byte 2 and 3 register 11	R/W			×	Undefined
0000262H	C0MDATA211	CAN0 message data byte 2 register 11	R/W	×	×		Undefined
0000263H	C0MDATA311	CAN0 message data byte 3 register 11	R/W	×	×		Undefined

Table 3-6: Programmable Peripheral I/O Registers (7/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
0000264H	C0MDATA4511	CAN0 message data byte 4 and 5 register 11	R/W			×	Undefined
0000264H	C0MDATA411	CAN0 message data byte 2 register 11	R/W	×	×		Undefined
0000265H	C0MDATA511	CAN0 message data byte 3 register 11	R/W	×	×		Undefined
0000266H	C0MDATA6711	CAN0 message data byte 6 and 7 register 11	R/W			×	Undefined
0000266H	C0MDATA611	CAN0 message data byte 6 register 11	R/W	×	×		Undefined
0000267H	C0MDATA711	CAN0 message data byte 7 register 11	R/W	×	×		Undefined
0000268H	C0MDLC11	CAN0 message data length code register 11	R/W	×	×		Undefined
0000269H	C0MCONF11	CAN0 message configuration register 11	R/W	×	×		Undefined
000026AH	C0MIDL11	CAN0 message identifier L register 11	R/W			×	Undefined
000026CH	C0MIDH11	CAN0 message identifier H register 11	R/W			×	Undefined
000026EH	C0MCTRL11	CAN0 message control register 11	R/W			×	Undefined
0000280H	C0MDATA0112	CAN0 message data byte 0 and 1 register 12	R/W			×	Undefined
0000280H	C0MDATA012	CAN0 message data byte 0 register 12	R/W	×	×		Undefined
0000281H	C0MDATA112	CAN0 message data byte 1 register 12	R/W	×	×		Undefined
0000282H	C0MDATA2312	CAN0 message data byte 2 and 3 register 12	R/W			×	Undefined
0000282H	C0MDATA212	CAN0 message data byte 2 register 12	R/W	×	×		Undefined
0000283H	C0MDATA312	CAN0 message data byte 3 register 12	R/W	×	×		Undefined
0000284H	C0MDATA4512	CAN0 message data byte 4 and 5 register 12	R/W			×	Undefined
0000284H	C0MDATA412	CAN0 message data byte 2 register 12	R/W	×	×		Undefined
0000285H	C0MDATA512	CAN0 message data byte 3 register 12	R/W	×	×		Undefined
0000286H	C0MDATA6712	CAN0 message data byte 6 and 7 register 12	R/W			×	Undefined
0000286H	C0MDATA612	CAN0 message data byte 6 register 12	R/W	×	×		Undefined
0000287H	C0MDATA712	CAN0 message data byte 7 register 12	R/W	×	×		Undefined
0000288H	C0MDLC12	CAN0 message data length code register 12	R/W	×	×		Undefined
0000289H	C0MCONF12	CAN0 message configuration register 12	R/W	×	×		Undefined
000028AH	C0MIDL12	CAN0 message identifier L register 12	R/W			×	Undefined
000028CH	C0MIDH12	CAN0 message identifier H register 12	R/W			×	Undefined
000028EH	C0MCTRL12	CAN0 message control register 12	R/W			×	Undefined
00002A0H	C0MDATA0113	CAN0 message data byte 0 and 1 register 13	R/W			×	Undefined
00002A0H	C0MDATA013	CAN0 message data byte 0 register 13	R/W	×	×		Undefined
00002A1H	C0MDATA113	CAN0 message data byte 1 register 13	R/W	×	×		Undefined
00002A2H	C0MDATA2313	CAN0 message data byte 2 and 3 register 13	R/W			×	Undefined
00002A2H	C0MDATA213	CAN0 message data byte 2 register 13	R/W	×	×		Undefined
00002A3H	C0MDATA313	CAN0 message data byte 3 register 13	R/W	×	×		Undefined
00002A4H	C0MDATA4513	CAN0 message data byte 4 and 5 register 13	R/W			×	Undefined
00002A4H	C0MDATA413	CAN0 message data byte 2 register 13	R/W	×	×		Undefined
00002A5H	C0MDATA513	CAN0 message data byte 3 register 13	R/W	×	×		Undefined
00002A6H	C0MDATA6713	CAN0 message data byte 6 and 7 register 13	R/W			×	Undefined
00002A6H	C0MDATA613	CAN0 message data byte 6 register 13	R/W	×	×		Undefined
00002A7H	C0MDATA713	CAN0 message data byte 7 register 13	R/W	×	×		Undefined

Table 3-6: Programmable Peripheral I/O Registers (8/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
00002A8H	C0MDLC13	CAN0 message data length code register 13	R/W	×	×		Undefined
00002A9H	C0MCONF13	CAN0 message configuration register 13	R/W	×	×		Undefined
00002AAH	C0MIDL13	CAN0 message identifier L register 13	R/W			×	Undefined
00002ACH	C0MIDH13	CAN0 message identifier H register 13	R/W			×	Undefined
00002AEH	C0MCTRL13	CAN0 message control register 13	R/W			×	Undefined
00002C0H	C0MDATA0114	CAN0 message data byte 0 and 1 register 14	R/W			×	Undefined
00002C0H	C0MDATA014	CAN0 message data byte 0 register 14	R/W	×	×		Undefined
00002C1H	C0MDATA114	CAN0 message data byte 1 register 14	R/W	×	×		Undefined
00002C2H	C0MDATA2314	CAN0 message data byte 2 and 3 register 14	R/W			×	Undefined
00002C2H	C0MDATA214	CAN0 message data byte 2 register 14	R/W	×	×		Undefined
00002C3H	C0MDATA314	CAN0 message data byte 3 register 14	R/W	×	×		Undefined
00002C4H	C0MDATA4514	CAN0 message data byte 4 and 5 register 14	R/W			×	Undefined
00002C4H	C0MDATA414	CAN0 message data byte 2 register 14	R/W	×	×		Undefined
00002C5H	C0MDATA514	CAN0 message data byte 3 register 14	R/W	×	×		Undefined
00002C6H	C0MDATA6714	CAN0 message data byte 6 and 7 register 14	R/W			×	Undefined
00002C6H	C0MDATA614	CAN0 message data byte 6 register 14	R/W	×	×		Undefined
00002C7H	C0MDATA714	CAN0 message data byte 7 register 14	R/W	×	×		Undefined
00002C8H	C0MDLC14	CAN0 message data length code register 14	R/W	×	×		Undefined
00002C9H	C0MCONF14	CAN0 message configuration register 14	R/W	×	×		Undefined
00002CAH	C0MIDL14	CAN0 message identifier L register 14	R/W			×	Undefined
00002CCH	C0MIDH14	CAN0 message identifier H register 14	R/W			×	Undefined
00002CEH	C0MCTRL14	CAN0 message control register 14	R/W			×	Undefined
00002E0H	C0MDATA0115	CAN0 message data byte 0 and 1 register 15	R/W			×	Undefined
00002E0H	C0MDATA015	CAN0 message data byte 0 register 15	R/W	×	×		Undefined
00002E1H	C0MDATA115	CAN0 message data byte 1 register 15	R/W	×	×		Undefined
00002E2H	C0MDATA2315	CAN0 message data byte 2 and 3 register 15	R/W			×	Undefined
00002E2H	C0MDATA215	CAN0 message data byte 2 register 15	R/W	×	×		Undefined
00002E3H	C0MDATA315	CAN0 message data byte 3 register 15	R/W	×	×		Undefined
00002E4H	C0MDATA4515	CAN0 message data byte 4 and 5 register 15	R/W			×	Undefined
00002E4H	C0MDATA415	CAN0 message data byte 2 register 15	R/W	×	×		Undefined
00002E5H	C0MDATA515	CAN0 message data byte 3 register 15	R/W	×	×		Undefined
00002E6H	C0MDATA6715	CAN0 message data byte 6 and 7 register 15	R/W			×	Undefined
00002E6H	C0MDATA615	CAN0 message data byte 6 register 15	R/W	×	×		Undefined
00002E7H	C0MDATA715	CAN0 message data byte 7 register 15	R/W	×	×		Undefined
00002E8H	C0MDLC15	CAN0 message data length code register 15	R/W	×	×		Undefined
00002E9H	C0MCONF15	CAN0 message configuration register 15	R/W	×	×		Undefined
00002EAH	C0MIDL15	CAN0 message identifier L register 15	R/W			×	Undefined
00002ECH	C0MIDH15	CAN0 message identifier H register 15	R/W			×	Undefined
00002EEH	C0MCTRL15	CAN0 message control register 15	R/W			×	Undefined

Table 3-6: Programmable Peripheral I/O Registers (9/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
0000600H	C1GMCTRL	CAN1 global macro control register	R/W			×	0000H
0000600H	C1GMCTRLL	CAN1 global macro control register L	R/W	×	×		00H
0000601H	C1GMCTRLH	CAN1 global macro control register H	R/W	×	×		00H
0000602H	C1GMCS	CAN1 global macro clock selection register	R/W	×	×		00H
0000606H	C1GMABT	CAN1 global macro automatic block transmission register	R/W			×	0000H
0000606H	C1GMABTL	CAN1 global macro automatic block transmission register L	R/W	×	×		00H
0000607H	C1GMABTH	CAN1 global macro automatic block transmission register H	R/W	×	×		00H
0000608H	C1GMABTD	CAN1 global macro automatic block transmission delay register	R/W	×	×		00H
0000640H	C1MASK1L	CAN1 module mask 1 register L	R/W			×	Undefined
0000642H	C1MASK1H	CAN1 module mask 1 register H	R/W			×	Undefined
0000644H	C1MASK2L	CAN1 module mask 2 register L	R/W			×	Undefined
0000646H	C1MASK2H	CAN1 module mask 2 register H	R/W			×	Undefined
0000648H	C1MASK3L	CAN1 module mask 3 register L	R/W			×	Undefined
000064AH	C1MASK3H	CAN1 module mask 3 register H	R/W			×	Undefined
000064CH	C1MASK4L	CAN1 module mask 4 register L	R/W			×	Undefined
000064EH	C1MASK4H	CAN1 module mask 4 register H	R/W			×	Undefined
0000650H	C1CTRL	CAN1 module control register	R/W			×	0000H
0000652H	C1LEC	CAN1 module last error code register	R/W	×	×		00H
0000653H	C1INFO	CAN1 module information register	R	×	×		00H
0000654H	C1ERC	CAN1 module error counter	R/W			×	0000H
0000656H	C1IE	CAN1 module interrupt enable register	R/W			×	0000H
0000656H	C1IEL	CAN1 module interrupt enable register L	R/W	×	×		00H
0000657H	C1IEH	CAN1 module interrupt enable register H	R/W	×	×		00H
0000658H	C1INTS	CAN1 module interrupt status register	R/W			×	0000H
0000658H	C1INTSL	CAN1 module interrupt status register L	R/W	×	×		00H
000065AH	C1BRP	CAN1 module bit-rate prescaler register	R/W	×	×		FFH
000065CH	C1BTR	CAN1 bit-rate register	R/W			×	370FH
000065EH	C1LIPT	CAN1 module last in-pointer register	R/W		×		Undefined
0000660H	C1RGPT	CAN1 module receive history list get pointer register	R/W			×	Undefined
0000660H	C1RGPTL	CAN1 module receive history list get pointer register L	R/W	×	×		01H
0000662H	C1LOPT	CAN1 module last out-pointer register	R		×		Undefined
0000664H	C1TGPT	CAN1 module transmit history list get pointer register	R/W			×	Undefined
0000664H	C1TGPTL	CAN1 module transmit history list get pointer register L	R/W	×	×		01H

Table 3-6: Programmable Peripheral I/O Registers (10/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
0000666H	C1TS	CAN1 module time stamp register	R/W			×	0000H
0000666H	C1TSL	CAN1 module time stamp register L	R/W	×	×		00H
0000667H	C1TSH	CAN1 module time stamp register H	R/W	×	×		00H
0000700H	C1MDATA0100	CAN1 message data byte 0 and 1 register 00	R/W			×	Undefined
0000700H	C1MDATA000	CAN1 message data byte 0 register 00	R/W	×	×		Undefined
0000701H	C1MDATA100	CAN1 message data byte 1 register 00	R/W	×	×		Undefined
0000702H	C1MDATA2300	CAN1 message data byte 2 and 3 register 00	R/W			×	Undefined
0000702H	C1MDATA200	CAN1 message data byte 2 register 00	R/W	×	×		Undefined
0000703H	C1MDATA300	CAN1 message data byte 3 register 00	R/W	×	×		Undefined
0000704H	C1MDATA4500	CAN1 message data byte 4 and 5 register 00	R/W			×	Undefined
0000704H	C1MDATA400	CAN1 message data byte 2 register 00	R/W	×	×		Undefined
0000705H	C1MDATA500	CAN1 message data byte 3 register 00	R/W	×	×		Undefined
0000706H	C1MDATA6700	CAN1 message data byte 6 and 7 register 00	R/W			×	Undefined
0000706H	C1MDATA600	CAN1 message data byte 6 register 00	R/W	×	×		Undefined
0000707H	C1MDATA700	CAN1 message data byte 7 register 00	R/W	×	×		Undefined
0000708H	C1MDLC00	CAN1 message data length code register 00	R/W	×	×		Undefined
0000709H	C1MCONF00	CAN1 message configuration register 00	R/W	×	×		Undefined
000070AH	C1MIDL00	CAN1 message identifier L register 00	R/W			×	Undefined
000070CH	C1MIDH00	CAN1 message identifier H register 00	R/W			×	Undefined
000070EH	C1MCTRL00	CAN1 message control register 00	R/W			×	Undefined
0000720H	C1MDATA0101	CAN1 message data byte 0 and 1 register 01	R/W			×	Undefined
0000720H	C1MDATA001	CAN1 message data byte 0 register 01	R/W	×	×		Undefined
0000721H	C1MDATA101	CAN1 message data byte 1 register 01	R/W	×	×		Undefined
0000722H	C1MDATA2301	CAN1 message data byte 2 and 3 register 01	R/W			×	Undefined
0000722H	C1MDATA201	CAN1 message data byte 2 register 01	R/W	×	×		Undefined
0000723H	C1MDATA301	CAN1 message data byte 3 register 01	R/W	×	×		Undefined
0000724H	C1MDATA4501	CAN1 message data byte 4 and 5 register 01	R/W			×	Undefined
0000724H	C1MDATA401	CAN1 message data byte 2 register 01	R/W	×	×		Undefined
0000725H	C1MDATA501	CAN1 message data byte 3 register 01	R/W	×	×		Undefined
0000726H	C1MDATA6701	CAN1 message data byte 6 and 7 register 01	R/W			×	Undefined
0000726H	C1MDATA601	CAN1 message data byte 6 register 01	R/W	×	×		Undefined
0000727H	C1MDATA701	CAN1 message data byte 7 register 01	R/W	×	×		Undefined
0000728H	C1MDLC01	CAN1 message data length code register 01	R/W	×	×		Undefined
0000729H	C1MCONF01	CAN1 message configuration register 01	R/W	×	×		Undefined
000072AH	C1MIDL01	CAN1 message identifier L register 01	R/W			×	Undefined
000072CH	C1MIDH01	CAN1 message identifier H register 01	R/W			×	Undefined
000072EH	C1MCTRL01	CAN1 message control register 01	R/W			×	Undefined
0000740H	C1MDATA0102	CAN1 message data byte 0 and 1 register 02	R/W			×	Undefined
0000740H	C1MDATA002	CAN1 message data byte 0 register 02	R/W	×	×		Undefined
0000741H	C1MDATA102	CAN1 message data byte 1 register 02	R/W	×	×		Undefined

Table 3-6: Programmable Peripheral I/O Registers (11/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
0000742H	C1MDATA2302	CAN1 message data byte 2 and 3 register 02	R/W			×	Undefined
0000742H	C1MDATA202	CAN1 message data byte 2 register 02	R/W	×	×		Undefined
0000743H	C1MDATA302	CAN1 message data byte 3 register 02	R/W	×	×		Undefined
0000744H	C1MDATA4502	CAN1 message data byte 4 and 5 register 02	R/W			×	Undefined
0000744H	C1MDATA402	CAN1 message data byte 2 register 02	R/W	×	×		Undefined
0000745H	C1MDATA502	CAN1 message data byte 3 register 02	R/W	×	×		Undefined
0000746H	C1MDATA6702	CAN1 message data byte 6 and 7 register 02	R/W			×	Undefined
0000746H	C1MDATA602	CAN1 message data byte 6 register 02	R/W	×	×		Undefined
0000747H	C1MDATA702	CAN1 message data byte 7 register 02	R/W	×	×		Undefined
0000748H	C1MDLC02	CAN1 message data length code register 02	R/W	×	×		Undefined
0000749H	C1MCONF02	CAN1 message configuration register 02	R/W	×	×		Undefined
000074AH	C1MIDL02	CAN1 message identifier L register 02	R/W			×	Undefined
000074CH	C1MIDH02	CAN1 message identifier H register 02	R/W			×	Undefined
000074EH	C1MCTRL02	CAN1 message control register 02	R/W			×	Undefined
0000760H	C1MDATA0103	CAN1 message data byte 0 and 1 register 03	R/W			×	Undefined
0000760H	C1MDATA003	CAN1 message data byte 0 register 03	R/W	×	×		Undefined
0000761H	C1MDATA103	CAN1 message data byte 1 register 03	R/W	×	×		Undefined
0000762H	C1MDATA2303	CAN1 message data byte 2 and 3 register 03	R/W			×	Undefined
0000762H	C1MDATA203	CAN1 message data byte 2 register 03	R/W	×	×		Undefined
0000763H	C1MDATA303	CAN1 message data byte 3 register 03	R/W	×	×		Undefined
0000764H	C1MDATA4503	CAN1 message data byte 4 and 5 register 03	R/W			×	Undefined
0000764H	C1MDATA403	CAN1 message data byte 2 register 03	R/W	×	×		Undefined
0000765H	C1MDATA503	CAN1 message data byte 3 register 03	R/W	×	×		Undefined
0000766H	C1MDATA6703	CAN1 message data byte 6 and 7 register 03	R/W			×	Undefined
0000766H	C1MDATA603	CAN1 message data byte 6 register 03	R/W	×	×		Undefined
0000767H	C1MDATA703	CAN1 message data byte 7 register 03	R/W	×	×		Undefined
0000768H	C1MDLC03	CAN1 message data length code register 03	R/W	×	×		Undefined
0000769H	C1MCONF03	CAN1 message configuration register 03	R/W	×	×		Undefined
000076AH	C1MIDL03	CAN1 message identifier L register 03	R/W			×	Undefined
000076CH	C1MIDH03	CAN1 message identifier H register 03	R/W			×	Undefined
000076EH	C1MCTRL03	CAN1 message control register 03	R/W			×	Undefined
0000780H	C1MDATA0104	CAN1 message data byte 0 and 1 register 04	R/W			×	Undefined
0000780H	C1MDATA004	CAN1 message data byte 0 register 04	R/W	×	×		Undefined
0000781H	C1MDATA104	CAN1 message data byte 1 register 04	R/W	×	×		Undefined
0000782H	C1MDATA2304	CAN1 message data byte 2 and 3 register 04	R/W			×	Undefined
0000782H	C1MDATA204	CAN1 message data byte 2 register 04	R/W	×	×		Undefined
0000783H	C1MDATA304	CAN1 message data byte 3 register 04	R/W	×	×		Undefined
0000784H	C1MDATA4504	CAN1 message data byte 4 and 5 register 04	R/W			×	Undefined
0000784H	C1MDATA404	CAN1 message data byte 2 register 04	R/W	×	×		Undefined
0000785H	C1MDATA504	CAN1 message data byte 3 register 04	R/W	×	×		Undefined

Table 3-6: Programmable Peripheral I/O Registers (12/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
0000786H	C1MDATA6704	CAN1 message data byte 6 and 7 register 04	R/W			×	Undefined
0000786H	C1MDATA604	CAN1 message data byte 6 register 04	R/W	×	×		Undefined
0000787H	C1MDATA704	CAN1 message data byte 7 register 04	R/W	×	×		Undefined
0000788H	C1MDLC04	CAN1 message data length code register 04	R/W	×	×		Undefined
0000789H	C1MCONF04	CAN1 message configuration register 04	R/W	×	×		Undefined
000078AH	C1MIDL04	CAN1 message identifier L register 04	R/W			×	Undefined
000078CH	C1MIDH04	CAN1 message identifier H register 04	R/W			×	Undefined
000078EH	C1MCTRL04	CAN1 message control register 04	R/W			×	Undefined
00007A0H	C1MDATA0105	CAN1 message data byte 0 and 1 register 05	R/W			×	Undefined
00007A0H	C1MDATA005	CAN1 message data byte 0 register 05	R/W	×	×		Undefined
00007A1H	C1MDATA105	CAN1 message data byte 1 register 05	R/W	×	×		Undefined
00007A2H	C1MDATA2305	CAN1 message data byte 2 and 3 register 05	R/W			×	Undefined
00007A2H	C1MDATA205	CAN1 message data byte 2 register 05	R/W	×	×		Undefined
00007A3H	C1MDATA305	CAN1 message data byte 3 register 05	R/W	×	×		Undefined
00007A4H	C1MDATA4505	CAN1 message data byte 4 and 5 register 05	R/W			×	Undefined
00007A4H	C1MDATA405	CAN1 message data byte 2 register 05	R/W	×	×		Undefined
00007A5H	C1MDATA505	CAN1 message data byte 3 register 05	R/W	×	×		Undefined
00007A6H	C1MDATA6705	CAN1 message data byte 6 and 7 register 05	R/W			×	Undefined
00007A6H	C1MDATA605	CAN1 message data byte 6 register 05	R/W	×	×		Undefined
00007A7H	C1MDATA705	CAN1 message data byte 7 register 05	R/W	×	×		Undefined
00007A8H	C1MDLC05	CAN1 message data length code register 05	R/W	×	×		Undefined
00007A9H	C1MCONF05	CAN1 message configuration register 05	R/W	×	×		Undefined
00007AAH	C1MIDL05	CAN1 message identifier L register 05	R/W			×	Undefined
00007ACH	C1MIDH05	CAN1 message identifier H register 05	R/W			×	Undefined
00007AEH	C1MCTRL05	CAN1 message control register 05	R/W			×	Undefined
00007C0H	C1MDATA0106	CAN1 message data byte 0 and 1 register 06	R/W			×	Undefined
00007C0H	C1MDATA006	CAN1 message data byte 0 register 06	R/W	×	×		Undefined
00007C1H	C1MDATA106	CAN1 message data byte 1 register 06	R/W	×	×		Undefined
00007C2H	C1MDATA2306	CAN1 message data byte 2 and 3 register 06	R/W			×	Undefined
00007C2H	C1MDATA206	CAN1 message data byte 2 register 06	R/W	×	×		Undefined
00007C3H	C1MDATA306	CAN1 message data byte 3 register 06	R/W	×	×		Undefined
00007C4H	C1MDATA4506	CAN1 message data byte 4 and 5 register 06	R/W			×	Undefined
00007C4H	C1MDATA406	CAN1 message data byte 2 register 06	R/W	×	×		Undefined
00007C5H	C1MDATA506	CAN1 message data byte 3 register 06	R/W	×	×		Undefined
00007C6H	C1MDATA6706	CAN1 message data byte 6 and 7 register 06	R/W			×	Undefined
00007C6H	C1MDATA606	CAN1 message data byte 6 register 06	R/W	×	×		Undefined
00007C7H	C1MDATA706	CAN1 message data byte 7 register 06	R/W	×	×		Undefined
00007C8H	C1MDLC06	CAN1 message data length code register 06	R/W	×	×		Undefined
00007C9H	C1MCONF06	CAN1 message configuration register 06	R/W	×	×		Undefined
00007CAH	C1MIDL06	CAN1 message identifier L register 06	R/W			×	Undefined

Table 3-6: Programmable Peripheral I/O Registers (13/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
00007CCH	C1MIDH06	CAN1 message identifier H register 06	R/W			×	Undefined
00007CEH	C1MCTRL06	CAN1 message control register 06	R/W			×	Undefined
00007E0H	C1MDATA0107	CAN1 message data byte 0 and 1 register 07	R/W			×	Undefined
00007E0H	C1MDATA007	CAN1 message data byte 0 register 07	R/W	×	×		Undefined
00007E1H	C1MDATA107	CAN1 message data byte 1 register 07	R/W	×	×		Undefined
00007E2H	C1MDATA2307	CAN1 message data byte 2 and 3 register 07	R/W			×	Undefined
00007E2H	C1MDATA207	CAN1 message data byte 2 register 07	R/W	×	×		Undefined
00007E3H	C1MDATA307	CAN1 message data byte 3 register 07	R/W	×	×		Undefined
00007E4H	C1MDATA4507	CAN1 message data byte 4 and 5 register 07	R/W			×	Undefined
00007E4H	C1MDATA407	CAN1 message data byte 2 register 07	R/W	×	×		Undefined
00007E5H	C1MDATA507	CAN1 message data byte 3 register 07	R/W	×	×		Undefined
00007E6H	C1MDATA6707	CAN1 message data byte 6 and 7 register 07	R/W			×	Undefined
00007E6H	C1MDATA607	CAN1 message data byte 6 register 07	R/W	×	×		Undefined
00007E7H	C1MDATA707	CAN1 message data byte 7 register 07	R/W	×	×		Undefined
00007E8H	C1MDLC07	CAN1 message data length code register 07	R/W	×	×		Undefined
00007E9H	C1MCONF07	CAN1 message configuration register 07	R/W	×	×		Undefined
00007EAH	C1MIDL07	CAN1 message identifier L register 07	R/W			×	Undefined
00007ECH	C1MIDH07	CAN1 message identifier H register 07	R/W			×	Undefined
00007EEH	C1MCTRL07	CAN1 message control register 07	R/W			×	Undefined
0000800H	C1MDATA0108	CAN1 message data byte 0 and 1 register 08	R/W			×	Undefined
0000800H	C1MDATA008	CAN1 message data byte 0 register 08	R/W	×	×		Undefined
0000801H	C1MDATA108	CAN1 message data byte 1 register 08	R/W	×	×		Undefined
0000802H	C1MDATA2308	CAN1 message data byte 2 and 3 register 08	R/W			×	Undefined
0000802H	C1MDATA208	CAN1 message data byte 2 register 08	R/W	×	×		Undefined
0000803H	C1MDATA308	CAN1 message data byte 3 register 08	R/W	×	×		Undefined
0000804H	C1MDATA4508	CAN1 message data byte 4 and 5 register 08	R/W			×	Undefined
0000804H	C1MDATA408	CAN1 message data byte 2 register 08	R/W	×	×		Undefined
0000805H	C1MDATA508	CAN1 message data byte 3 register 08	R/W	×	×		Undefined
0000806H	C1MDATA6708	CAN1 message data byte 6 and 7 register 08	R/W			×	Undefined
0000806H	C1MDATA608	CAN1 message data byte 6 register 08	R/W	×	×		Undefined
0000807H	C1MDATA708	CAN1 message data byte 7 register 08	R/W	×	×		Undefined
0000808H	C1MDLC08	CAN1 message data length code register 08	R/W	×	×		Undefined
0000809H	C1MCONF08	CAN1 message configuration register 08	R/W	×	×		Undefined
000080AH	C1MIDL08	CAN1 message identifier L register 08	R/W			×	Undefined
000080CH	C1MIDH08	CAN1 message identifier H register 08	R/W			×	Undefined
000080EH	C1MCTRL08	CAN1 message control register 08	R/W			×	Undefined
0000820H	C1MDATA0109	CAN1 message data byte 0 and 1 register 09	R/W			×	Undefined
0000820H	C1MDATA009	CAN1 message data byte 0 register 09	R/W	×	×		Undefined
0000821H	C1MDATA109	CAN1 message data byte 1 register 09	R/W	×	×		Undefined

Table 3-6: Programmable Peripheral I/O Registers (14/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
0000822H	C1MDATA2309	CAN1 message data byte 2 and 3 register 09	R/W			×	Undefined
0000822H	C1MDATA209	CAN1 message data byte 2 register 09	R/W	×	×		Undefined
0000823H	C1MDATA309	CAN1 message data byte 3 register 09	R/W	×	×		Undefined
0000824H	C1MDATA4509	CAN1 message data byte 4 and 5 register 09	R/W			×	Undefined
0000824H	C1MDATA409	CAN1 message data byte 2 register 09	R/W	×	×		Undefined
0000825H	C1MDATA509	CAN1 message data byte 3 register 09	R/W	×	×		Undefined
0000826H	C1MDATA6709	CAN1 message data byte 6 and 7 register 09	R/W			×	Undefined
0000826H	C1MDATA609	CAN1 message data byte 6 register 09	R/W	×	×		Undefined
0000827H	C1MDATA709	CAN1 message data byte 7 register 09	R/W	×	×		Undefined
0000828H	C1MDLC09	CAN1 message data length code register 09	R/W	×	×		Undefined
0000829H	C1MCONF09	CAN1 message configuration register 09	R/W	×	×		Undefined
000082AH	C1MIDL09	CAN1 message identifier L register 09	R/W			×	Undefined
000082CH	C1MIDH09	CAN1 message identifier H register 09	R/W			×	Undefined
000082EH	C1MCTRL09	CAN1 message control register 09	R/W			×	Undefined
0000840H	C1MDATA0110	CAN1 message data byte 0 and 1 register 10	R/W			×	Undefined
0000840H	C1MDATA010	CAN1 message data byte 0 register 10	R/W	×	×		Undefined
0000841H	C1MDATA110	CAN1 message data byte 1 register 10	R/W	×	×		Undefined
0000842H	C1MDATA2310	CAN1 message data byte 2 and 3 register 10	R/W			×	Undefined
0000842H	C1MDATA210	CAN1 message data byte 2 register 10	R/W	×	×		Undefined
0000843H	C1MDATA310	CAN1 message data byte 3 register 10	R/W	×	×		Undefined
0000844H	C1MDATA4510	CAN1 message data byte 4 and 5 register 10	R/W			×	Undefined
0000844H	C1MDATA410	CAN1 message data byte 2 register 10	R/W	×	×		Undefined
0000845H	C1MDATA510	CAN1 message data byte 3 register 10	R/W	×	×		Undefined
0000846H	C1MDATA6710	CAN1 message data byte 6 and 7 register 10	R/W			×	Undefined
0000846H	C1MDATA610	CAN1 message data byte 6 register 10	R/W	×	×		Undefined
0000847H	C1MDATA710	CAN1 message data byte 7 register 10	R/W	×	×		Undefined
0000848H	C1MDLC10	CAN1 message data length code register 10	R/W	×	×		Undefined
0000849H	C1MCONF10	CAN1 message configuration register 10	R/W	×	×		Undefined
000084AH	C1MIDL10	CAN1 message identifier L register 10	R/W			×	Undefined
000084CH	C1MIDH10	CAN1 message identifier H register 10	R/W			×	Undefined
000084EH	C1MCTRL10	CAN1 message control register 10	R/W			×	Undefined
0000860H	C1MDATA0111	CAN1 message data byte 0 and 1 register 11	R/W			×	Undefined
0000860H	C1MDATA011	CAN1 message data byte 0 register 11	R/W	×	×		Undefined
0000861H	C1MDATA111	CAN1 message data byte 1 register 11	R/W	×	×		Undefined
0000862H	C1MDATA2311	CAN1 message data byte 2 and 3 register 11	R/W			×	Undefined
0000862H	C1MDATA211	CAN1 message data byte 2 register 11	R/W	×	×		Undefined
0000863H	C1MDATA311	CAN1 message data byte 3 register 11	R/W	×	×		Undefined
0000864H	C1MDATA4511	CAN1 message data byte 4 and 5 register 11	R/W			×	Undefined
0000864H	C1MDATA411	CAN1 message data byte 2 register 11	R/W	×	×		Undefined
0000865H	C1MDATA511	CAN1 message data byte 3 register 11	R/W	×	×		Undefined

Table 3-6: Programmable Peripheral I/O Registers (15/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
0000866H	C1MDATA6711	CAN1 message data byte 6 and 7 register 11	R/W			×	Undefined
0000866H	C1MDATA611	CAN1 message data byte 6 register 11	R/W	×	×		Undefined
0000867H	C1MDATA711	CAN1 message data byte 7 register 11	R/W	×	×		Undefined
0000868H	C1MDLC11	CAN1 message data length code register 11	R/W	×	×		Undefined
0000869H	C1MCONF11	CAN1 message configuration register 11	R/W	×	×		Undefined
000086AH	C1MIDL11	CAN1 message identifier L register 11	R/W			×	Undefined
000086CH	C1MIDH11	CAN1 message identifier H register 11	R/W			×	Undefined
000086EH	C1MCTRL11	CAN1 message control register 11	R/W			×	Undefined
0000880H	C1MDATA0112	CAN1 message data byte 0 and 1 register 12	R/W			×	Undefined
0000880H	C1MDATA012	CAN1 message data byte 0 register 12	R/W	×	×		Undefined
0000881H	C1MDATA112	CAN1 message data byte 1 register 12	R/W	×	×		Undefined
0000882H	C1MDATA2312	CAN1 message data byte 2 and 3 register 12	R/W			×	Undefined
0000882H	C1MDATA212	CAN1 message data byte 2 register 12	R/W	×	×		Undefined
0000883H	C1MDATA312	CAN1 message data byte 3 register 12	R/W	×	×		Undefined
0000884H	C1MDATA4512	CAN1 message data byte 4 and 5 register 12	R/W			×	Undefined
0000884H	C1MDATA412	CAN1 message data byte 2 register 12	R/W	×	×		Undefined
0000885H	C1MDATA512	CAN1 message data byte 3 register 12	R/W	×	×		Undefined
0000886H	C1MDATA6712	CAN1 message data byte 6 and 7 register 12	R/W			×	Undefined
0000886H	C1MDATA612	CAN1 message data byte 6 register 12	R/W	×	×		Undefined
0000887H	C1MDATA712	CAN1 message data byte 7 register 12	R/W	×	×		Undefined
0000888H	C1MDLC12	CAN1 message data length code register 12	R/W	×	×		Undefined
0000889H	C1MCONF12	CAN1 message configuration register 12	R/W	×	×		Undefined
000088AH	C1MIDL12	CAN1 message identifier L register 12	R/W			×	Undefined
000088CH	C1MIDH12	CAN1 message identifier H register 12	R/W			×	Undefined
000088EH	C1MCTRL12	CAN1 message control register 12	R/W			×	Undefined
00008A0H	C1MDATA0113	CAN1 message data byte 0 and 1 register 13	R/W			×	Undefined
00008A0H	C1MDATA013	CAN1 message data byte 0 register 13	R/W	×	×		Undefined
00008A1H	C1MDATA113	CAN1 message data byte 1 register 13	R/W	×	×		Undefined
00008A2H	C1MDATA2313	CAN1 message data byte 2 and 3 register 13	R/W			×	Undefined
00008A2H	C1MDATA213	CAN1 message data byte 2 register 13	R/W	×	×		Undefined
00008A3H	C1MDATA313	CAN1 message data byte 3 register 13	R/W	×	×		Undefined
00008A4H	C1MDATA4513	CAN1 message data byte 4 and 5 register 13	R/W			×	Undefined
00008A4H	C1MDATA413	CAN1 message data byte 2 register 13	R/W	×	×		Undefined
00008A5H	C1MDATA513	CAN1 message data byte 3 register 13	R/W	×	×		Undefined
00008A6H	C1MDATA6713	CAN1 message data byte 6 and 7 register 13	R/W			×	Undefined
00008A6H	C1MDATA613	CAN1 message data byte 6 register 13	R/W	×	×		Undefined
00008A7H	C1MDATA713	CAN1 message data byte 7 register 13	R/W	×	×		Undefined
00008A8H	C1MDLC13	CAN1 message data length code register 13	R/W	×	×		Undefined
00008A9H	C1MCONF13	CAN1 message configuration register 13	R/W	×	×		Undefined
00008AAH	C1MIDL13	CAN1 message identifier L register 13	R/W			×	Undefined

Table 3-6: Programmable Peripheral I/O Registers (16/16)

Address Offset	Symbol	Function Register Name	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
00008ACH	C1MIDH13	CAN1 message identifier H register 13	R/W			×	Undefined
00008AEH	C1MCTRL13	CAN1 message control register 13	R/W			×	Undefined
00008C0H	C1MDATA0114	CAN1 message data byte 0 and 1 register 14	R/W			×	Undefined
00008C0H	C1MDATA014	CAN1 message data byte 0 register 14	R/W	×	×		Undefined
00008C1H	C1MDATA114	CAN1 message data byte 1 register 14	R/W	×	×		Undefined
00008C2H	C1MDATA2314	CAN1 message data byte 2 and 3 register 14	R/W			×	Undefined
00008C2H	C1MDATA214	CAN1 message data byte 2 register 14	R/W	×	×		Undefined
00008C3H	C1MDATA314	CAN1 message data byte 3 register 14	R/W	×	×		Undefined
00008C4H	C1MDATA4514	CAN1 message data byte 4 and 5 register 14	R/W			×	Undefined
00008C4H	C1MDATA414	CAN1 message data byte 2 register 14	R/W	×	×		Undefined
00008C5H	C1MDATA514	CAN1 message data byte 3 register 14	R/W	×	×		Undefined
00008C6H	C1MDATA6714	CAN1 message data byte 6 and 7 register 14	R/W			×	Undefined
00008C6H	C1MDATA614	CAN1 message data byte 6 register 14	R/W	×	×		Undefined
00008C7H	C1MDATA714	CAN1 message data byte 7 register 14	R/W	×	×		Undefined
00008C8H	C1MDLC14	CAN1 message data length code register 14	R/W	×	×		Undefined
00008C9H	C1MCONF14	CAN1 message configuration register 14	R/W	×	×		Undefined
00008CAH	C1MIDL14	CAN1 message identifier L register 14	R/W			×	Undefined
00008CCH	C1MIDH14	CAN1 message identifier H register 14	R/W			×	Undefined
00008CEH	C1MCTRL14	CAN1 message control register 14	R/W			×	Undefined
00008E0H	C1MDATA0115	CAN1 message data byte 0 and 1 register 15	R/W			×	Undefined
00008E0H	C1MDATA015	CAN1 message data byte 0 register 15	R/W	×	×		Undefined
00008E1H	C1MDATA115	CAN1 message data byte 1 register 15	R/W	×	×		Undefined
00008E2H	C1MDATA2315	CAN1 message data byte 2 and 3 register 15	R/W			×	Undefined
00008E2H	C1MDATA215	CAN1 message data byte 2 register 15	R/W	×	×		Undefined
00008E3H	C1MDATA315	CAN1 message data byte 3 register 15	R/W	×	×		Undefined
00008E4H	C1MDATA4515	CAN1 message data byte 4 and 5 register 15	R/W			×	Undefined
00008E4H	C1MDATA415	CAN1 message data byte 2 register 15	R/W	×	×		Undefined
00008E5H	C1MDATA515	CAN1 message data byte 3 register 15	R/W	×	×		Undefined
00008E6H	C1MDATA6715	CAN1 message data byte 6 and 7 register 15	R/W			×	Undefined
00008E6H	C1MDATA615	CAN1 message data byte 6 register 15	R/W	×	×		Undefined
00008E7H	C1MDATA715	CAN1 message data byte 7 register 15	R/W	×	×		Undefined
00008E8H	C1MDLC15	CAN1 message data length code register 15	R/W	×	×		Undefined
00008E9H	C1MCONF15	CAN1 message configuration register 15	R/W	×	×		Undefined
00008EAH	C1MIDL15	CAN1 message identifier L register 15	R/W			×	Undefined
00008ECH	C1MIDH15	CAN1 message identifier H register 15	R/W			×	Undefined
00008EEH	C1MCTRL15	CAN1 message control register 15	R/W			×	Undefined

3.4.8 Specific registers

Specific registers are registers that prevent invalid data from being written if an inadvertent program behaviour occurs.
The V850E/PH2 has the following specific registers:

- Port registers 5 and 6 (P5, P6)
- Port mode registers 5 and 6 (PM5, PM6)
- Port mode control registers 5 and 6 (PMC5, PMC6)
- Port emergency shut off control registers 5 and 6 (PESC5, PESC6)
- Port emergency shut off status registers 5 and 6 (ESOST5, ESOST6)

Moreover, there is also a command register (PRCMD), which is a protection register against write operations to the specific registers. Write access to the specific registers is performed with a special sequence and illegal store operations are notified to the system status register (PHS).

This section of the manual describes the access method to these specific registers, rather than the values that can be written to these registers. For details on these register values, please refer to sections **20.3.6 ”Port 5” on page 909** and **20.3.7 ”Port 6” on page 914**.

(1) Setting data to specific registers

Setting data to a specific registers is done in the following sequence.

<1> Prepare the data to be set to the special register in a general-purpose register.
<2> Write the data prepared in <1> to the command register (PRCMD).
<3> Write the data to the specific register (using the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

Example

```
<1>  MOV  0x02, r10          ; Prepare data in r10
<2>  ST.B  r10, PRCMD[r0]    ; Write PRCMD register
<3>  ST.B  r10, P5           ; Set P5 register
(next instruction)
```

Cautions: 1. Interrupts are not acknowledged when executing the store instruction to the PRCMD register.
If another instruction is placed between steps <2> and <3>, the correct sequence may not be realized if an interrupt is acknowledged for that instruction, resulting in the writing to the protected register to be not done, and an error to be stored in the PRERR bit of the PHS register.

2. If there is a possibility of an active DMA register before <2> and <3>, the specific register may not be written. In this case, ensure that no DMA register is active during the sequence <2> to <3>, or repeat the sequencer <2> to <3> as long as the PRERR bit of the PHS register is set to <1>.

(2) Processor command register (PRCMD)

The PRCMD register is an 8-bit register used to prevent data from being written to registers that may have a large influence on the system, possibly causing the application system to unexpectedly stop. Only the first write operation to a specific register following the execution of a write operation to the PRCMD register, is valid.
As a result, register values can be overwritten only using a preset sequence, preventing invalid write operations.

PRCMD register must be written with store instruction execution by CPU only (not with DMA transfer). If an illegal store operation to a command or specific register takes place, it is reported by the PRERR flag of the system status register (PHS).

This register can be written in 8-bit units only. Undefined data is read from this register.

Figure 3-21: Processor Command Register (PRCMD)

After reset: Undefined W Address: FFFFF1FCH

	7	6	5	4	3	2	1	0
PRCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

(3) System status register (PHS)

The PHS register is an 8-bit register to which the PRERR flag showing the generation of protection errors is assigned.

If a write operation to a specific register has not been executed in the correct sequence including the access to the command register (PRCMD), the write operation to the intended register is not executed, a protection error is generated and the PRERR flag is set to 1. The value of this register becomes “00H” by $\overline{\text{RESET}}$ input.

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

Figure 3-22: System Status Register Format PHS

After reset: 00H R/W Address: FFFFF802H

	7	6	5	4	3	2	1	0
PHS	0	0	0	0	0	0	0	PRERR

PRERR	Detection of Protection Error
0	Protection error did not occur
1	Protection error occurred

The PRERR flag operates under the following conditions.

(a) Setting condition (PRERR flag = 1)

- When a write operation is not performed on the PRCMD register and an operation to write a specific register is performed (when <4> in the example **3.4.8 (1) Setting data to specific registers** is executed without <3>).
- If a write operation (including a bit manipulation instruction) is performed on an on-chip peripheral I/O register other than a specific register after a write operation to the PRCMD register (when <4> in the example **3.4.8 (1) Setting data to specific registers** is not performed for a specific register).

Remark: Even if an on-chip peripheral I/O register is read (including a bit manipulation instruction) between writing the PRCMD register and writing a specific register (such as an access to the internal RAM), the PRERR flag is not set, and data can be written to the special register.

(b) Clearing condition (PREER flag = 0)

- When 0 is written to the PRERR flag of the PHS register.
- When system reset is executed.

Cautions: 1. If 0 is written to the PRERR bit of the PHS register (that is not a specific register) immediately following write to the PRCMD register, the PRERR bit becomes 0 (write priority).

2. If data is written to the PRCMD register (that is not a specific registers) immediately following write to the PRCMD register, the PRERR bit becomes 1.

3.4.9 System wait control register (VSWC)

The system wait control register (VSWC) is a register that controls the bus access wait for the on-chip peripheral I/O registers.
Access to on-chip peripheral I/O registers is made in 3 clocks (without wait), however, in the V850E/PH2 waits may be required depending on the operation frequency. Set the values described in the table below to the VSWC register in accordance with the operation frequency used.
This register can be read or written in 1-bit or 8-bit units.

Register Name	Operating frequency	Set Value	Address	After reset
VSWC	64 MHz	13H	FFFFF06EH	77H

3.4.10 DMA wait control registers 0 and 1 (DMAWC0, DMAWC1)

The DMA wait control registers 0 and 1 (DMAWC0, DMAWC1) are a registers that control the bus access wait and signal timing for DMA transfers.
Set the values described in the table below to the DMAWC0 and DMAWC1 registers in accordance with the operation frequency used.
This register can be read or written in 1-bit or 8-bit units.

Register Name	Operating frequency	Set Value	Address	After reset
DMAWC0	64 MHz	13H	FFFFFE00H	37H
DMAWC1		04H	FFFFFE02H	07H

3.4.11 Cautions

- Initialize the following registers immediately after reset signal release in the following sequence:

  - System wait control register (VSWC)
    (refer to **3.4.9 System wait control register (VSWC)**)

  - DMA wait control registers 0 and 1 (DMAWC0,DMAWC1)
    (refer to **3.4.10 DMA wait control registers 0 and 1 (DMAWC0, DMAWC1)**)

[MEMO]

Chapter 4 Bus Control Function

The V850E/PH2 is provided with an external bus interface function by which external memories, such as ROM and RAM, and external I/O can be connected.

4.1 Features

- 32-bit/16-bit/8-bit data bus sizing function
- 8 chip areas select function
- 4 chip area select signals externally available ($\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$ and $\overline{CS4}$)
- Wait function
  - Programmable wait function, capable of inserting up to 7 wait states for each memory block
  - External wait function via $\overline{WAIT}$ pin
- Idle state insertion function
- External device connection can be enabled via bus control/port alternate function pins
- Programmable Endian format (Little Endian/Big Endian)

4.2 Bus Control Pins

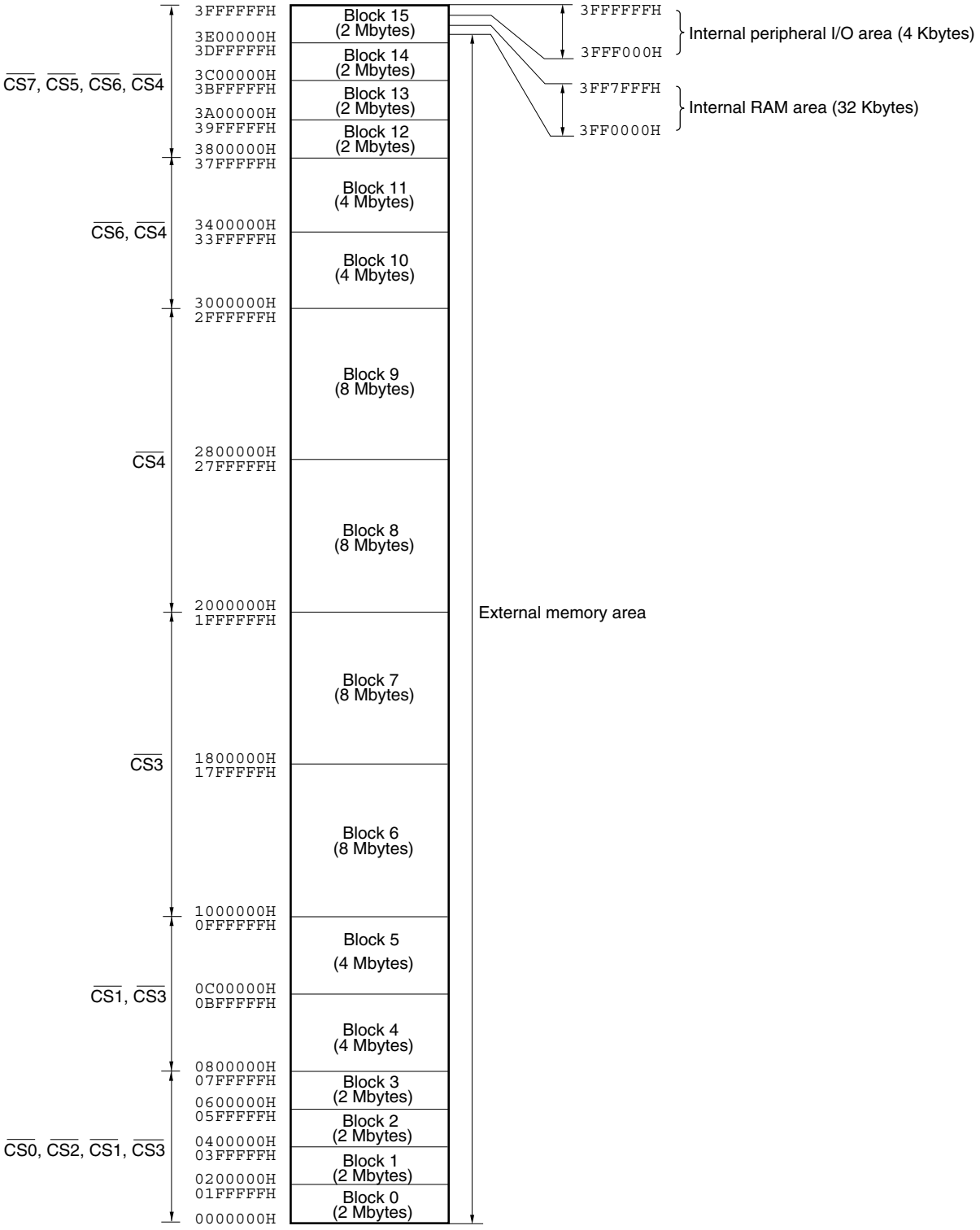
The following pins are used for connecting to external devices.

Bus Control Pin (Function when in Control Mode)	Function when in Port Mode	Register for Port/ Control Mode Switching
Data bus (D0 to D15)	PDL0 to PDL15 (Port DL)	PMCDL
Data bus (D16 to D31)	PDH0 to PDH15 (Port DH)	PMCDH
Address bus (A0 to A15)	PAL0 to PAL15 (Port AL)	PMCAL
Address bus (A16 to A21)	PAH0 to PAH5 (Port AH)	PMCAH
Chip select ($\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$ and $\overline{CS4}$)	PCS0, PCS1, PCS3 and PCS4 (Port CS)	PMCCS
Read/write control ($\overline{RD}$,$\overline{WR}$)	PCT4, PCT5 (Port CT)	PMCCT
Byte enable control ($\overline{BE0}$ to $\overline{BE3}$)	PCD2 to PCD5 (Port CD)	PMCCD
External wait control ($\overline{WAIT}$)	PCM0 (Port CM)	PMCCM

4.3 Memory Block Function

The 64 MB memory space is divided into memory blocks of 2 MB, 4 MB, and 8 MB units.

Figure 4-1: Memory Block Function



4.3.1 Chip select control function

The 64 MB memory area can be divided into 2 MB, 4 MB and 8 MB memory blocks by the chip area selection control registers 0 and 1 (CSC0, CSC1) to control the chip select signals.
The memory area can be effectively used by dividing the memory area into memory blocks using the chip select control function. The priority order is described below.

(1) Chip area selection control registers 0, 1 (CSC0, CSC1)

These registers can be read/written in 16-bit units. Valid by setting each bit (to 1). If different chip area select signals are set to the same block, the priority order is controlled as follows.

CSC0: Peripheral I/O area > $\overline{CS0}$ > $\overline{CS2}$ > $\overline{CS1}$ > $\overline{CS3}$ **Note**
CSC1: Peripheral I/O area > $\overline{CS7}$ > $\overline{CS5}$ > $\overline{CS6}$ > $\overline{CS4}$ **Note**

Note: Not all the chip area select signals are externally available on output pins. Even so, enabling chip area select signals other than $\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$ or $\overline{CS4}$, the setting for the corresponding memory blocks will be effective too, regardless of an external chip select output pin.

Figure 4-2: Chip Area Select Control Registers 0, 1 (1/2)

Figure 4-2: Chip Area Select Control Registers 0, 1 (2/2)

CSnm	Chip Select Operation
CS00	$\overline{CS0}$ active during block 0 access
CS01	$\overline{CS0}$ active during block 1 access.
CS02	$\overline{CS0}$ active during block 2 access.
CS03	$\overline{CS0}$ active during block 3 access.
CS10	$\overline{CS1}$ active during block 0 or 1 access.
CS11	$\overline{CS1}$ active during block 2 or 3 access.
CS12	$\overline{CS1}$ active during block 4 access.
CS13	$\overline{CS1}$ active during block 5 access.
CS20	$\overline{CS2}$ active during block 0 access.
CS21	$\overline{CS2}$ active during block 1 access.
CS22	$\overline{CS2}$ active during block 2 access.
CS23	$\overline{CS2}$ active during block 3 access.
CS30	$\overline{CS3}$ active during block 0, 1, 2, or 3 access.
CS31	$\overline{CS3}$ active during block 4 or 5 access.
CS32	$\overline{CS3}$ active during block 6 access.
CS33	$\overline{CS3}$ active during block 7 access.
CS40	$\overline{CS4}$ active during block 12, 13, 14, or 15 access.
CS41	$\overline{CS4}$ active during block 10 or 11 access.
CS42	$\overline{CS4}$ active during block 9 access.
CS43	$\overline{CS4}$ active during block 8 access.
CS50	$\overline{CS5}$ active during block 15 access.
CS51	$\overline{CS5}$ active during block 14 access.
CS52	$\overline{CS5}$ active during block 13 access.
CS53	$\overline{CS5}$ active during block 12 access.
CS60	$\overline{CS6}$ active during block 14 or 15 access.
CS61	$\overline{CS6}$ active during block 12 or 13 access.
CS62	CS6 active during block 11 access.
CS63	$\overline{CS6}$ active during block 10 access.
CS70	$\overline{CS7}$ active during block 15 access.
CS71	$\overline{CS7}$ active during block 14 access.
CS72	$\overline{CS7}$ active during block 13 access.
CS73	$\overline{CS7}$ active during block 12 access.
Remark:	Dedicated chip select operation is enabled when corresponding CSnm bit is set (1), and disabled when CSnm is cleared (0) (n = 0 to 7, m = 0 to 3)

4.4 Bus Cycle Type Control Function

In the V850E/PH2, the following external devices can be connected directly to each memory block.

- SRAM, external ROM, external I/O

Connected external devices are specified by the bus cycle type configuration registers 0, 1 (BCT0, BCT1).

4.4.1 Bus cycle type configuration

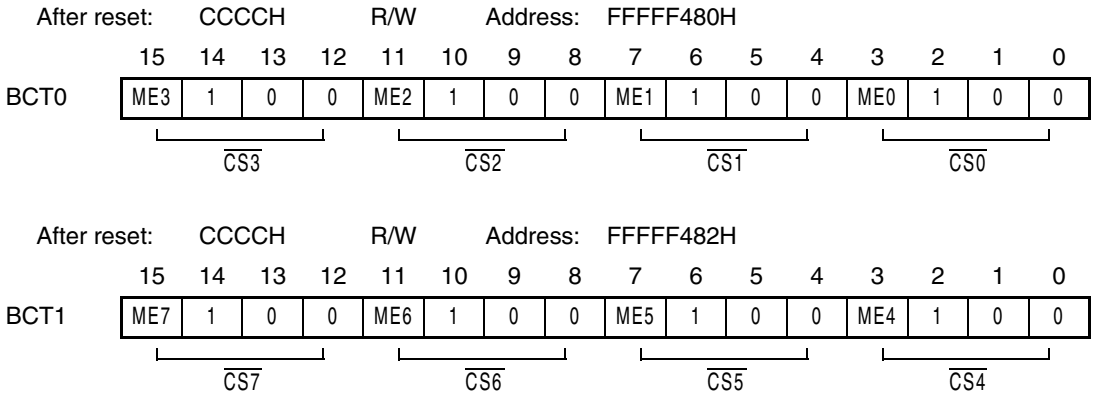
(1) Bus cycle configuration registers 0, 1 (BCT0, BCT1)

These registers can be read/written in 16-bit units

Cautions: 1. Write to the BCT0 and BCT1 registers after reset, and then do not change the set value. Also, do not access an external memory area other than that for this initialization routine until initial setting of the BCT0 and BCT1 registers is finished. However, it is possible to access external memory areas whose initialization has been finished.

2. The bits marked as 0 and 1 are reserved. The values of these bits must not be changed. Otherwise the operation of the external bus interface cannot be ensured.

Figure 4-3: Bus Cycle Configuration Registers 0, 1 (BCT0, BCT1)



MEn	Memory Controller Operation Enable for $\overline{CSn}$ Area
0	Operation disable
1	Operation enable

4.5 Bus Access

4.5.1 Number of access clocks

The number of basic clocks necessary for accessing each resource is as follows.

Table 4-1: Number of Bus Access Clocks

Resources (Bus width) Bus Cycle Configuration		Internal RAM (32 bits)	Peripheral I/O (16 bits)	External memory (16 bits)
Instruction fetch	Normal access	1 Note 1	-	2 Note 2
	Branch	1	-	2 Note 2
Operand data access		1	3 Note 2	2 Note 2

Notes: 1. The instruction fetch becomes 2 clocks, in case of contention with data access.

2. This is the minimum value.

4.5.2 Bus sizing function

The bus sizing function controls data bus width for each CS area. The data bus width is specified by using the bus size configuration register (BSC).

(1) Bus size configuration register (BSC)

This register can be read/written in 16-bit units.

Caution: Write to the BSC register after reset, and then do not change the set value. Also, do not access an external memory area other than that for this initialization routine until initial setting of the BSC register is finished. However, it is possible to access external memory areas whose initialization has been finished.

Figure 4-4: Bus Size Configuration Register (BSC)

After reset: AAAAH R/W Address: FFFFF066H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSC	BS71	BS70	BS61	BS60	BS51	BS50	BS41	BS40	BS31	BS30	BS21	BS20	BS11	BS10	BS01	BS00
	$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$	

BEn1	BEn0	Data Bus Width of $\overline{CSn}$ Area
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	Setting prohibited

4.5.3 Endian control function

The Endian control function can be used to set processing of word data in memory either by the Big Endian method or the Little Endian method for each CS area selected with the chip select signal ($\overline{CS0}$ to $\overline{CS7}$). Switching of the Endian method is specified with the Endian configuration register (BEC).

Figure 4-5: Big Endian Addresses within Word

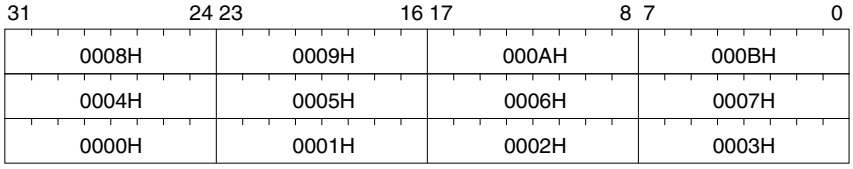
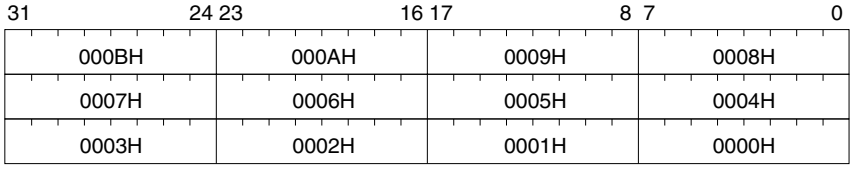


Figure 4-6: Little Endian Addresses within Word



(1) Endian configuration register (BEC)

This register can be read/written in 16-bit units.

Cautions: 1. Bits 15, 13, 11, 9, 7, 5, 3, and 1 of the BEC register must be cleared (0). If these bits are set to 1, the operation is not guaranteed.

2. Set the CSn area specified as the programmable peripheral I/O area to Little Endian format (n = 0 to 7).

3. In the following areas, the data processing method is fixed to Little Endian method. Any setting of Big Endian method for these areas according to the BEC register is invalid.
- **On-chip peripheral I/O area**
- **Internal RAM area**
- **Fetch area of external memory**

Figure 4-7: Endian Configuration Register (BEC)

After reset: 0000H R/W Address: FFFFF068H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BEC	0	BE70	0	BE60	0	BE50	0	BE40	0	BE30	0	BE20	0	BE10	0	BE00
	$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$	

BEn0	Endian Control
0	Little Endian method
1	Big Endian method

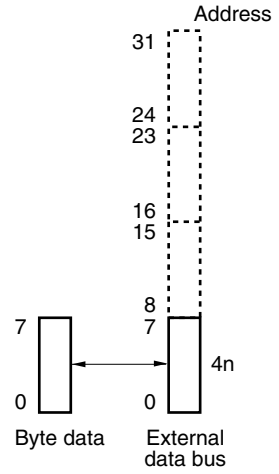
4.5.4 Bus width

The V850E/PH2 accesses peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The following shows the operation for each type of access. Access all data in order starting from the lower order side.

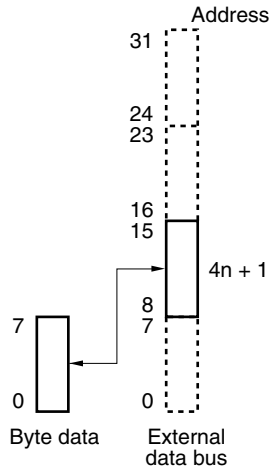
(1) Byte access (8 bits)

(a) When the data bus width is 32 bits (Little Endian)

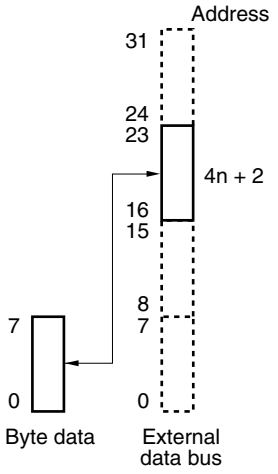
<1> Access to address (4n)



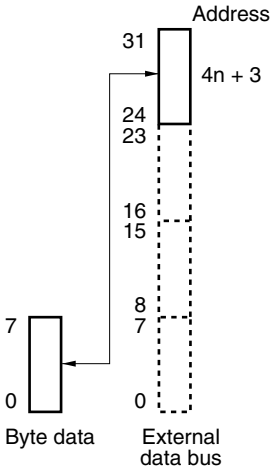
<2> Access to address (4n+1)



<3> Access to address (4n+2)

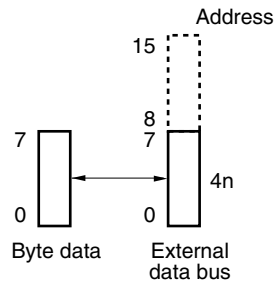


<4> Access to address (4n+3)

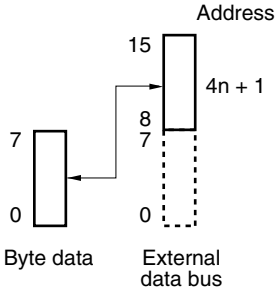


(b) When the data bus width is 16 bits (Little Endian)

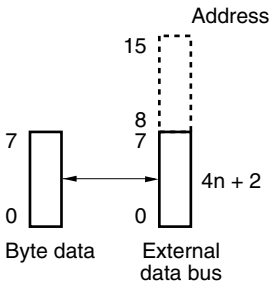
<1> Access to address (4n)



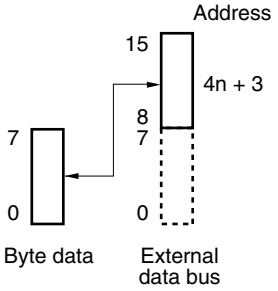
<2> Access to address (4n+1)



<3> Access to address (4n+2)

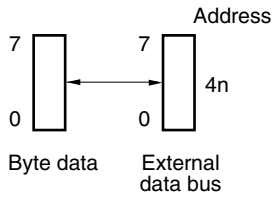


<4> Access to address (4n+3)

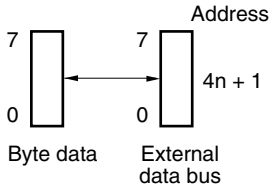


(c) When the data bus width is 8 bits (Little Endian)

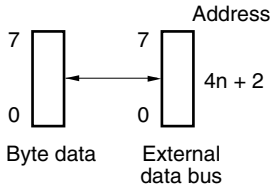
<1> Access to address (4n)



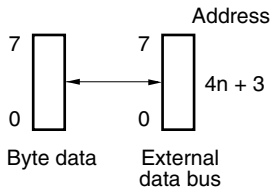
<2> Access to address (4n+1)



<3> Access to address (4n+2)

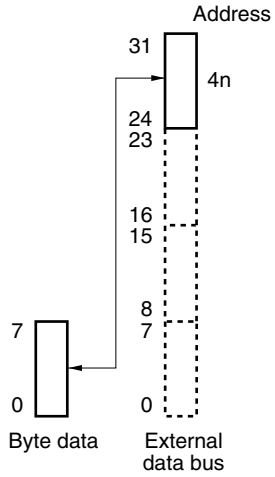


<4> Access to address (4n+3)

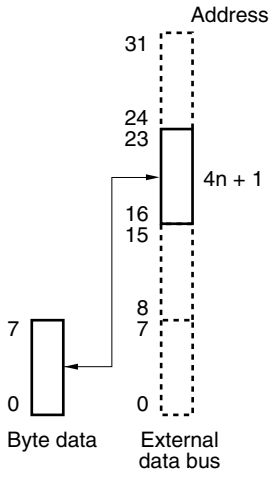


(d) When the data bus width is 32 bits (Big Endian)

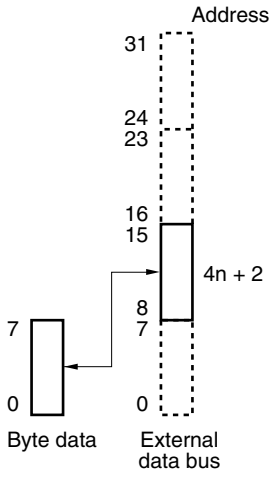
<1> Access to address (4n)



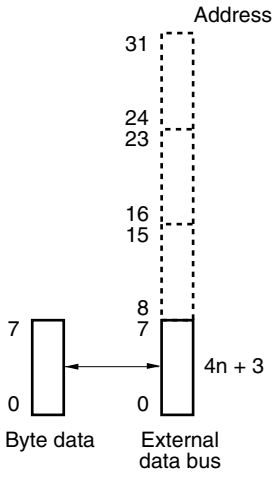
<2> Access to address (4n+1)



<3> Access to address (4n+2)

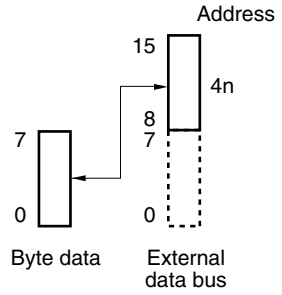


<4> Access to address (4n+3)

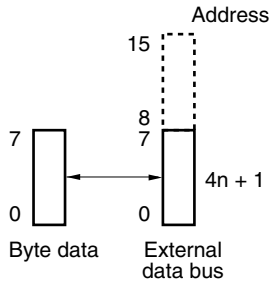


(e) When the data bus width is 16 bits (Big Endian)

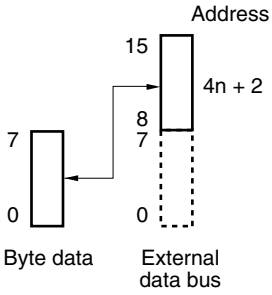
<1> Access to address (4n)



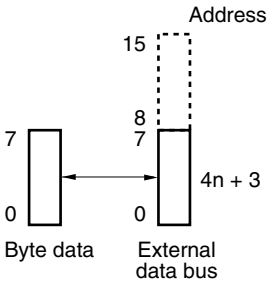
<2> Access to address (4n+1)



<3> Access to address (4n+2)

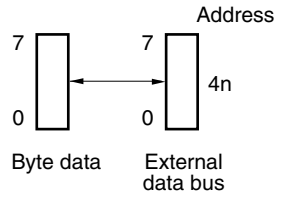


<4> Access to address (4n+3)

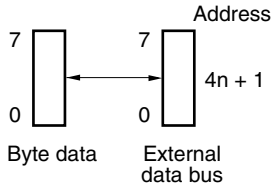


(f) When the data bus width is 8 bits (Big Endian)

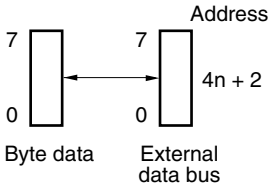
<1> Access to address (4n)



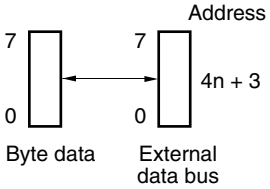
<2> Access to address (4n+1)



<3> Access to address (4n+2)



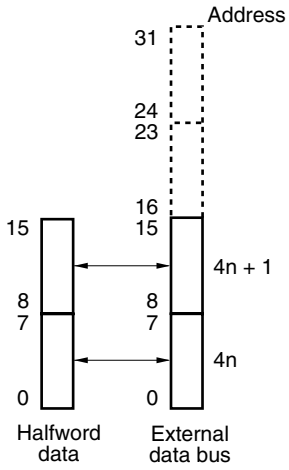
<4> Access to address (4n+3)



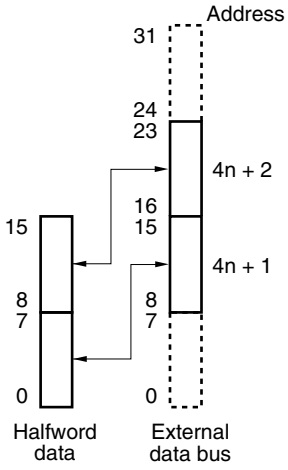
(2) Halfword access (16 bits)

(a) When the data bus width is 32 bits (Little Endian)

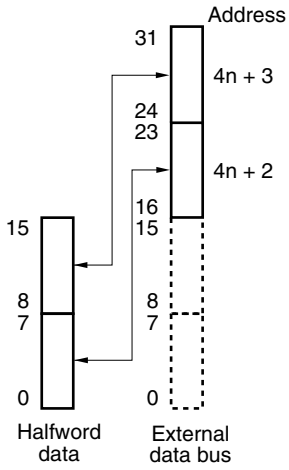
<1> Access to address (4n)



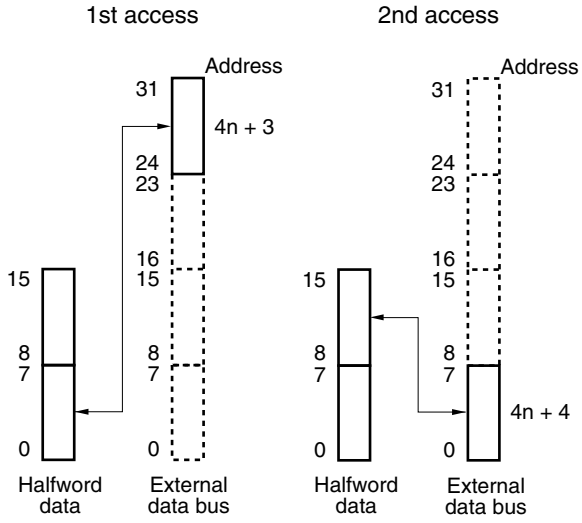
<2> Access to address (4n+1)



<3> Access to address (4n+2)

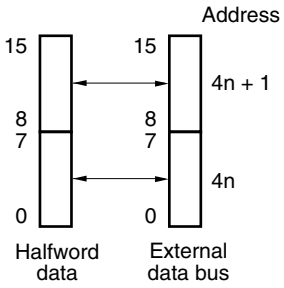


<4> Access to address (4n+3)

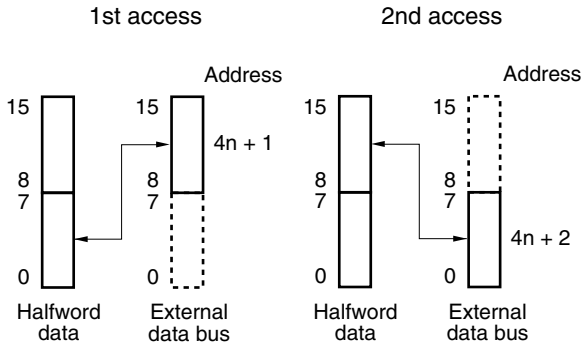


(b) When the data bus width is 16 bits (Little Endian)

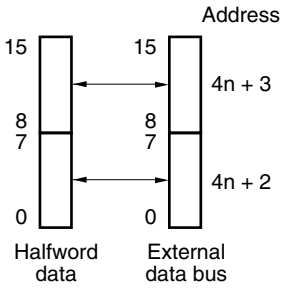
<1> Access to address (4n)



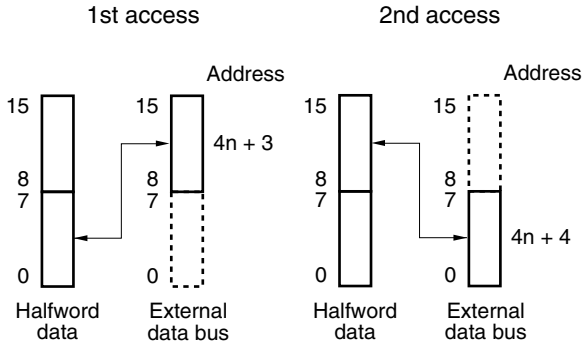
<2> Access to address (4n+1)



<3> Access to address (4n+2)

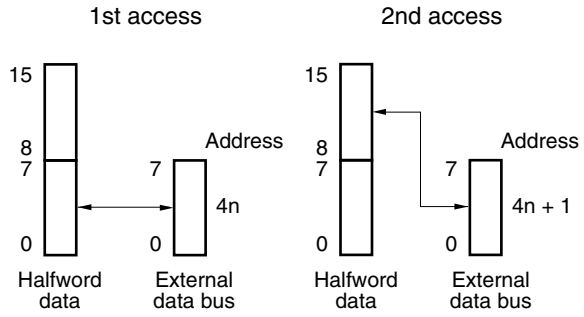


<4> Access to address (4n+3)

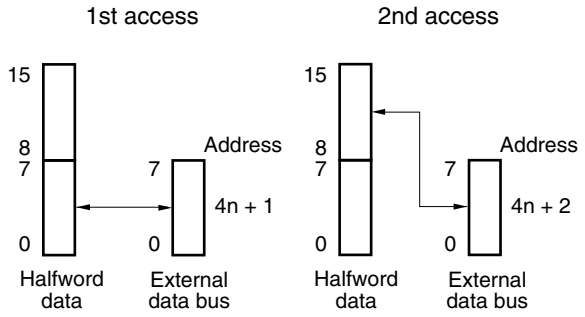


(c) When the data bus width is 8 bits (Little Endian)

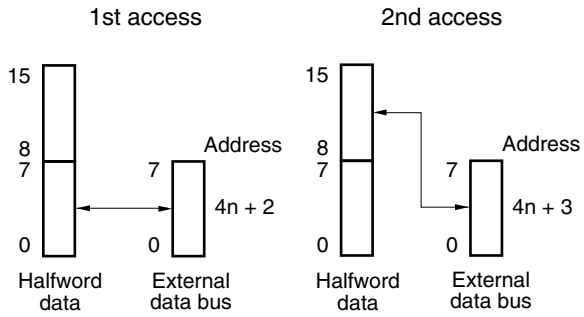
<1> Access to address (4n)



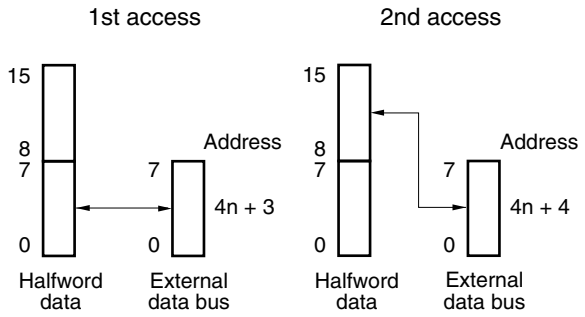
<2> Access to address (4n+1)



<3> Access to address (4n+2)

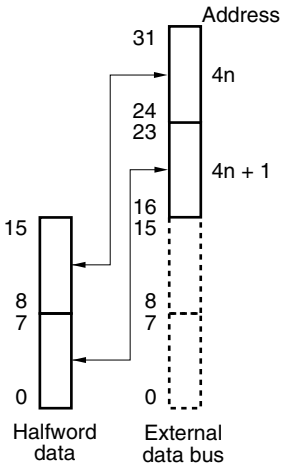


<4> Access to address (4n+3)

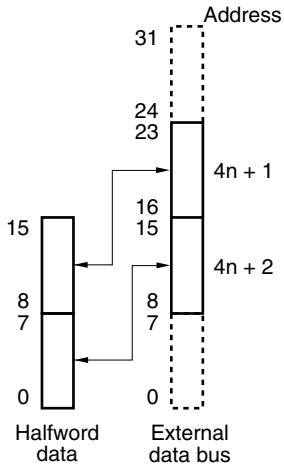


(d) When the data bus width is 32 bits (Big Endian)

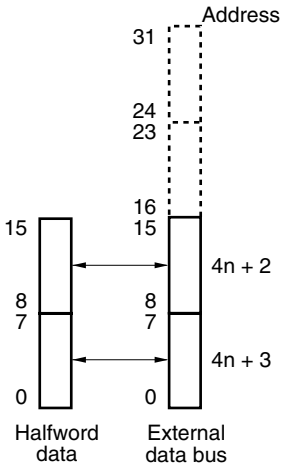
<1> Access to address (4n)



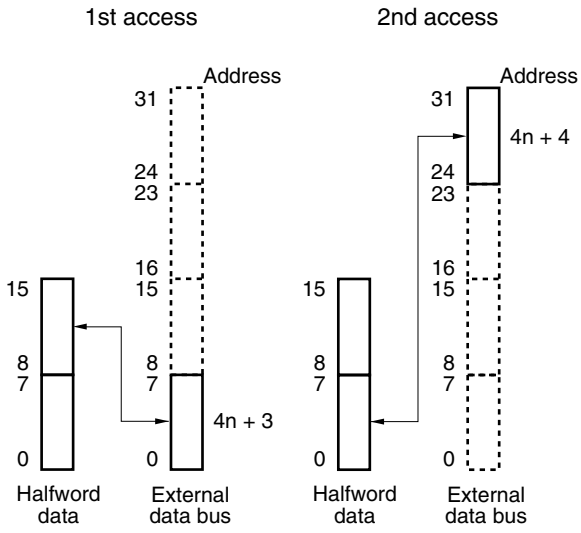
<2> Access to address (4n+1)



<3> Access to address (4n+2)

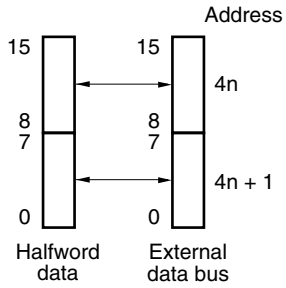


<4> Access to address (4n+3)

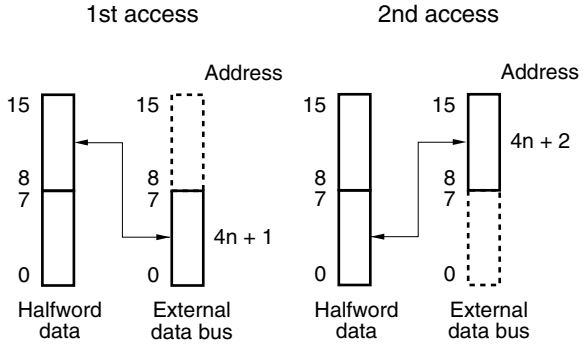


(e) When the data bus width is 16 bits (Big Endian)

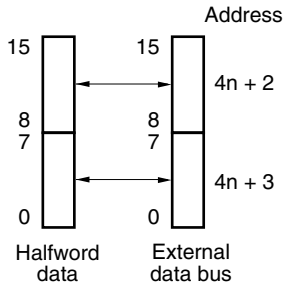
<1> Access to address (4n)



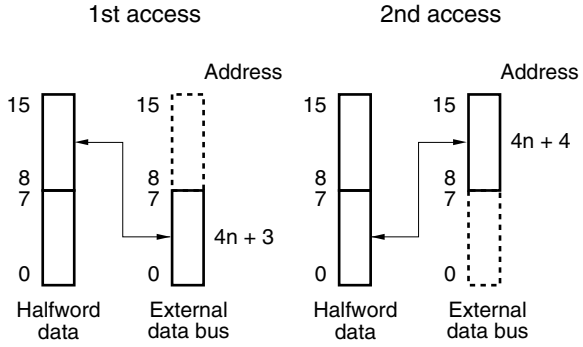
<2> Access to address (4n+1)



<3> Access to address (4n+2)

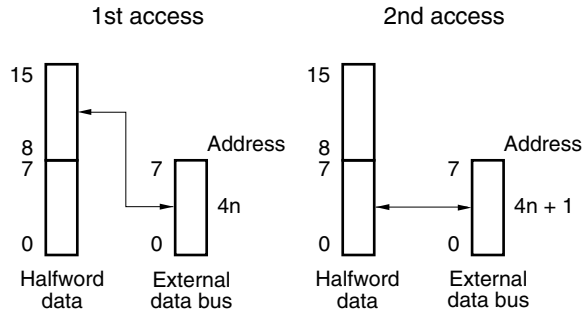


<4> Access to address (4n+3)

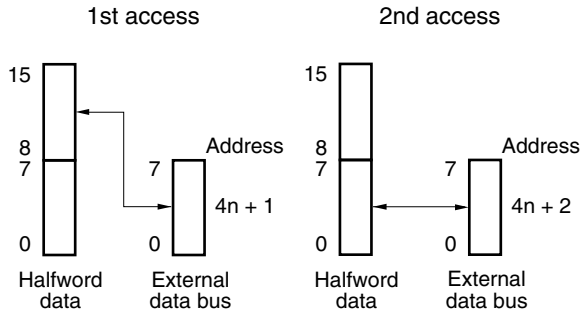


(f) When the data bus width is 8 bits (Big Endian)

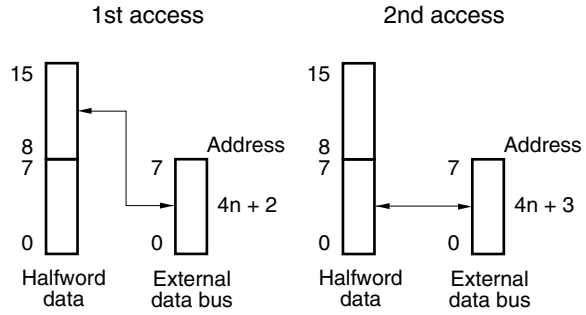
<1> Access to address (4n)



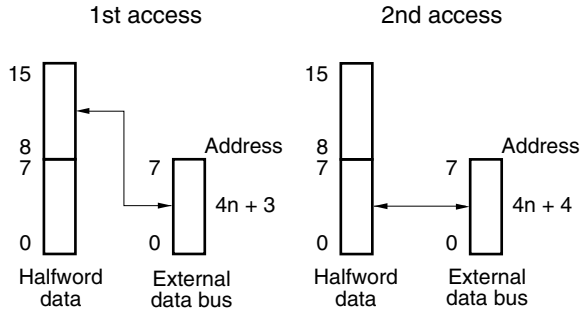
<2> Access to address (4n+1)



<3> Access to address (4n+2)



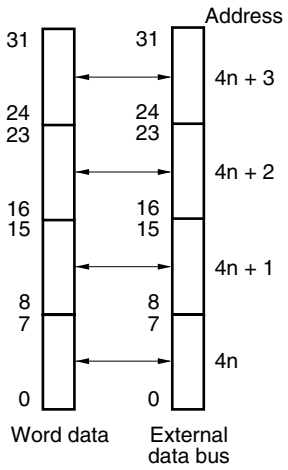
<4> Access to address (4n+3)



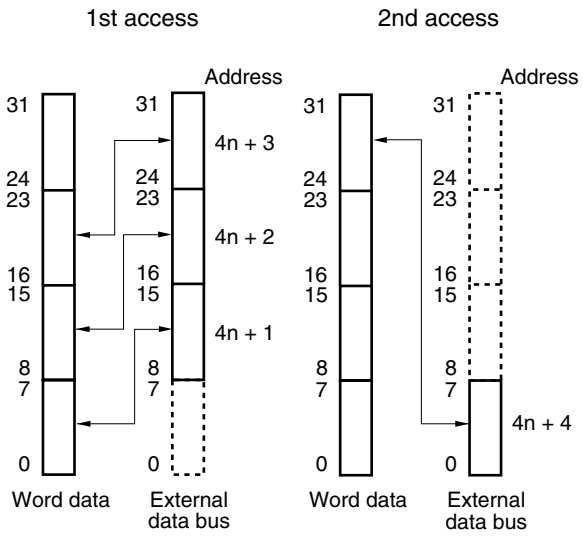
(3) Word access (32 bits)

(a) When the bus width is 32 bits (Little Endian)

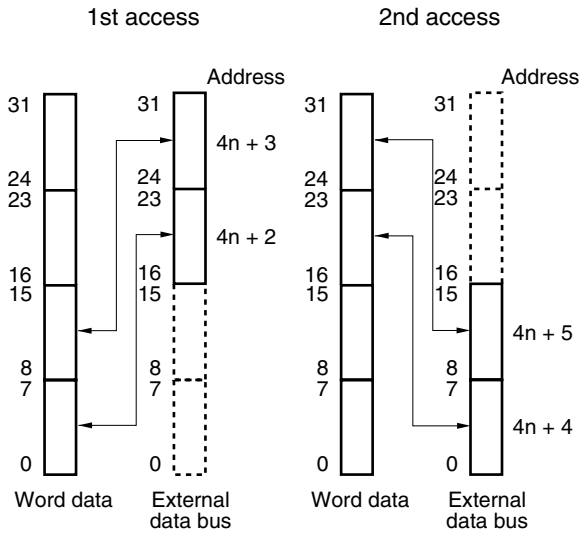
<1> Access to address (4n)



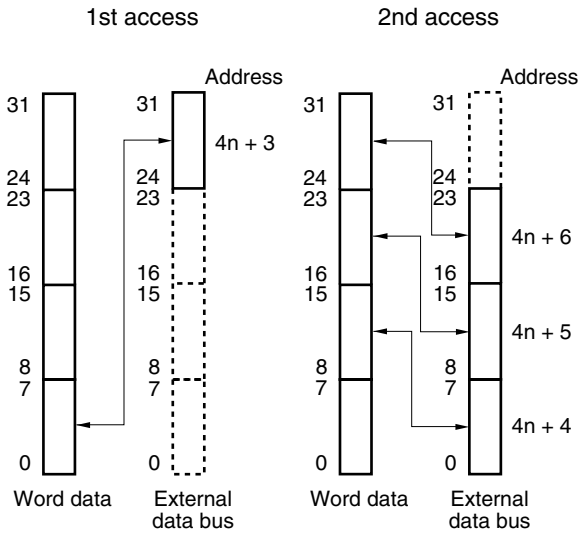
<2> Access to address (4n+1)



<3> Access to address (4n+2)

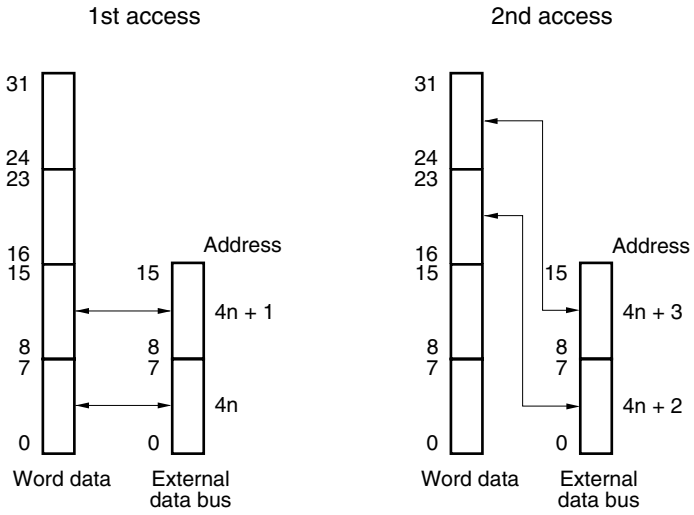


<4> Access to address (4n+3)

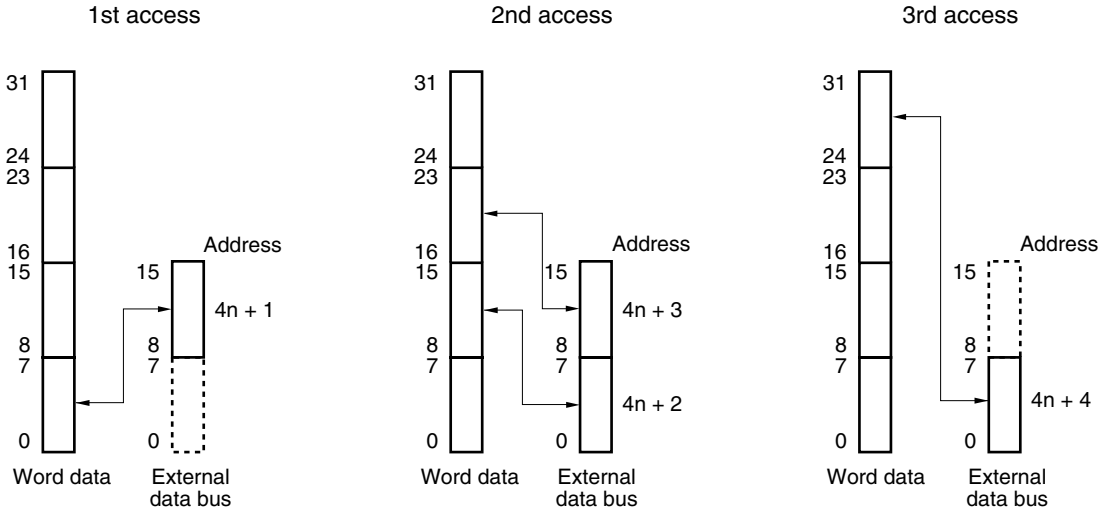


(b) When the bus width is 16 bits (Little Endian)

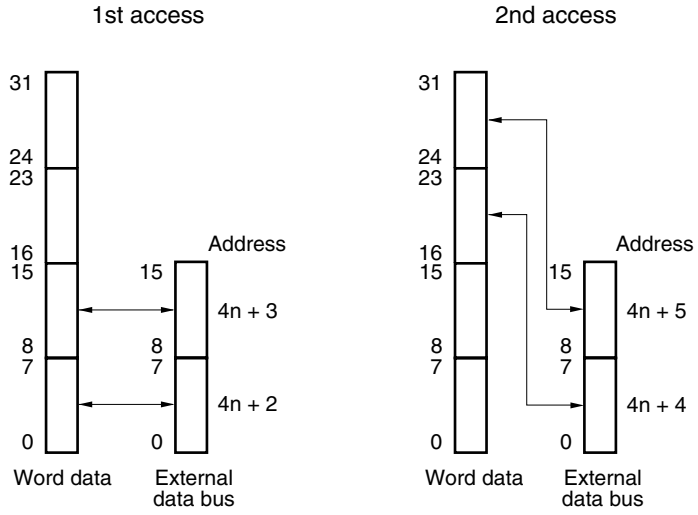
<1> Access to address (4n)



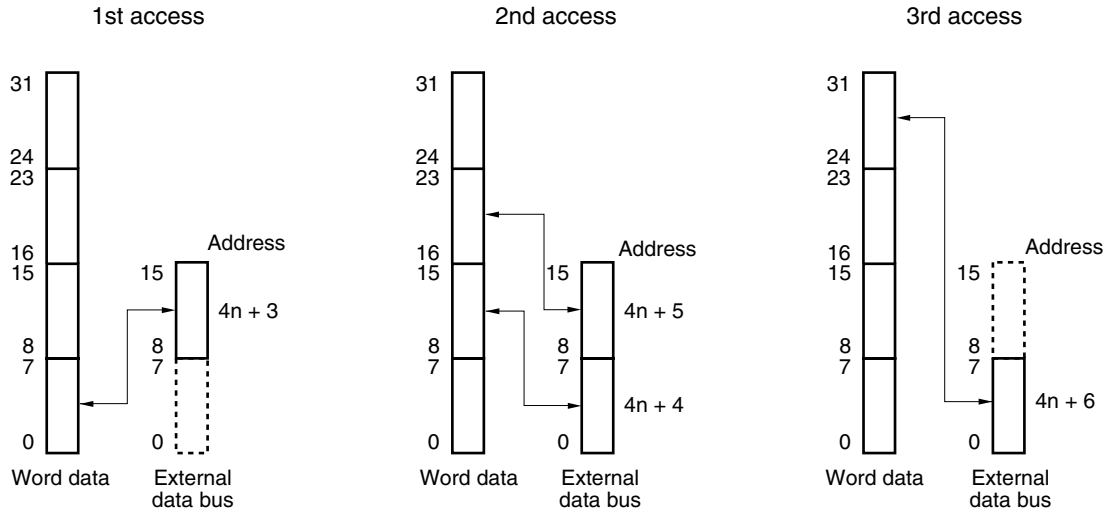
<2> Access to address (4n+1)



<3> Access to address (4n+2)

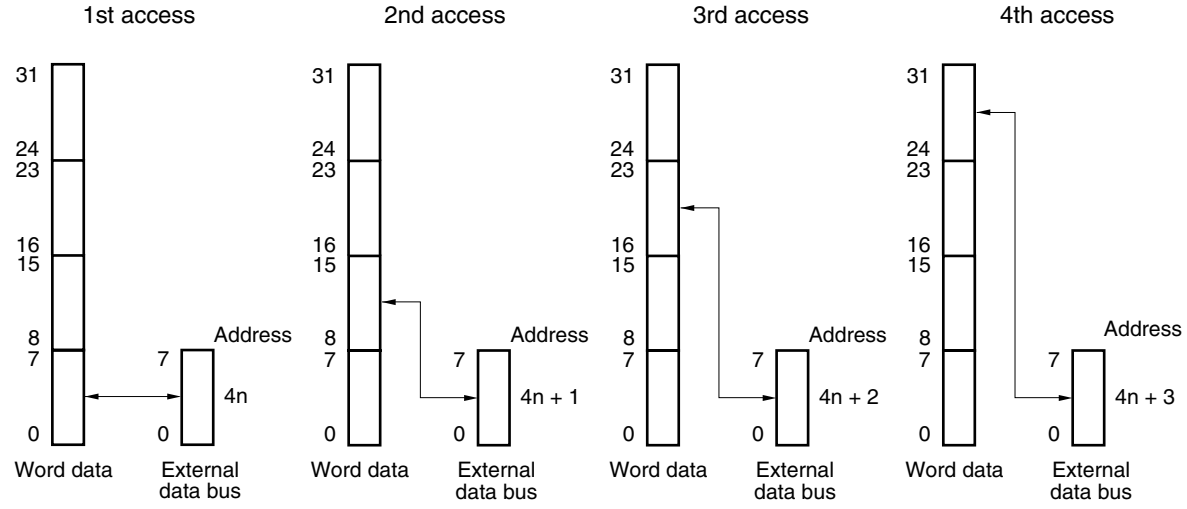


<4> Access to address (4n+3)

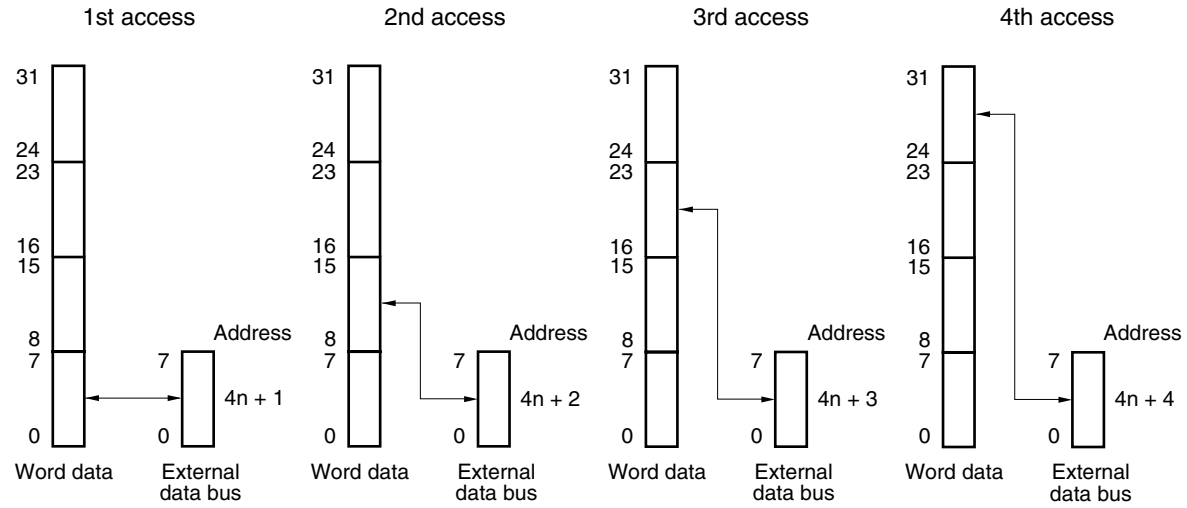


(c) When the data bus width is 8 bits (Little Endian)

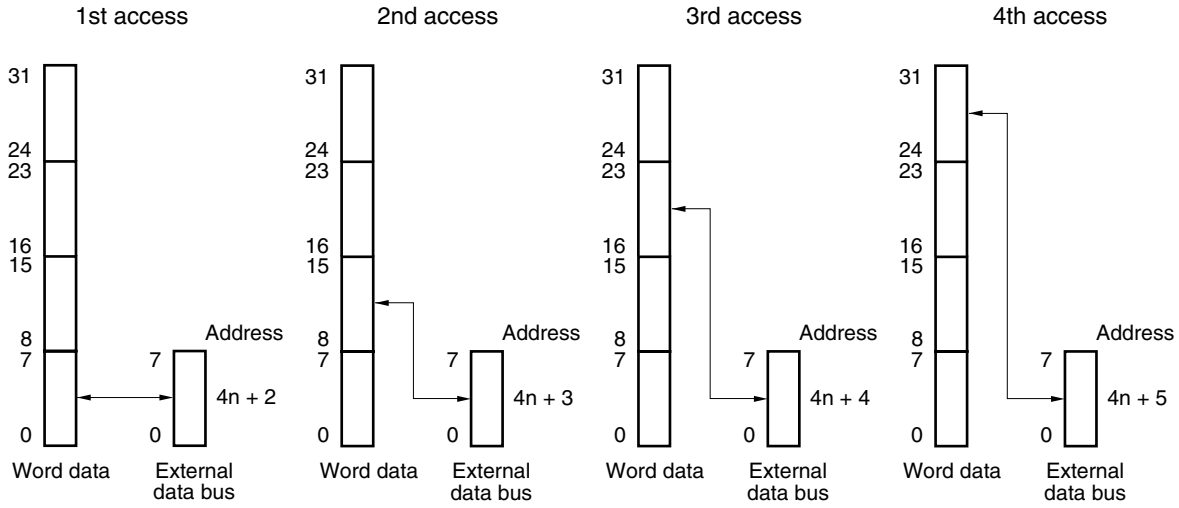
<1> Access to address (4n)



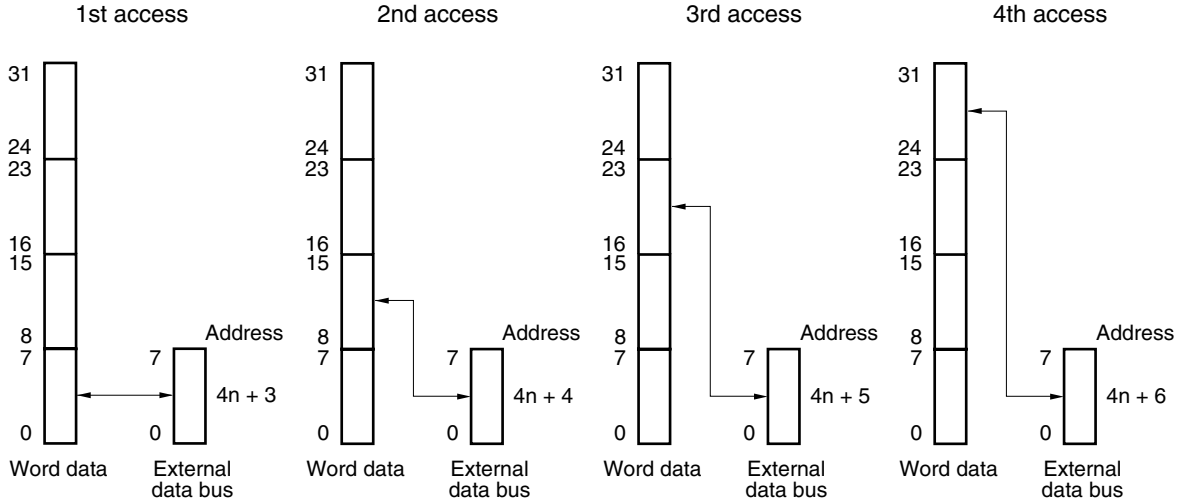
<2> Access to address (4n+1)



<3> Access to address (4n+2)

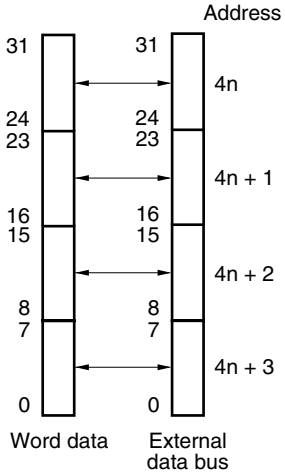


<4> Access to address (4n+3)

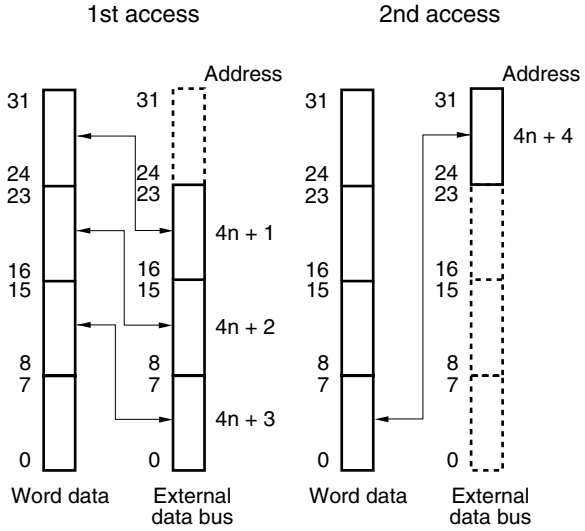


(d) When the data bus width is 32 bits (Big Endian)

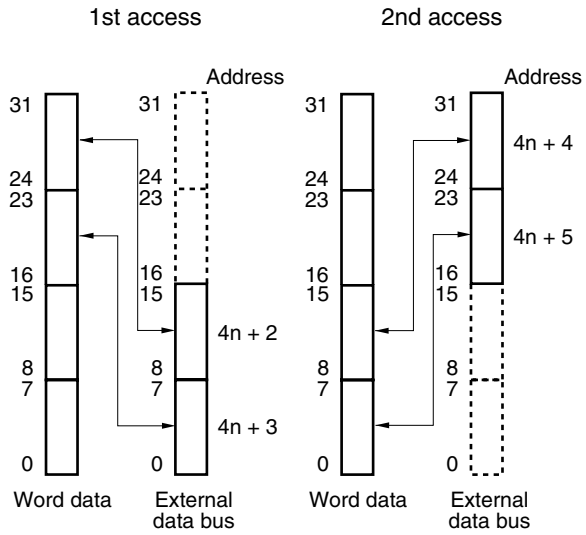
<1> Access to address (4n)



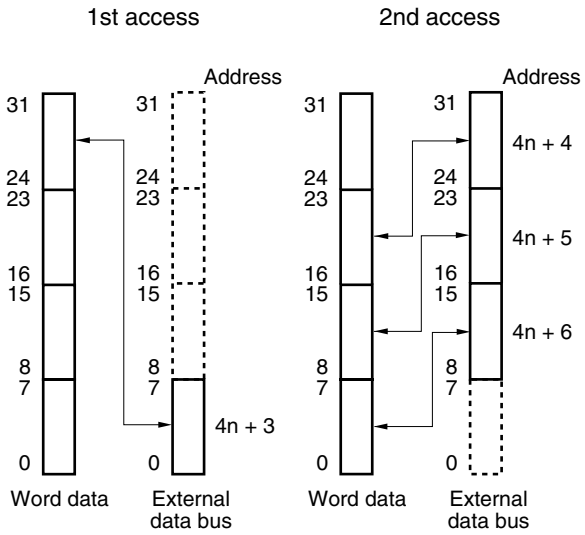
<2> Access to address (4n+1)



<3> Access to address (4n+2)

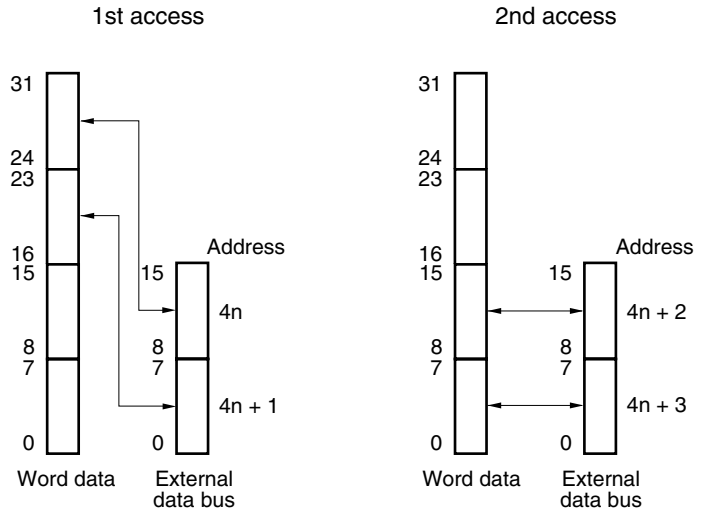


<4> Access to address (4n+3)

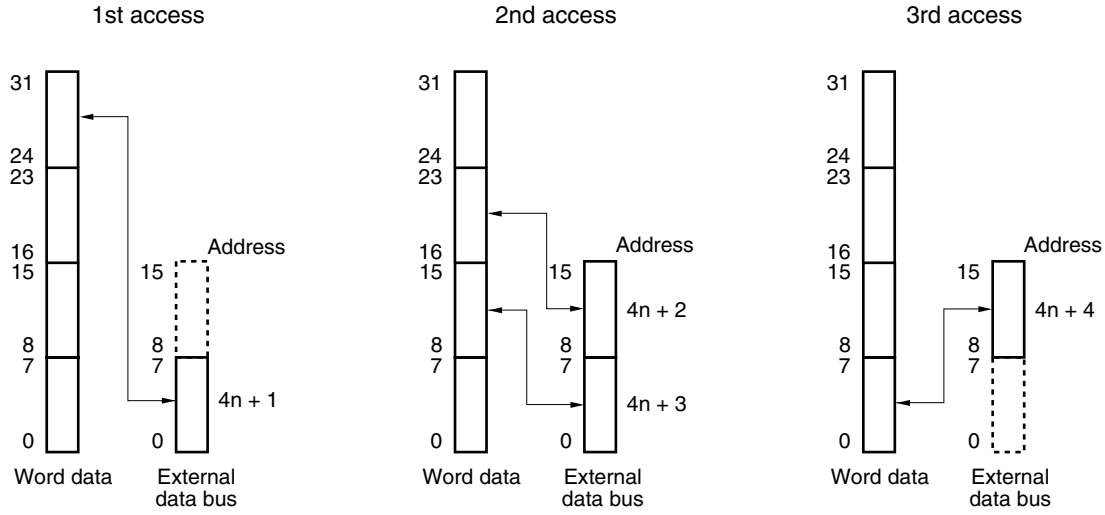


(e) When the data bus width is 16 bits (Big Endian)

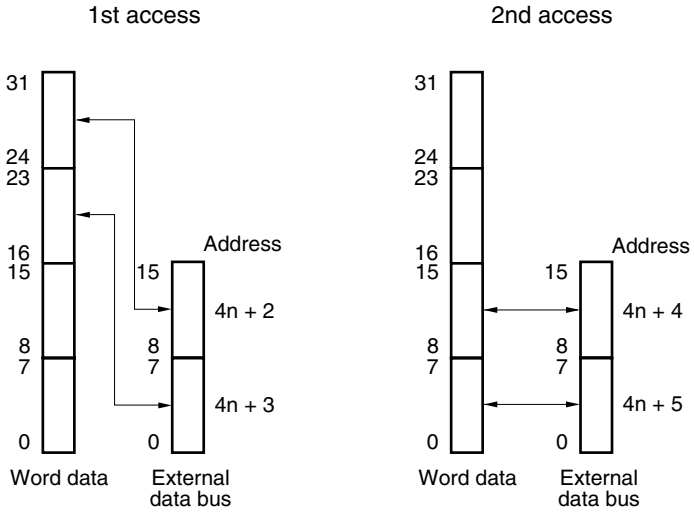
<1> Access to address (4n)



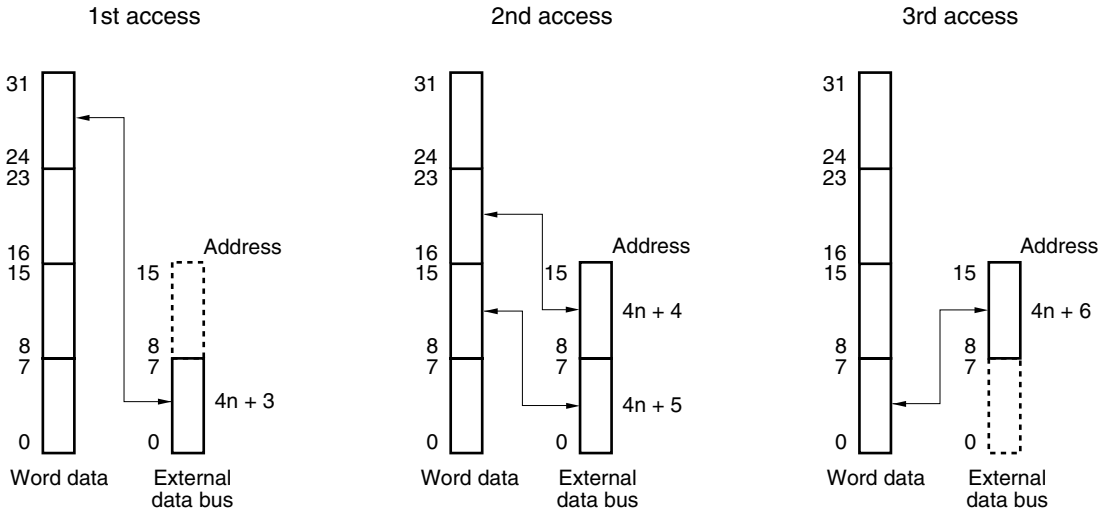
<2> Access to address (4n+1)



<3> Access to address (4n+2)

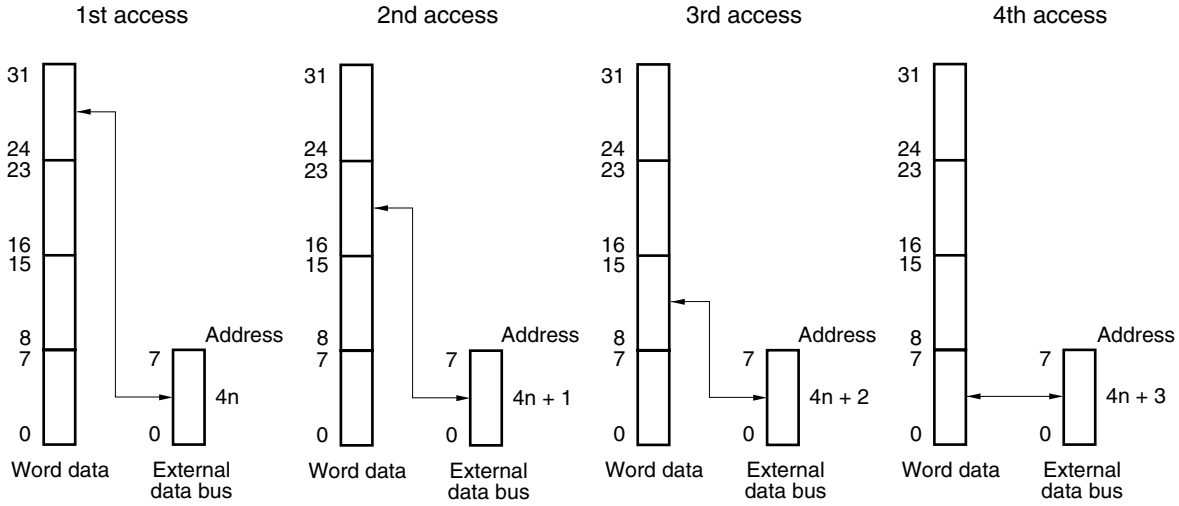


<4> Access to address (4n+3)

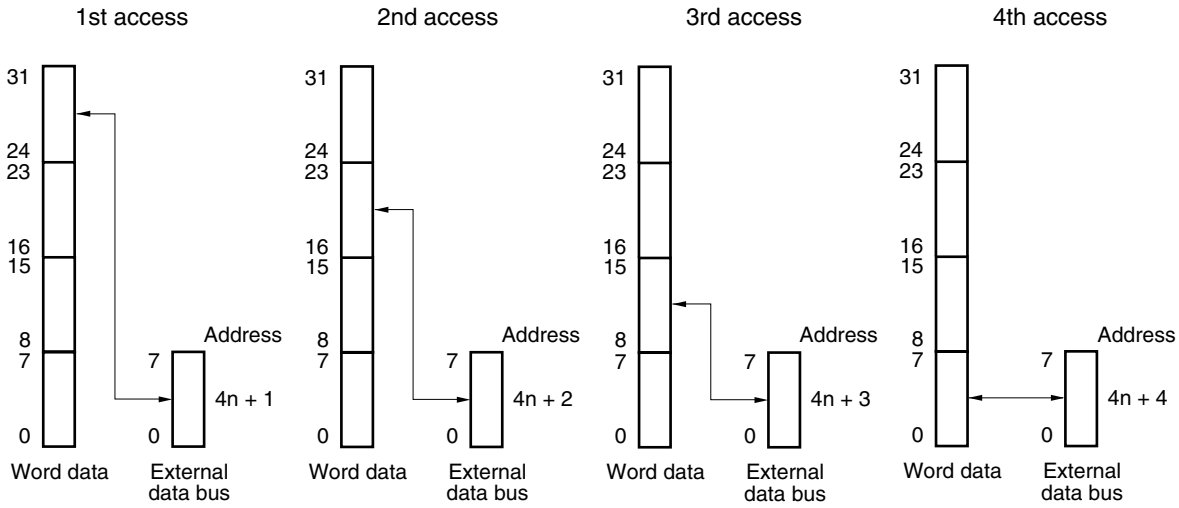


(f) When the data bus width is 8 bits (Big Endian)

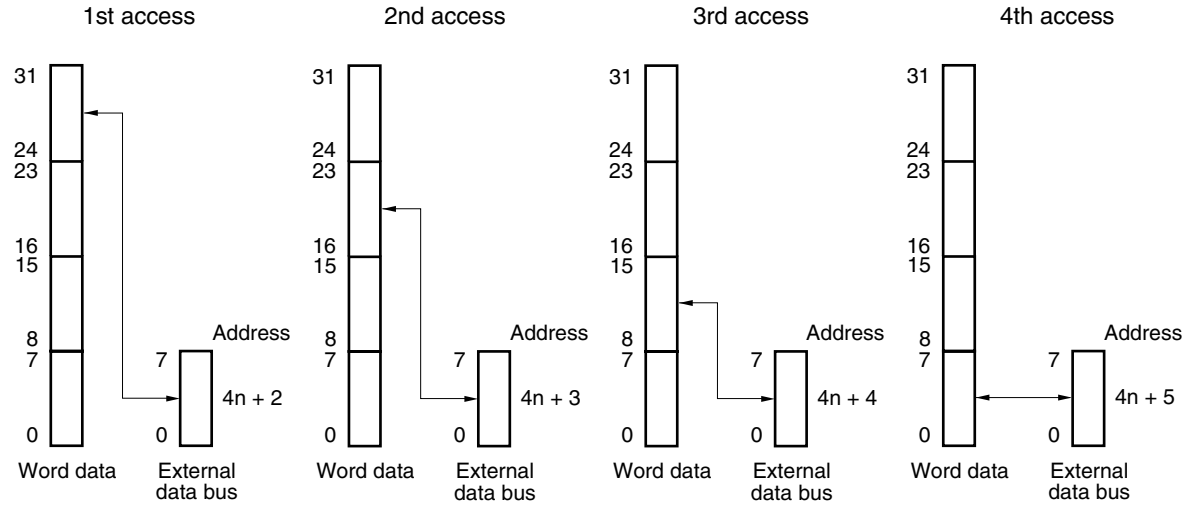
<1> Access to address (4n)



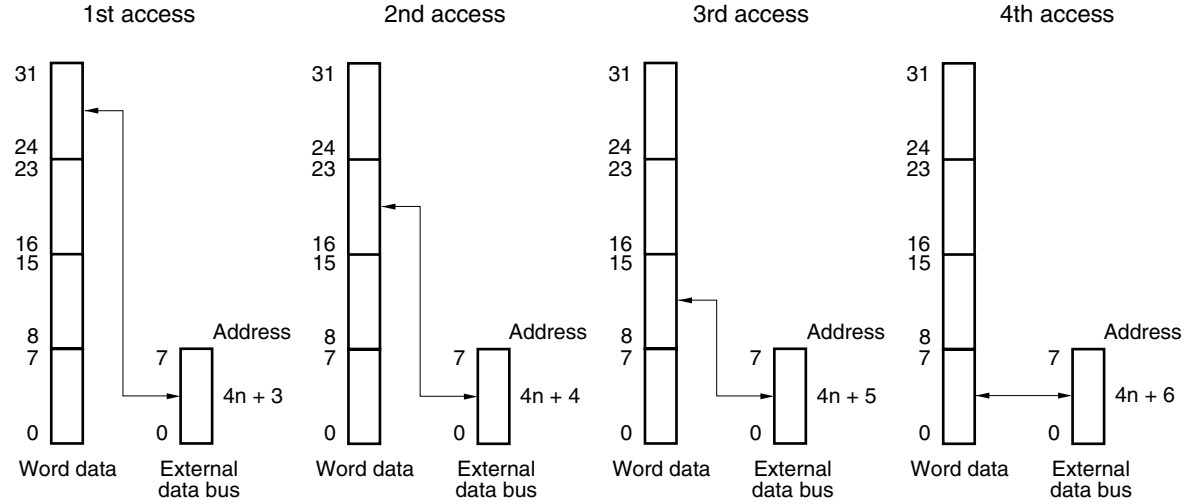
<2> Access to address (4n+1)



<3> Access to address (4n+2)



<4> Access to address (4n+3)



4.6 Wait Function

4.6.1 Programmable wait function

(1) Data wait control registers 0, 1 (DWC0, DWC1)

To facilitate interfacing with low-speed memory or with I/Os, it is possible to insert up to 7 data wait states with respect to the starting bus cycle for each CS area.
The number of wait states can be specified by data wait control registers 0 and 1 (DWC0, DWC1) in programming. Just after system reset, all blocks have 7 data wait states inserted.
These registers can be read/written in 16-bit units.

Cautions: 1. The internal ROM area (flash memory) and the internal RAM area are not subject to programmable waits and ordinarily no wait access is carried out. The internal peripheral I/O area is also not subject to programmable wait states, with wait control performed only by each peripheral function.

2. Write to the DWC0 and DWC1 registers after reset, and then do not change the set values. Also, do not access an external memory area other than that for this initialization routine until initial setting of the DWC0 and DWC1 registers is finished. However, it is possible to access external memory areas whose initialization has been finished.

Figure 4-8: Data Wait Control Registers 0, 1 (DWC0, DWC1) Format

After reset: 7777H R/W Address: FFFFF484H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWC0	0	DW32	DW31	DW30	0	DW22	DW21	DW20	0	DW12	DW11	DW10	0	DW02	DW01	DW00
		$\overline{CS3}$				$\overline{CS2}$				$\overline{CS1}$				$\overline{CS0}$		

After reset: 7777H R/W Address: FFFFF486H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWC1	0	DW72	DW71	DW70	0	DW62	DW61	DW60	0	DW52	DW51	DW50	0	DW42	DW41	DW40
		$\overline{CS7}$				$\overline{CS6}$				$\overline{CS5}$				$\overline{CS4}$		

DWCn2	DWCn1	DWCn0	Number of Inserted Data Wait States During $\overline{CSn}$ Area Access
0	0	0	No wait states inserted
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Remark: n = 0 to 7

(2) Address wait control register (AWC)

The V850E/PH2 allows insertion of address setup wait and address hold wait states before and after the T1 cycle.

The address setup wait and address hold wait states can be set with the AWC register for each CS area.

This register can be read/written in 16-bit units.

Cautions: 1. The internal ROM area (flash memory) and the internal RAM area are not subject to programmable waits and ordinarily no wait access is carried out. The internal peripheral I/O area is also not subject to programmable wait states, with wait control performed only by each peripheral function.

2. Write to the AWC registers after reset, and then do not change the set values. Also, do not access an external memory area other than that for this initialization routine until initial setting of the AWC registers is finished. However, it is possible to access external memory areas whose initialization has been finished.

Figure 4-9: Address Wait Control Register (AWC)

After reset: 0000H R/W Address: FFFFF488H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWC	AHW7	ASW7	AHW6	ASW6	AHW5	ASW5	AHW4	ASW4	AHW3	ASW3	AHW2	ASW2	AHW1	ASW1	AHW0	ASW0
	$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$	

AHWn	Address Hold Wait Insertion During $\overline{CSn}$ Area Access
0	No Insertion
1	Address hold wait state inserted after T1 bus cycle

ASW1	Address Setup Wait Insertion During $\overline{CSn}$ Area Access
0	No Insertion
1	Address setup wait state inserted before T1 bus cycle

Remark: n = 0 to 7

4.7 Idle State Insertion Function

To facilitate interfacing with low-speed memory devices, an idle state (TI) can be inserted into the current bus cycle after the T2 state to meet the data output float delay time (tdf) on memory read access for each CS space. The bus cycle following the T2 state starts after the idle state is inserted.

An idle state is inserted after read/write cycles for SRAM, external I/O, or external ROM.

In the following cases, an idle state is inserted in the timing.

- after read/write cycles for SRAM, external I/O, or external ROM

The idle state insertion setting can be specified by program using the bus cycle control register (BCC) and the bus clock dividing control register (DVC).
Immediately after the system reset, idle state insertion is automatically programmed for all memory blocks on read access.

(1) Bus cycle control register (BCC)

This register can be read/written in 16-bit units.
Reset input changes the value of this register to initial setting AAAAH.

Cautions: 1. Idle states cannot be inserted in internal ROM, internal RAM, on-chip peripheral I/O, or programmable peripheral I/O areas.

2. Write to the BCC register after reset, and then do not change the set value. Also, do not access an external memory area other than that for this initialization routine until initial setting of the BCC register is finished. However, it is possible to access external memory areas whose initialization has been finished.

3. Do not change the settings of bits that are 0 after reset. Otherwise the operation of the external bus interface cannot be ensured.

Figure 4-10: Bus Cycle Control Register (BCC)

After reset: AAAAH R/W Address: FFFFF48AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCC	BC71	0	BC61	0	BC51	0	BC41	0	BC31	0	BC21	0	BC11	0	BC01	0

BCn1	Idle State Insertion During $\overline{CSn}$ Area Access
0	No insertion
1	Idle state inserted
Remark:	When bit BCn1 bit is set to “1”, an idle state will be inserted after any read access. If an idle state after write access is necessary, the BCWI bit of the DVC register has to be set additionally.

Remark: n = 0 to 7

(2) Bus clock dividing control register (DVC)

This register can be read/written in 8-bit units.
Reset input changes the value of this register to initial setting 01H.

Cautions: 1. Idle states cannot be inserted in internal ROM, internal RAM, on-chip peripheral I/O, or programmable peripheral I/O areas.

2. Write to the DVC register after reset, and then do not change the set value. Also, do not access an external memory area other than that for this initialization routine until initial setting of the DVC register is finished. However, it is possible to access external memory areas whose initialization has been finished.

3. Do not change the settings of bits 0 to 6. Otherwise the operation of the external bus interface cannot be ensured.

Figure 4-11: Bus Clock Dividing Control Register (DVC)

After reset: 01H R/W Address: FFFFF48EH

	7	6	5	4	3	2	1	0
DVC	BCWI	0	0	0	0	0	0	1

BCWI	Idle State Insertion after Write Cycle
0	Idle state not inserted after write access
1	Idle state inserted after write access[Note]

Note: BCWI bit setting is only valid when BCn1 bit of the BCC register, corresponding to the $\overline{CSn}$ area for which the write access will be performed, is set to "1". (n = 0 to 7)

4.8 Bus Priority Order

There are two external bus cycles: operand data access and instruction fetch.
As for the priority order, the highest priority has the instruction fetch than operand data access.
An instruction fetch may be inserted between read access and write access during read modify write access.
Also, an instruction fetch may be inserted between bus access and bus access during CPU bus clock.

Table 4-2: Bus Priority Order

Priority Order	External Bus Cycle	Bus Master
Low ↕ High	Operand data access	CPU
	Instruction fetch	CPU

4.9 Boundary Operation Conditions

4.9.1 Program space

Branching to the on-chip peripheral I/O area is prohibited. If the above is performed, undefined data is fetched, and fetching from the external memory is not performed.

4.9.2 Data space

The V850E/PH2 is provided with an address misalign function.
Through this function, regardless of the data format (word, halfword or byte), data can be allocated to all addresses. However, in the case of word data and halfword data, if the data is not subject to boundary alignment, the bus cycle will be generated at least 2 times and bus efficiency will drop.

(1) External bus width: 16 bits

(a) In the case of halfword-length data access

When the address's LSB is 1, a byte-length bus cycle will be generated 2 times.

(b) In the case of word-length data access

- When the address's LSB is 1, bus cycles will be generated in the order of byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle.
- When the address's lower 2 bits are 10B, a halfword-length bus cycle will be generated 2 times.

(2) External bus width: 32 bits

(a) In the case of halfword-length data access

When the address's lower 2 bits are 11B, a byte-length bus cycle will be generated 2 times.

(b) In the case of word-length data access

When the address's lower 2 bits are 10B, a halfword-length bus cycle will be generated 2 times.

Chapter 5 Memory Access Control Function

5.1 SRAM, External ROM, External I/O Interface

5.1.1 Features

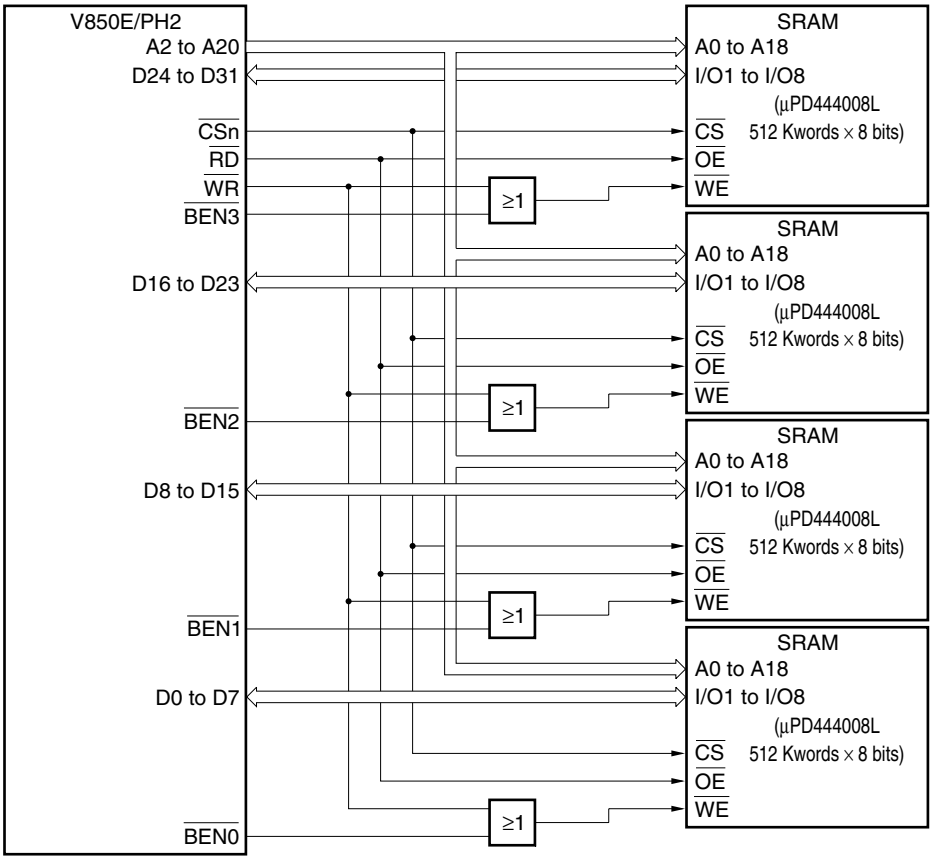
- SRAM is accessed in a minimum of 2 states.
- Up to 7 states of programmable data waits can be inserted by setting the DWC0 and DWC1 registers.
- Data wait can be controlled via $\overline{\text{WAIT}}$ pin input.
- An idle state can be inserted after a read/write cycle by setting the BCC and DVC registers.
- An address setup wait state and an address hold state can be inserted by setting the ASC register.

5.1.2 SRAM connection

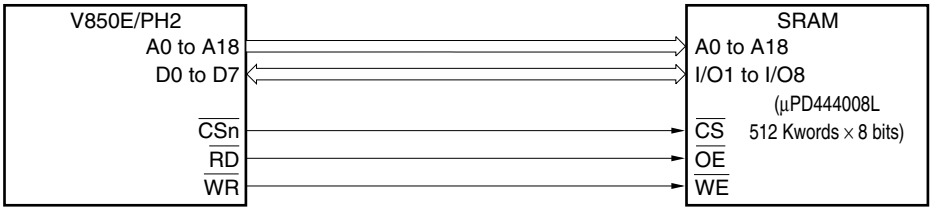
Examples of connection to SRAM are shown below.

Figure 5-1: Examples of Connection to SRAM (1/2)

(a) When Data Bus Width is 32 Bits and Data Size of SRAM is 8 Bits



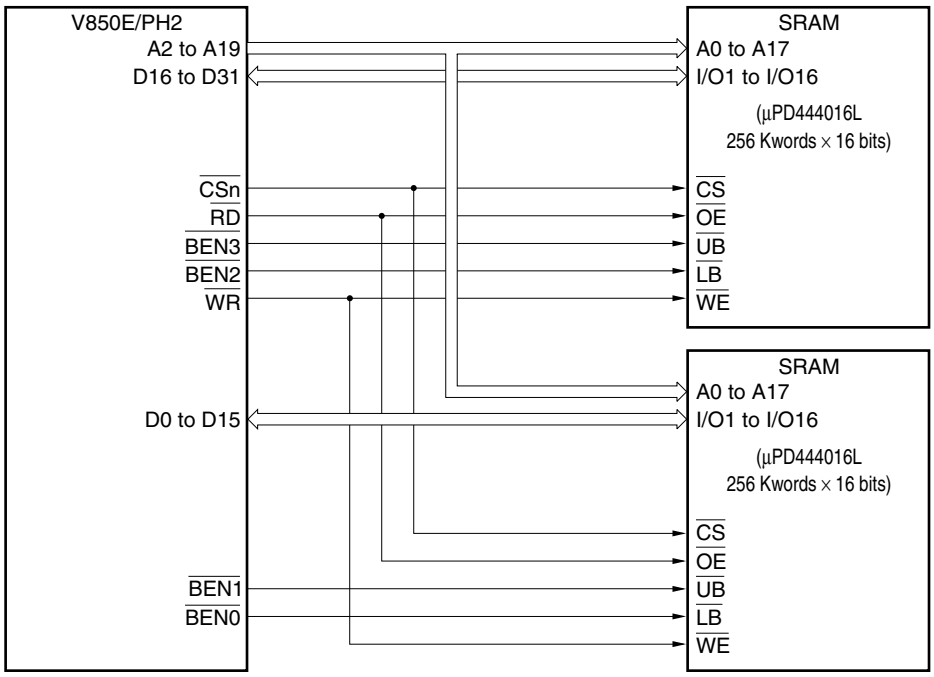
(b) When Data Bus Width is 8 Bits and Data Size of SRAM is 8 Bits



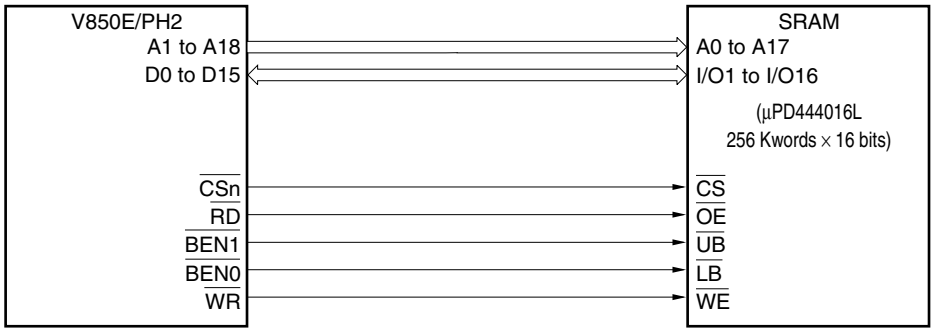
Remark: n = 0, 1, 3, 4

Figure 5-1: Examples of Connection to SRAM (2/2)

(c) When Data Bus Width is 32 Bits and Data Size of SRAM is 16 Bits



(d) When Data Bus Width is 16 Bits and Data Size of SRAM is 16 Bits

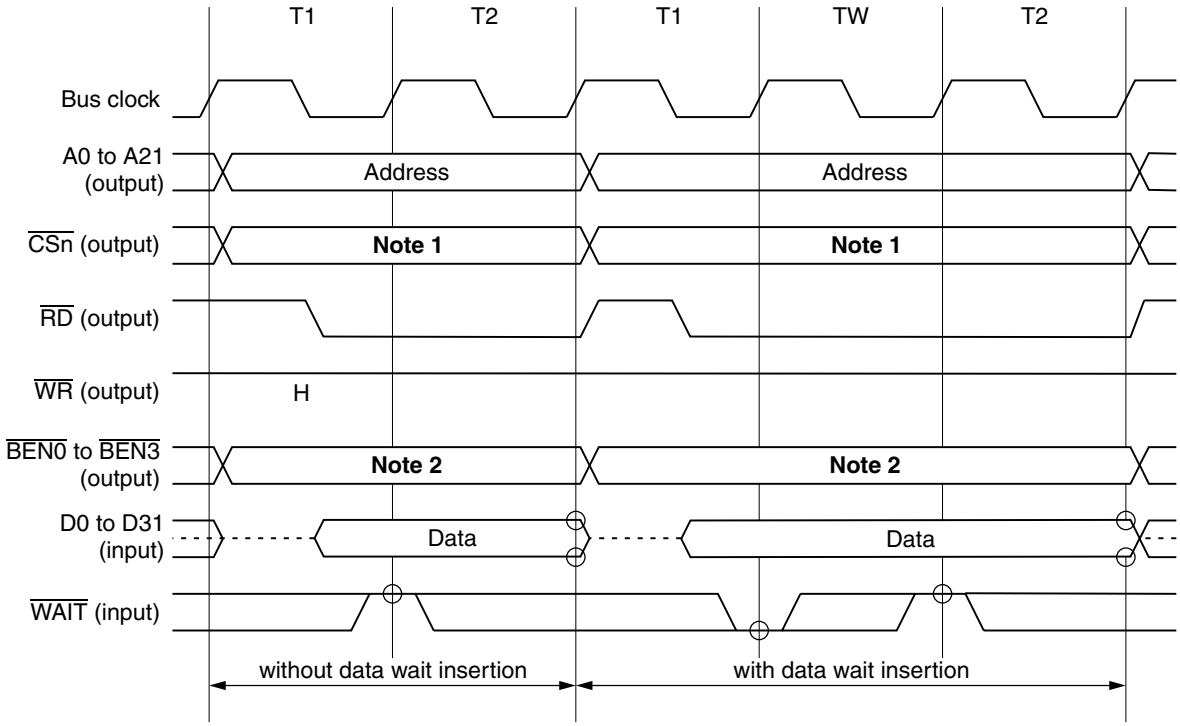


Remark: n = 0, 1, 3, 4

5.1.3 SRAM, external ROM, external I/O access

Figure 5-2: SRAM, External ROM, External I/O Access Timing (1/8)

(a) Read



Notes: 1. $\overline{\text{CSn}}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.

2. $\overline{\text{BEN0}}$ to $\overline{\text{BEN3}}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks: 1. n = 0, 1, 3, 4

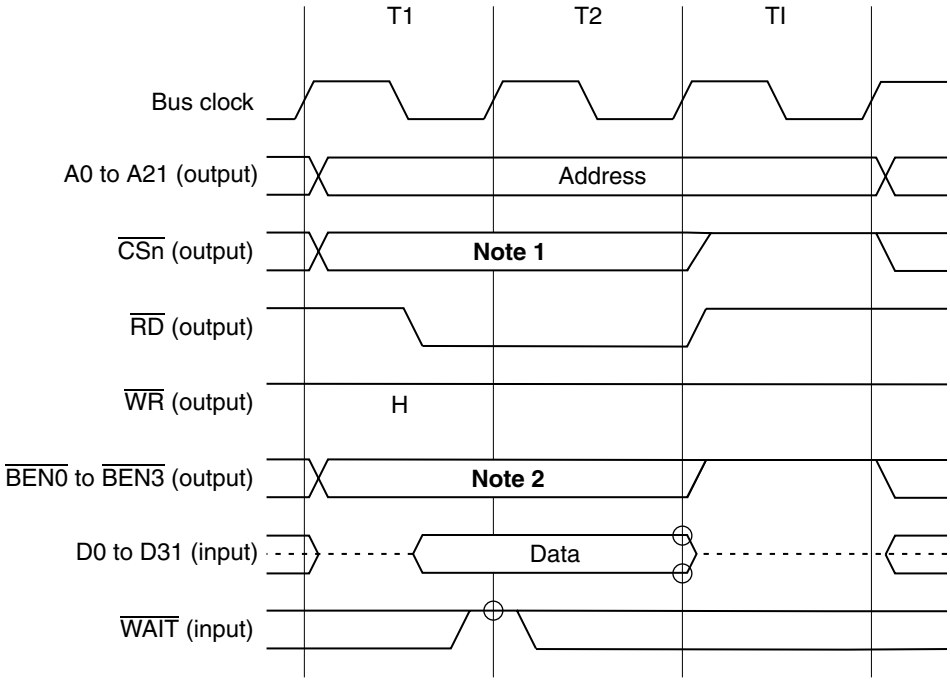
2. Bus clock = $f_{XX}/2$

3. The circle indicates the sampling timing.

4. The dashed line indicates the high impedance state.

Figure 5-2: SRAM, External ROM, External I/O Access Timing (2/8)

(b) Read (Idle State Inserted)



Notes: 1. $\overline{\text{CSn}}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.

2. $\overline{\text{BEN0}}$ to $\overline{\text{BEN3}}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks: 1. n = 0, 1, 3, 4

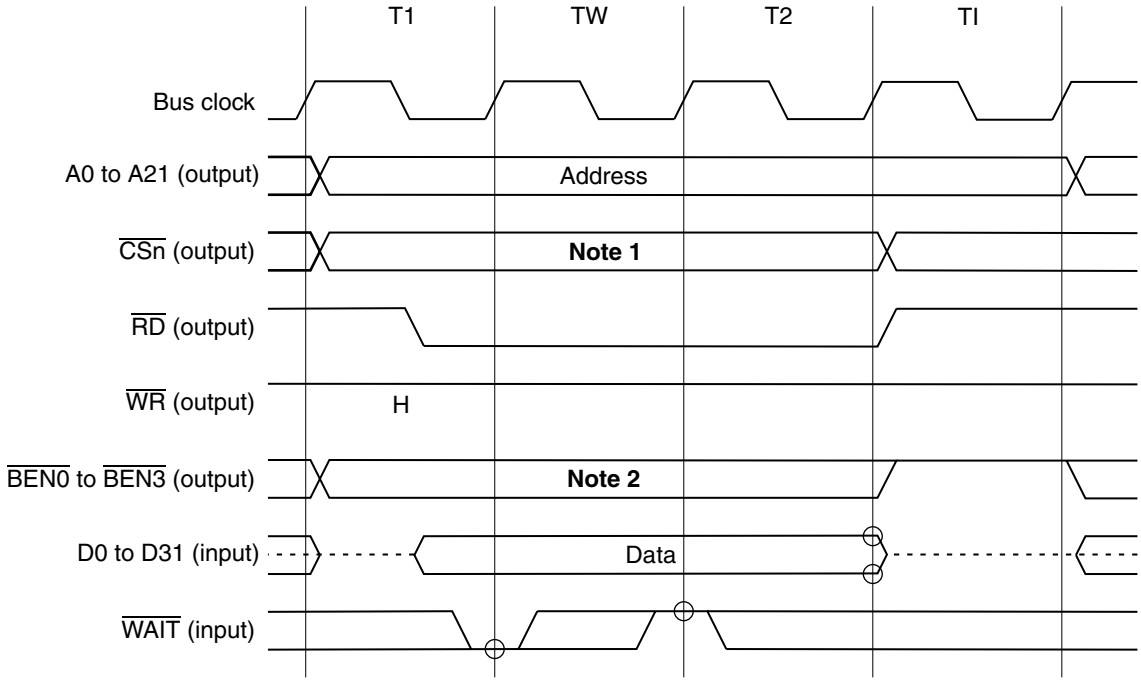
2. Bus clock = $f_{XX}/2$

3. The circle indicates the sampling timing.

4. The dashed line indicates the high impedance state.

Figure 5-2: SRAM, External ROM, External I/O Access Timing (3/8)

(c) Read (Data Wait, Idle State Inserted)



Notes: 1. $\overline{\text{CSn}}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.

2. $\overline{\text{BEN0}}$ to $\overline{\text{BEN3}}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks: 1. n = 0, 1, 3, 4

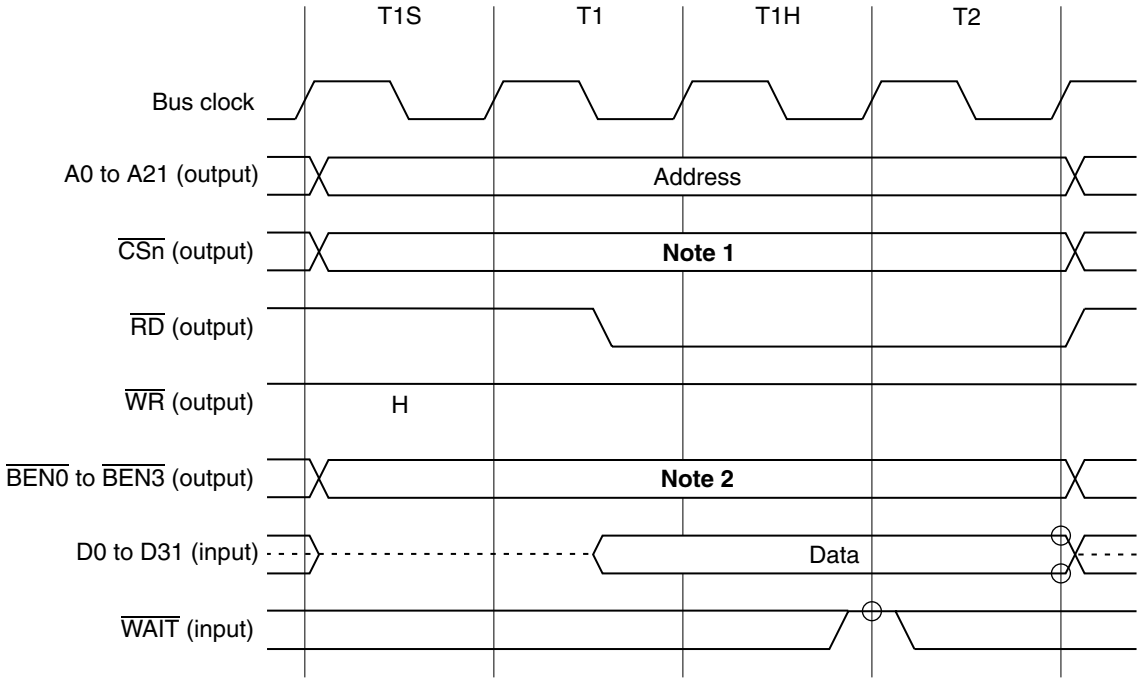
2. Bus clock = $f_{XX}/2$

3. The circle indicates the sampling timing.

4. The dashed line indicates the high impedance state.

Figure 5-2: SRAM, External ROM, External I/O Access Timing (4/8)

(d) Read (Address Setup Wait and Address Hold Wait State Inserted)



Notes: 1. $\overline{\text{CSn}}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.

2. $\overline{\text{BEN0}}$ to $\overline{\text{BEN3}}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks: 1. n = 0, 1, 3, 4

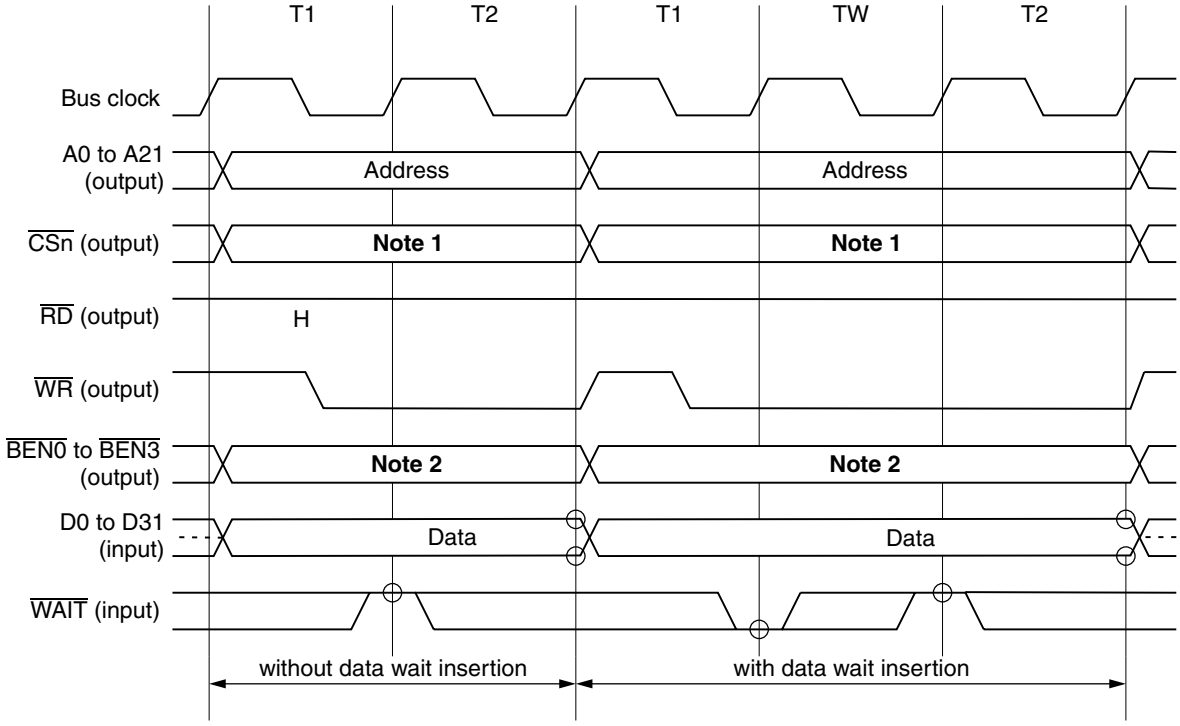
2. Bus clock = $f_{XX}/2$

3. The circle indicates the sampling timing.

4. The dashed line indicates the high impedance state.

Figure 5-2: SRAM, External ROM, External I/O Access Timing (5/8)

(e) Write



Notes: 1. $\overline{\text{CSn}}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.

2. $\overline{\text{BEN0}}$ to $\overline{\text{BEN3}}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks: 1. n = 0, 1, 3, 4

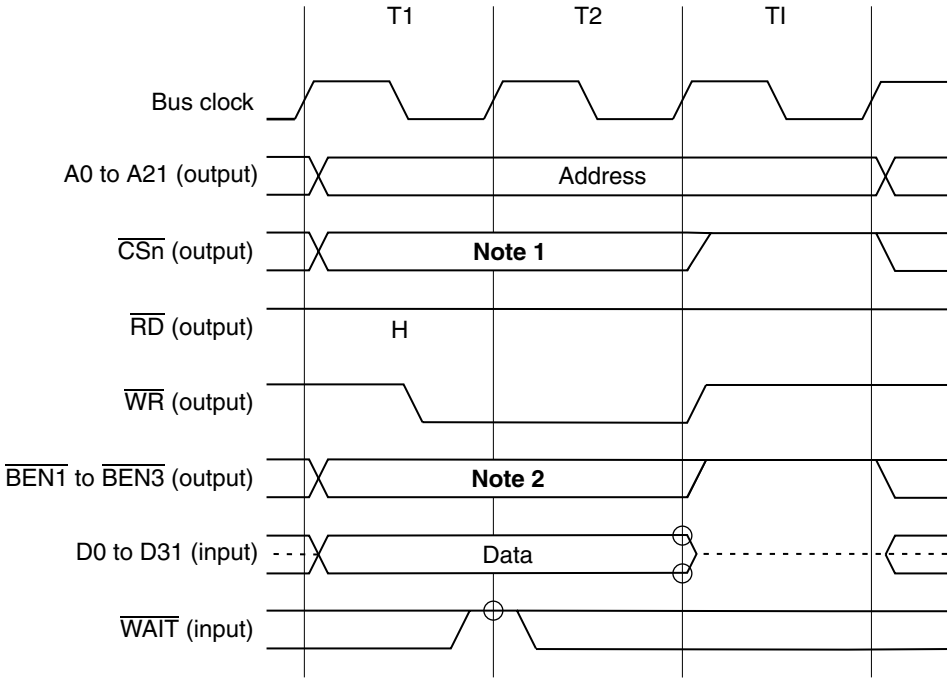
2. Bus clock = $f_{XX}/2$

3. The circle indicates the sampling timing.

4. The dashed line indicates the high impedance state.

Figure 5-2: SRAM, External ROM, External I/O Access Timing (6/8)

(f) Write (Idle State Inserted)



Notes: 1. CSn output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.

2. BEN0 to BEN3 output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks: 1. n = 0, 1, 3, 4

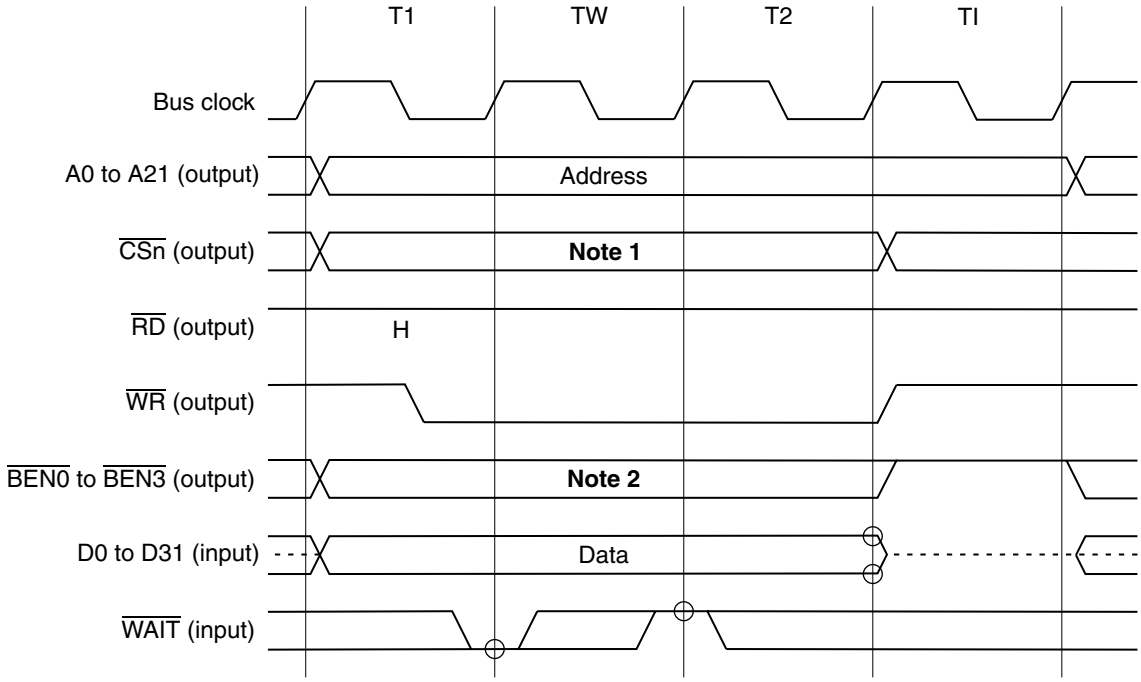
2. Bus clock = $f_{XX}/2$

3. The circle indicates the sampling timing.

4. The dashed line indicates the high impedance state.

Figure 5-2: SRAM, External ROM, External I/O Access Timing (7/8)

(g) Write (Data Wait, Idle State Inserted)



Notes: 1. $\overline{\text{CSn}}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.

2. $\overline{\text{BEN0}}$ to $\overline{\text{BEN3}}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks: 1. n = 0, 1, 3, 4

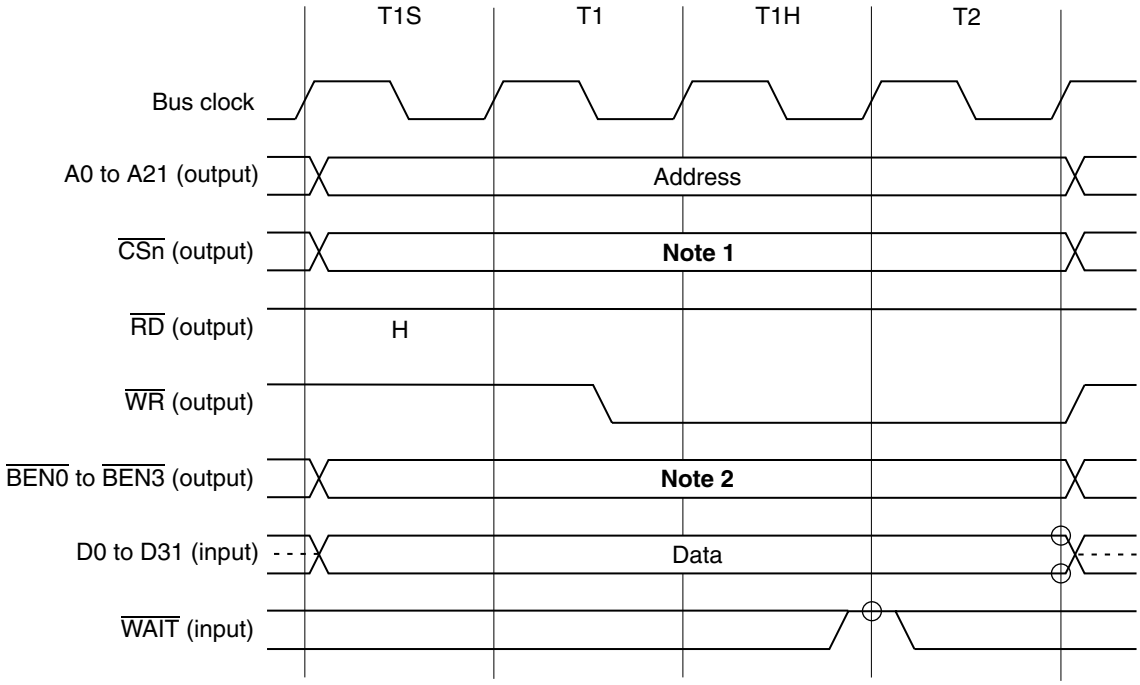
2. Bus clock = $f_{XX}/2$

3. The circle indicates the sampling timing.

4. The dashed line indicates the high impedance state.

Figure 5-2: SRAM, External ROM, External I/O Access Timing (8/8)

(h) Read (Address Setup Wait and Address Hold Wait State Inserted)



Notes: 1. CSn output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.

2. BEN0 to BEN3 output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks: 1. n = 0, 1, 3, 4

2. Bus clock = $f_{XX}/2$

3. The circle indicates the sampling timing.

4. The dashed line indicates the high impedance state.

[MEMO]

Chapter 6 DMA Functions (DMA Controller)

6.1 Features

The V850E/PH2 includes a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer.
The DMAC controls data transfer between internal RAM (iRAM) and peripheral I/O registers, based on DMA requests issued by the on-chip peripheral I/O (A/D converters, inverter timers, and serial interfaces), with the following features.

- 2 channels for DMA transfer from A/D converter (ADC0, ADC1)
  - Transfer object: I/O →iRAM
  - Transfer size: 16 bits
  - Dedicated transfer channels for ADC0 and ADC1

- 2 channels for DMA transfer to PWM timer (TMR0, TMR1)
  - Transfer object: iRAM →I/O
  - Transfer size: 16 bits
  - Dedicated transfer channels for TMR0 and TMR1

- 2 channels for DMA transfer from serial interfaces on reception completion
  - Transfer object: I/O →iRAM
  - Transfer size: 8 or 16 bits
  - DMA request for each channel selectable
    Clocked serial interfaces: CSIB0, CSIB1, CSI30, CSI31
    Asynchronous serial interface: UARTC0, UARTC1

- 2 channels for DMA transfer to serial interfaces on transmission repetition
  - Transfer object: iRAM →I/O
  - Transfer size: 8 or 16 bits
  - DMA request for each channel selectable
    Clocked serial interfaces: CSIB0, CSIB1, CS30, CSI31
    Asynchronous serial interface: UARTC0, UARTC1

- Up to 256 transfer counts for each channel.

6.2 Control Registers

(1) DMA transfer memory start address registers 0 to 7 (MAR0 to MAR7)

The MARn register specifies the subordinated 16 bits of the DMA transfer start address within the internal RAM area for the DMA channel n (n = 0 to 7).
This register can be read or written in 16-bit units.
After reset the register content is undefined.

Figure 6-1: DMA Transfer Memory Start Address Registers 0 to 7 (MAR0 to MAR7)

After reset: undefined R/W Address: MAR0 FFFFF300H, MAR1 FFFFF302H
MAR2 FFFFF304H, MAR3 FFFFF306H
MAR4 FFFFF308H, MAR5 FFFFF30AH
MAR6 FFFFF30CH, MAR7 FFFFF30EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MARn (n = 0 to 7)	MARn 15	MARn 14	MARn 13	MARn 12	MARn 11	MARn 10	MARn 9	MARn 8	MARn 7	MARn 6	MARn 5	MARn 4	MARn 3	MARn 2	MARn 1	MARn 0

Cautions: 1. Since the internal RAM area is mapped between 3FF0000H and 3FF7FFFH the value written to the MARn register has to be in the range from 0000H to 7FFFH.

2. The value set to the MARn register is increased by each DMA transfer of channels. It does not keep the initial value after the DMA transfer ends.

(2) DMA transfer SFR start address registers 2, 3 (SAR2, SAR3)

The SARn register specifies the start address of the TMR register for which the DMA transfer is started on DMA channel n (n = 2, 3).
This register can be read or written in 8-bit units.
After reset the register content is undefined.

Figure 6-2: DMA Transfer SFR Start Address Registers 2, 3 (SAR2, SAR3)

After reset: undefined R/W Address: SAR2 FFFFF314H, SAR3 FFFFF316H

	7	6	5	4	3	2	1	0
SARn (n = 2, 3)	0	0	0	0	0	SARn2	SARn1	SARn0

SARn2	SARn1	SARn0	DMA Transfer Start Address of TMR Reload Register			
			n = 2		n = 3	
			Register	Address	Register	Address
0	0	0	TR0CCR5	FFFFF590H	TR1CCR5	FFFFF5D0H
0	0	1	TR0CCR4	FFFFF592H	TR1CCR4	FFFFF5D2H
0	1	0	_Note 1	FFFFF594H	_Note 1	FFFFF5D4H
0	1	1	_Note 2	FFFFF596H	_Note 2	FFFFF5D6H
1	0	0	TR0CCR0	FFFFF598H	TR1CCR0	FFFFF5D8H
1	0	1	TR0CCR3	FFFFF59AH	TR1CCR3	FFFFF5DAH
1	1	0	TR0CCR2	FFFFF59CH	TR1CCR2	FFFFF5DCH
1	1	1	TR0CCR1	FFFFF59EH	TR1CCR1	FFFFF5DEH

Notes: 1. Although the register address is meaningless, a transfer to this address is always performed when SARn2 to SARn0 bits are equal to 010B or less.

2. Although the register address is meaningless, a transfer to this address is always performed when SARn2 to SARn0 bits are equal to 011B or less.

Caution: During DMA transfer (DEn = 1) the contents of the SARn register may change. After each DMA transfer the contents is incremented by 1 until the final value (07H) is reached.
When the SARn register contents becomes 07H, the initial set value is reloaded.

(3) DMA transfer count registers 0 to 7 (DTCR0 to DTCR7)

The DTCRn register is an 8-bit register that set the transfer count for DMA channel n and stores the remaining transfer count during DMA transfer (n = 0 to 7).
This register can be read or written in 8-bit units.
After reset the register content is undefined.

Figure 6-3: DMA Transfer Count Registers 0 to 7 (DTCR0 to DTCR7)

After reset: undefined R/W Address: DTCR0 FFFFF320H, DTCR1 FFFFF322H
DTCR2 FFFFF324H, DTCR3 FFFFF326H
DTCR4 FFFFF328H, DTCR5 FFFFF32AH
DTCR6 FFFFF32CH, DTCR7 FFFFF32EH

	7	6	5	4	3	2	1	0
DTCRn	DTCRn7	DTCRn6	DTCRn5	DTCRn4	DTCRn3	DTCRn2	DTCRn1	DTCRn0

DTCRn 7	DTCRn 6	DTCRn 5	DTCRn 4	DTCRn 3	DTCRn 2	DTCRn 1	DTCRn 0	Remaining DMA Transfer Counts
0	0	0	0	0	0	0	0	256
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	1	1	3
				. . .				. . .
1	1	1	1	1	1	1	0	254
1	1	1	1	1	1	1	1	255

Cautions: 1. The value set to the DTCRn register is decreased by each DMA transfer of channel n. It does not keep the initial value after the DMA transfer ends. Therefore, after DMA transfer end the DTCRn register values becomes 00H.

2. A DMA request becomes only effective after the DTCRn register was written. Even if 00H (means a transfer count of 256) is the initial value, the DTRCn register must be rewritten in order to enable a new DMA transfer.

Remark: n = 0 to 7

(4) DMA mode control register (DMAMC)

The DMAMC register is an 8-bit register that controls the operation of the DMA channels.
This register can be read or written in 8-bit units.
Reset input clears this register to 00H.

Figure 6-4: DMA Mode Control Register (DMAMC)

After reset: 00H R/W Address: DMAMC FFFFF330H

	7	6	5	4	3	2	1	0
DMAMC	DE7	DE6	DE5	DE4	DE3	DE2	DE1	DE0

DEn	Control Bit of DMA Channel n
0	DMA transfer operation of channel n disabled
1	DMA transfer operation of channel n enabled

Caution: Writing of the DE1 and DE0 bits is prohibited if the corresponding A/D converter is operating.

(5) DMA status register (DMAS)

The DMAS register is an 8-bit register that displays the transfer status of the DMA channels.
This register can be read or written in 8-bit units.
Reset input clears this register to 00H.

Figure 6-5: DMA Status Register (DMAS)

After reset: 00H R/W Address: DMAMC FFFFF332H

	7	6	5	4	3	2	1	0
DMAS	DMAS7	DMAS6	DMAS5	DMAS4	DMAS3	DMAS2	DMAS1	DMAS0

DMASn	Status Bit of DMA Channel n
0	DMA transfer of channel n is idle or in progress
1	DMA transfer of channel n is completed
• The DMASn bit can be read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. • Since the DMASn bit is not cleared by the DMAC, it has to be cleared by software before DMA transfer is started.	

Remark: n = 0 to 7

(6) DMA data size control register (DMDSC)

The DMADSC register is an 8-bit register that controls the transfer data size of DMA channels 4 to 7. The data size of DMA channels 0 to 3 is fixed, and therefore not selectable.
This register can be read or written in 8-bit units.
Reset input clears this register to 00H.

Figure 6-6: DMA Data Size Control Register (DMDSC)

After reset: 00H R/W Address: DMAMC FFFFF334H

	7	6	5	4	3	2	1	0
DMADSC	DMADSC7	DMADSC6	DMADSC5	DMADSC4	0	0	0	0

DMADSCn	Transfer Data Size of DMA Channel n
0	8 bits
1	16 bits

Remark: n = 4 to 7

(7) DMA trigger factor registers 4 to 7 (DTFR4 to DTFR7)

The DTFRn register is an 8-bit register that controls the DMA transfer start trigger of DMA channel n via interrupt requests from on-chip peripheral I/O (n = 4 to 7).
The interrupt request set by this register serves as DMA transfer start factor.
This register can be read or written in 8-bit units.
Reset input clears this register to 00H.

Cautions: 1. Do not set the same transfer start factor by different DTFRn registers.

2. Do not rewrite the DTFRn register until a started DMA transfer ends (corresponding DTCRn register value is 00H).

3. Write the DTFRn register before setting the corresponding DTCRn register. According to the present transfer start factor in the DTFRn register a DMA might be started when the DTCRn register is written previously.

Figure 6-7: DMA Trigger Factor Registers 4 to 7 (DTFR4 to DTFR7)

After reset: 00H R/W Address: DTFR4 FFFFF348H, DTFR5 FFFFF34AH DTFR6 FFFFF34CH, DTFR7 FFFFF34EH

	7	6	5	4	3	2	1	0
DTFRn	0	0	0	0	0	IFCn2	IFCn1	IFCn0

IFCn2	IFCn1	IFCn0	DMA Transfer Start Factor	
			when n = 4, 5	when n = 6, 7
0	0	0	DMA request from on-chip peripheral I/O disabled	
0	0	1	INTUC0R	INTUC0T
0	1	0	INTUC1R	INTUC1T
0	1	1	INTCB0R	INTCB0T
1	0	0	INTCB1R	INTCB1T
1	0	1	INTC30	INTC30
1	1	0	INTC31	INTC31
1	1	1	Setting prohibited	

Remark: n = 4 to 7

6.3 DMA Channel Priorities

The DMA channel priorities are fixed as follows.

DMA channel 0 > DMA channel 1 > DMA channel 2 >... > DMA channel 7

6.4 DMA Operation

6.4.1 DMA transfer of A/D converter result registers (ADC0, ADC1)

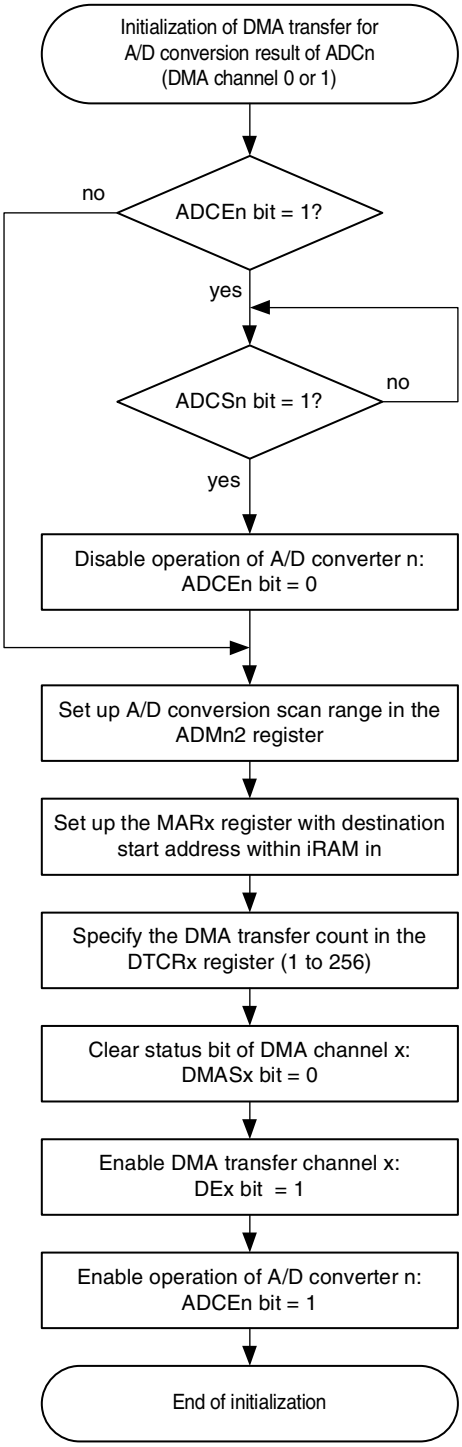
The DMAC has two dedicated channels to support DMA transfer for both A/D converters independently, DMA channel 0 for A/D converter 0 and DMA channel 1 for A/D converter 1. As DMA trigger factor, which requests and starts the DMA transfer, the end of conversion interrupt signal of the corresponding A/D converter is pre-defined (INTADn) (n = 0, 1).

For each DMA trigger the data will be transferred from the A/D conversion result register for DMA (ADDMAn) into the internal RAM specified as destination. While the source transfer address is fixed to the ADDMAn register of the corresponding A/D converter (ADCn), the destination start address can be set up to any even address in the internal RAM.

When the DMA transfer count of a DMA channel terminates, the DMA transfer is stopped and a termination interrupt is generated. The maximum DMA transfer count is 256.

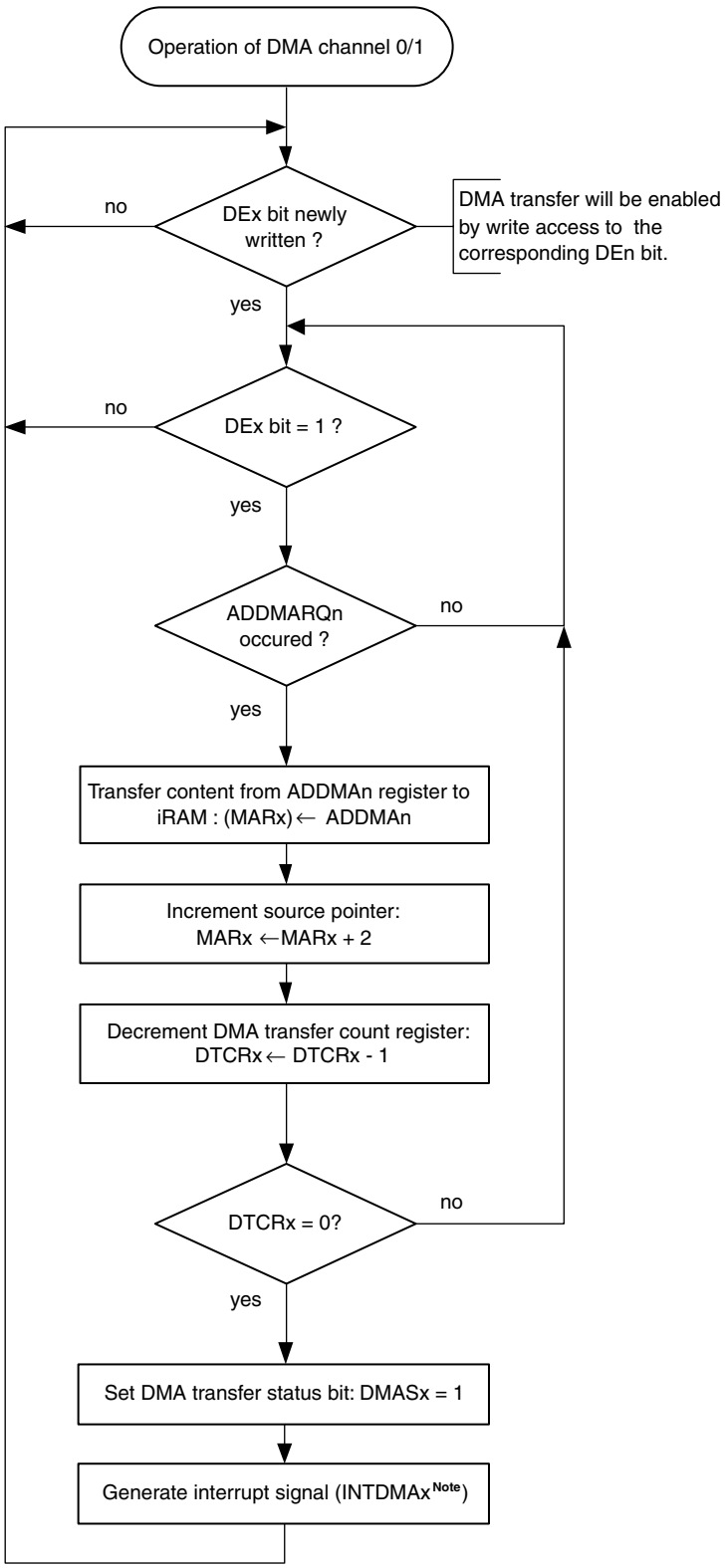
Since the DMA transfer is performed for each finished A/D conversion, it is possible to transfer more than conversion results of one A/D converter scan sequence. However, the user has to take care that the number of transfer counts complies with the product of A/D converter scan area size and the number of A/D converter start triggers.

Figure 6-8: Initialization of DMA Transfer for A/D Conversion Result



Remark: n = 0, 1 (number of ADC channel)
x = n (number of DMA channel)

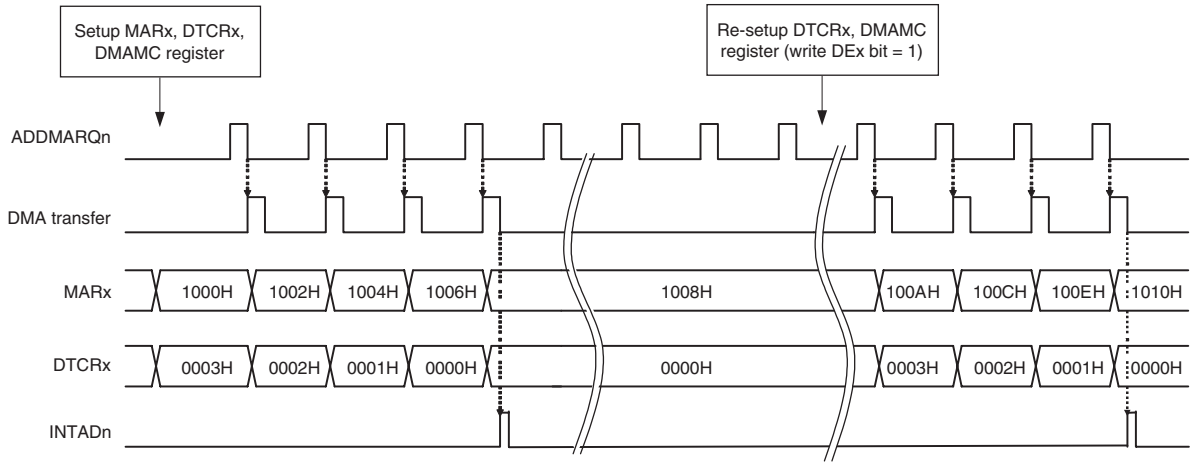
Figure 6-9: Operation of DMA Channel 0/1



Note: DMA transfer completion interrupt has the same interrupt vector address as the corresponding A/D conversion completion interrupt (INTADn), and replaces that interrupt.

Remark: n = 0, 1 (number of ADC channel)
x = n (number of DMA transfer channel)

Figure 6-10: DMA Channel 0 and 1 Trigger Signal Timing



Remarks: 1. The DMA request by ADDMARQ is disregarded after INTDMA is generated, and the DMA transfer is not restarted automatically. Write “1” in the corresponding DEx bit of the DMAMC register again to enable the next transfer of DMA channel x. The DEx bit is not cleared by hardware.

2. n = 0, 1 (number of the A/D converter channel)
x = n (number of the DMA channel)

6.4.2 DMA transfer of PWM timer reload (TMR0, TMR1)

The DMAC has two dedicated channels to support DMA transfer for both PWM timers TMRn independently, DMA channel 2 for TMR0 and DMA channel 3 for TMR1. As DMA trigger factor, which requests and starts the DMA transfer, two corresponding timer interrupt signals are pre-defined (INTTRnOD or INTTRnCD). These are the same signals as for reloading the internal buffer compare registers by the contents of the capture/compare registers TRnCCRm (n = 0, 1)(m = 0 to 5).

For each DMA trigger data will be transferred from internal RAM to the capture/compare registers of corresponding timer TMRn. The destination start address of the TMRn register (TRnCC0, TRnCC2 to TRnCC5) can be set up by the SARx register, as well as the source start address in the internal RAM by the MARx register. The destination end address is always fixed to TRnCC1 register, which also enables the buffer reload in the timer TMRn period (ref. to Table 6-1).

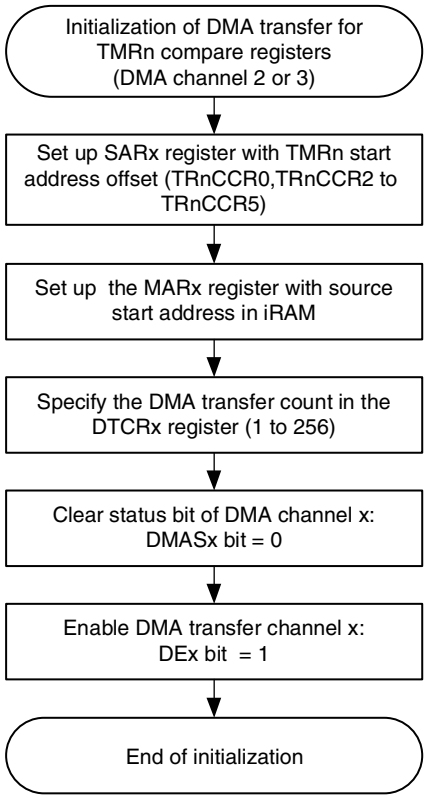
The DMA transfer count is defined by the destination start and end address. However, an additionally DMA trigger count is available, which can be specified in the DTCRx register from 1 to 256. After decrementing the DTCRx register the DMAC will be prepared for a new DMA transfer from internal RAM to the timer TMRn registers until the DMA trigger count terminates (DTCRx register = 0).

Table 6-1: Timer TMR Address Mapping for DMA Transfer

DMA Transfer Destination Address			DMA Transfer Source Address
TMRn registers	Address Offset		
TRnCCR5	00H	Selectable as start address	Any even address in internal RAM area
TRnCCR4	02H		
TRnCCR0	08H		
TRnCCR3	0AH		
TRnCCR2	0CH		
TRnCCR1	0EH	Always end address	

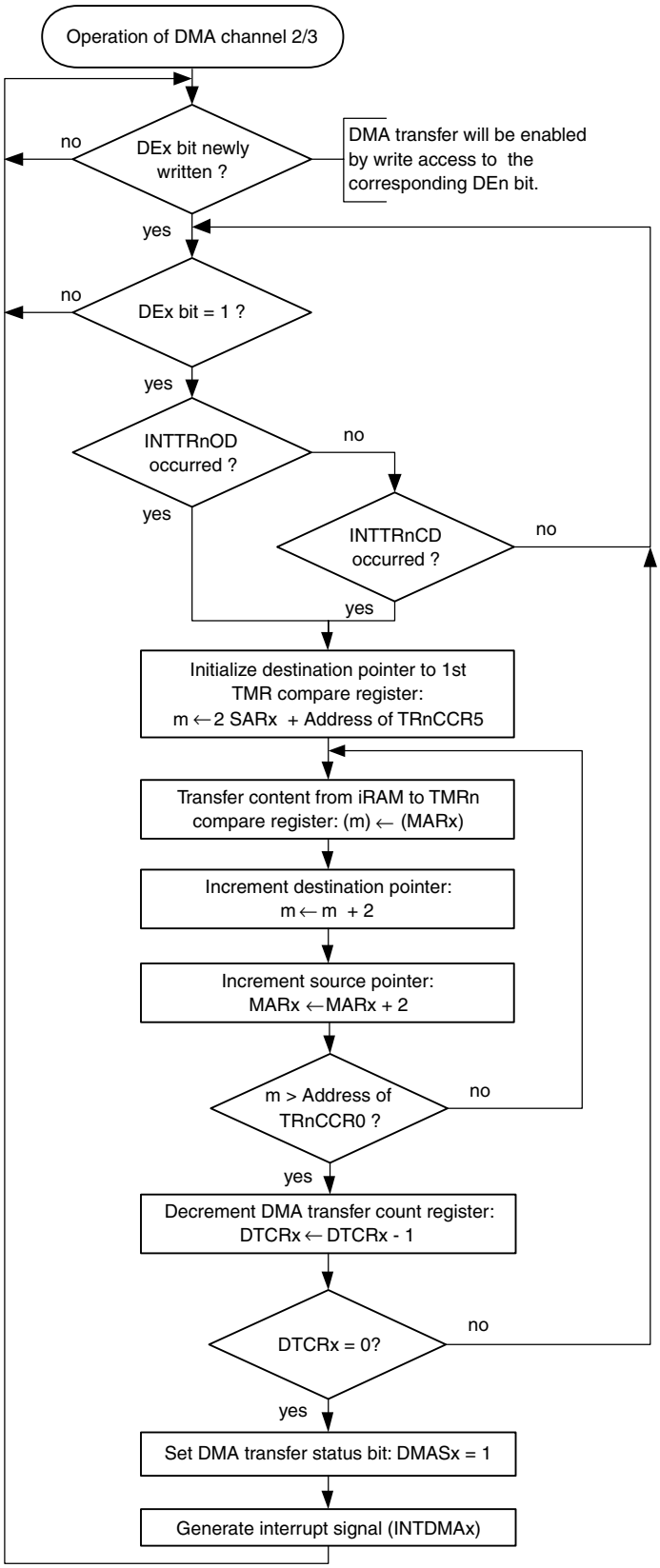
Remark: n = 0, 1
m = 0 to 5
x = n + 2

Figure 6-11: Initialization of DMA Transfer for TMRn Compare Registers



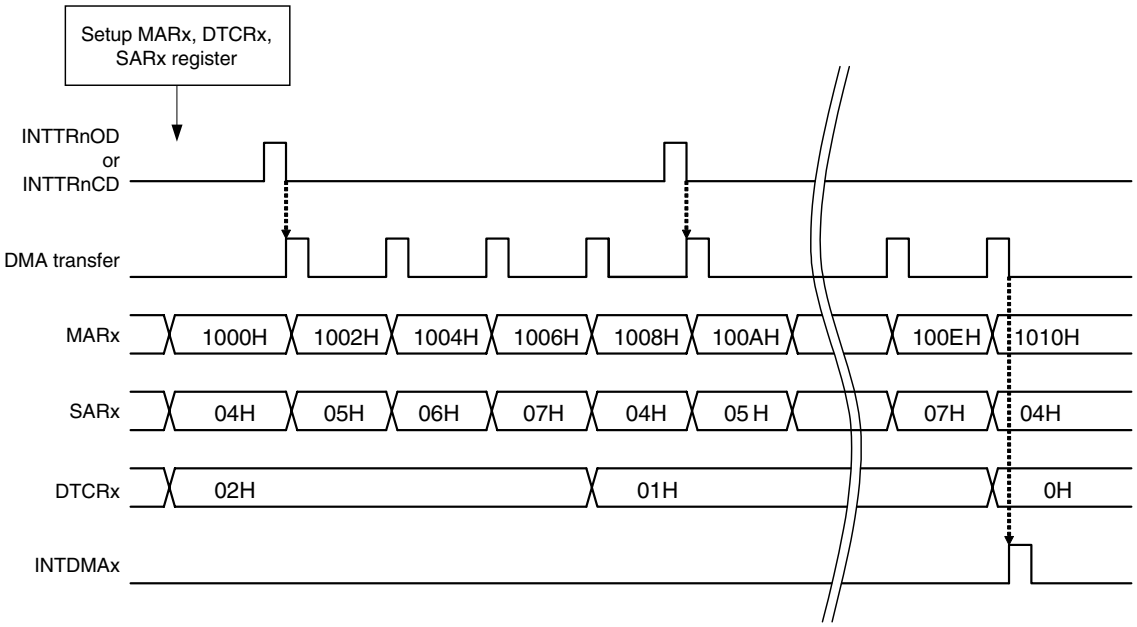
Remark: n = 0, 1 (number of TMR channel)
x = n + 2 (number of DMA channel)

Figure 6-12: Operation of DMA Channel 2/3



Remark: n = 0, 1 (number of TMR channel)
x = n + 2 (number of DMA transfer channel)

Figure 6-13: DMA Channel 2 and 3 Trigger Signal Timing



Remarks: 1. The DMA request by INTTRnOD or INTTRnCD is disregarded after INTDMAx is generated, and the DMA transfer is not restarted automatically. Write “1” in the corresponding DEx bit of the DMAMC register again to enable the next transfer of DMA channel x. The DEx bit is not cleared by hardware.

2. n = 0, 1 (number of the TMR channel)
x = n+2 (number of the DMA channel)

6.4.3 DMA transfer of serial interfaces

(1) Serial data reception with DMA transfer

The DMAC has two dedicated channels (4 and 5) to support the serial data reception. Each of both channels can be assigned to a serial interface (CSI30, CSI31, CSIB0, CSIB1, UARTC0, UARTC1). As DMA trigger factor, which requests and starts the DMA transfer, the corresponding interrupt signal at the end of reception is pre-defined (ref. to Table 6-2).

For each DMA trigger the data will be transferred from the corresponding serial reception register to internal RAM. Depending on the serial interface the transfer data size can be set to 8 or 16 bits (refer to Table 6-2).

In case of 8 bits transfer data size, the destination address is incremented by 1 for each occurrence of DMA trigger. When selecting 16 bits transfer data size the destination address must be even, and is incremented by 2 for each DMA trigger.

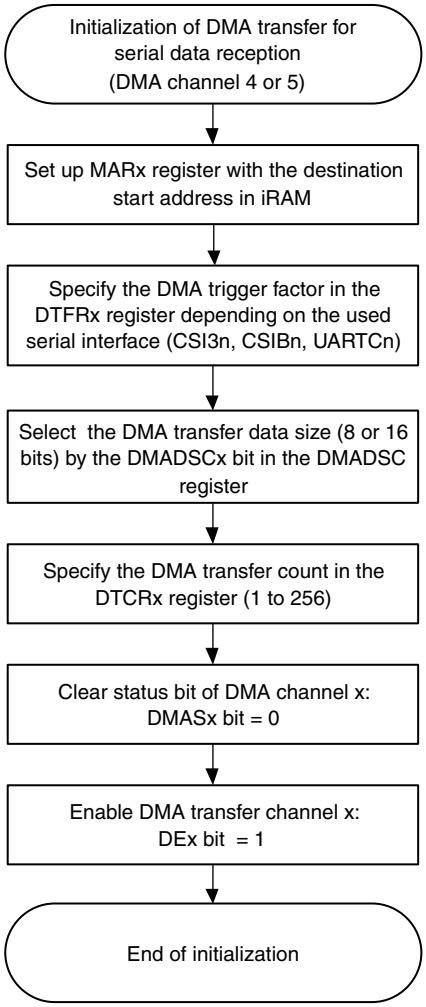
When the DMA transfer count of a DMA channel terminates, the DMA transfer is stopped and a DMA completion interrupt is generated. The maximum DMA transfer count is 255.

Table 6-2: DMA Configuration of Serial Data Reception

Serial Interface	DMA Trigger Factor	Transfer Data Size	Source	Destination
CSI30	INTC30	8 bits	SIRB0L	Any iRAM address
		16 bits	SIRB0	Any even iRAM address
CSI31	INTC31	8 bits	SIRB1L	Any iRAM address
		16 bits	SIRB1	Any even iRAM address
CSIB0	INTCB0T	8 bits	CB0RXL	Any iRAM address
		16 bits	CB0RX	Any even iRAM address
CSIB1	INTCB1T	8 bits	CB1RXL	Any iRAM address
		16 bits	CB1RX	Any even iRAM address
UARTC0	INTUC0T	8 bits	UC0RX	Any iRAM address
		16 bits	Setting prohibited	
UARTC1	INTUC1T	8 bits	UC1RX	Any iRAM address
		16 bits	Setting prohibited	

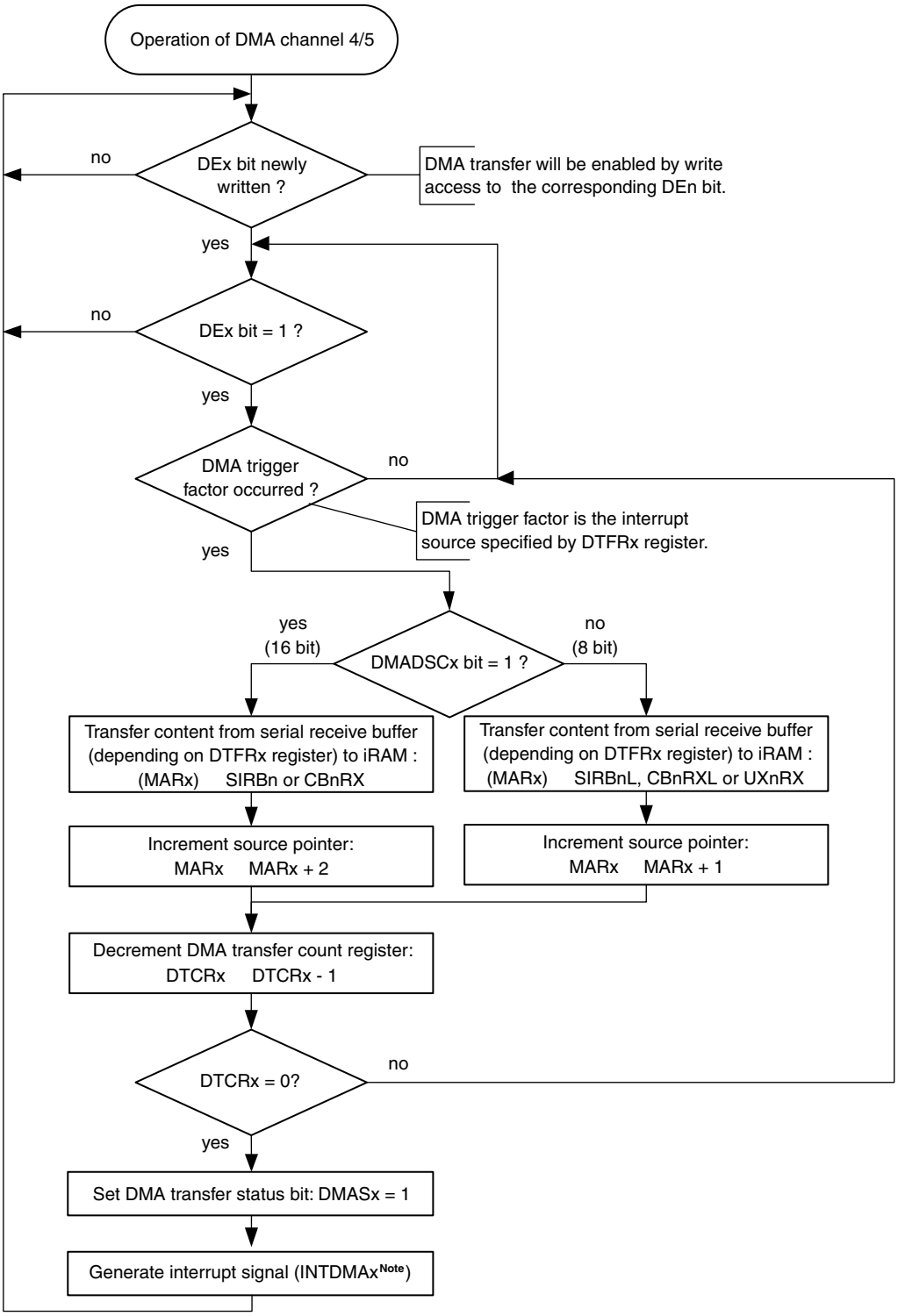
The procedure of the DMA transfer in case of serial data reception is shown in Figure 6-14.

Figure 6-14: Initialization of DMA Transfer for Serial Data Reception



Remark: n = 0, 1 (number of serial interface channel)
x = 4, 5 (number of DMA channel)

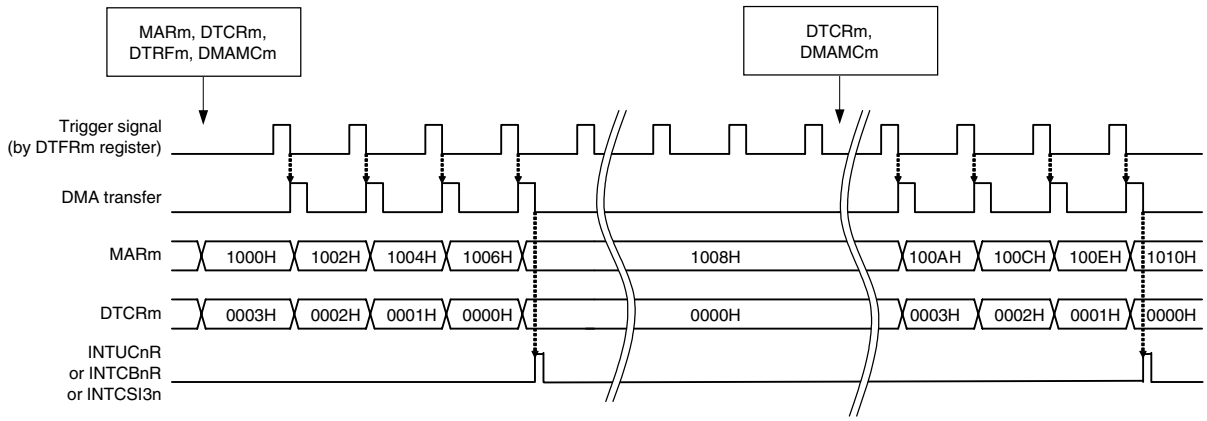
Figure 6-15: Operation of DMA Channel 4/5



Note: DMA transfer completion interrupt has the same interrupt vector address as the corresponding reception completion interrupt specified by DTFRX register, and replaces that interrupt.

Remark: n = 0, 1 (number of serial interface channel)
x = 4, 5 (number of DMA transfer channel)

Figure 6-16: DMA Channel 4 and 5 Trigger Signal Timing



Remark: m = 4, 5
n = 0, 1

(2) Serial data transmission with DMA transfer

The DMAC has two dedicated channels (6 and 7) to support the serial data transmission. Each of both channels can be assigned to a serial interface (CSI30, CSI31, CSIB0, CSIB1, UARTC0, UARTC1). As DMA trigger factor, which requests and starts the DMA transfer, the corresponding transmission enable interrupt signal is pre-defined (refer to Table 6-3).

For each DMA trigger the data will be transferred from internal RAM to the corresponding serial transmit register. Depending on the serial interface the transfer data size can be set to 8 or 16 bits (refer to Table 6-3).

In case of 8 bits transfer data size, the source address is incremented by 1 for each occurrence of DMA trigger. When selecting 16 bits transfer data size the source address must be even, and is incremented by 2 for each DMA trigger.

When the DMA transfer count of a DMA channel terminates, the DMA transfer is stopped and a DMA completion interrupt is generated. The maximum DMA transfer count is 255.

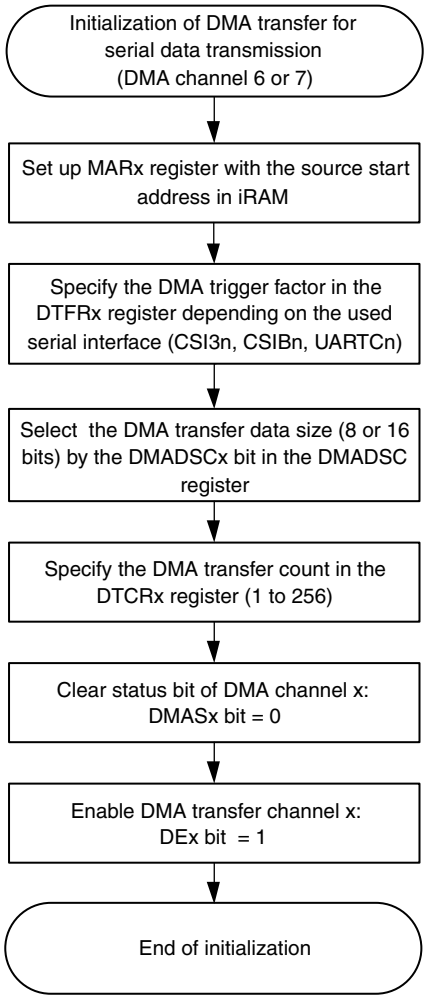
Table 6-3: DMA Configuration of Serial Data Transmission

Serial Interface	DMA Trigger Factor	Transfer Data Size	Source	Destination
CSI30**Note**	INTC30	8 bits	Any iRAM address	SFDB0L
		16 bits	Any even iRAM address	SFDB0
CSI31**Note**	INTC31	8 bits	Any iRAM address	SFDB1L
		16 bits	Any even iRAM address	SFDB1
CSIB0	INTCB0T	8 bits	Any iRAM address	CB0TXL
		16 bits	Any even iRAM address	CB0TX
CSIB1	INTCB1T	8 bits	Any iRAM address	CB1TXL
		16 bits	Any even iRAM address	CB1TX
UARTC0	INTUC0T	8 bits	Any iRAM address	UC0TX
		16 bits	Setting prohibited	
UARTC1	INTUC1T	8 bits	Any iRAM address	UC1TX
		16 bits	Setting prohibited	

Note: The serial peripheral chip select lines SCS0 to SCS3 will not be supported by DMA transfer.

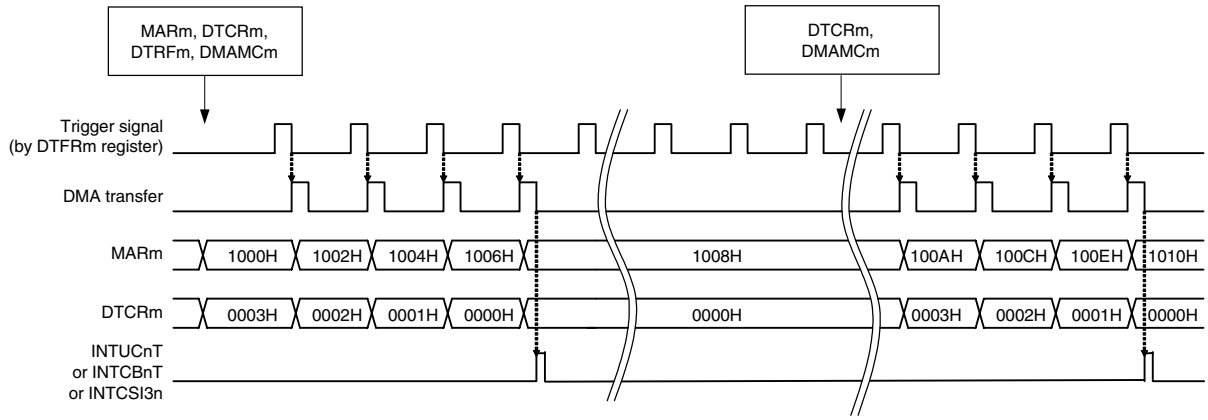
The procedure of the DMA transfer in case of serial data transmission is shown in Figure 6-17.

Figure 6-17: Initialization of DMA Transfer for Serial Data Transmission



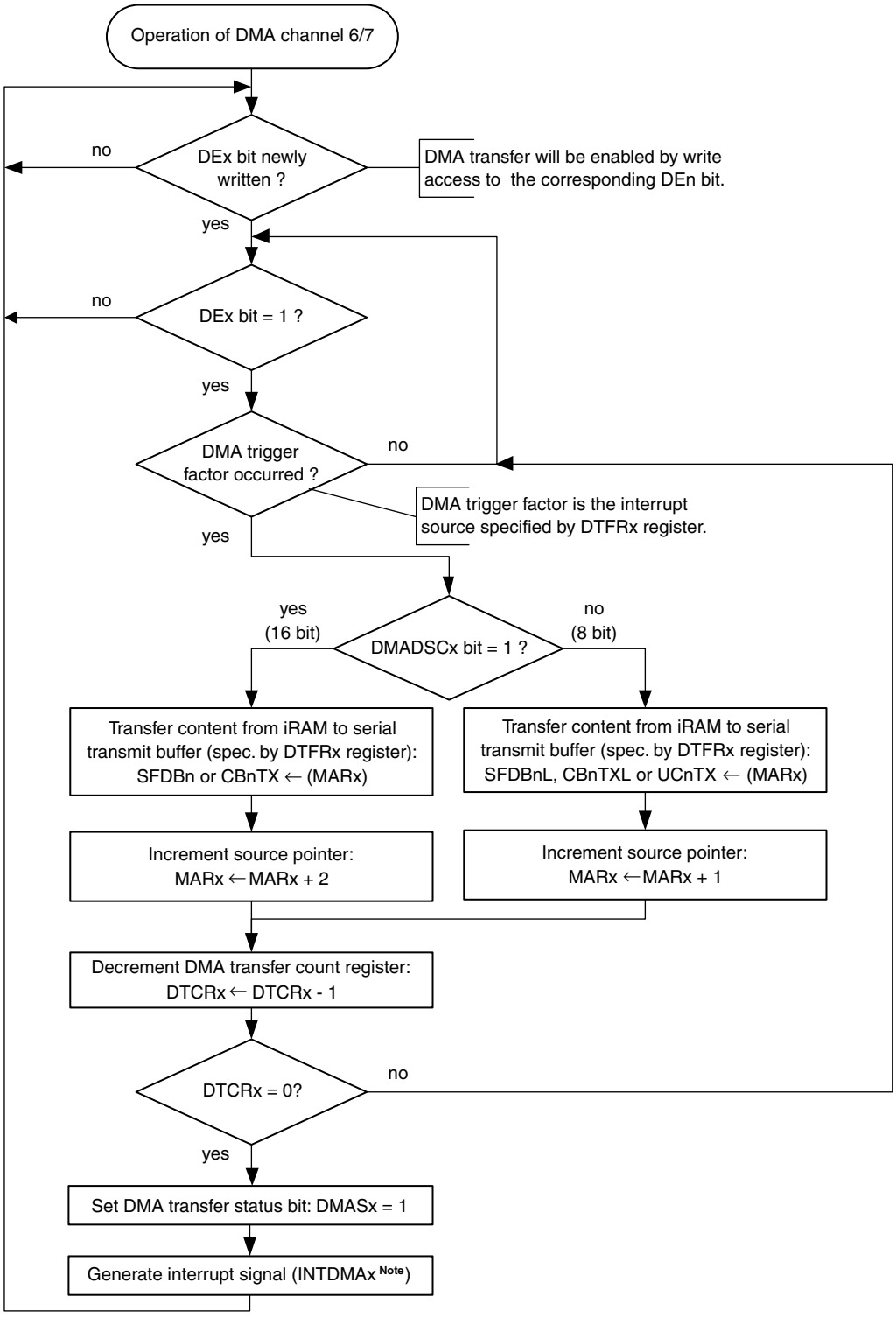
Remark: n = 0, 1 (number of serial interface channel)
x = 6, 7 (number of DMA channel)

Figure 6-18: DMA Channel 6 and 7 Trigger Signal Timing



Remark: m = 6, 7
n = 0, 1

Figure 6-19: Operation of DMA Channel 6/7



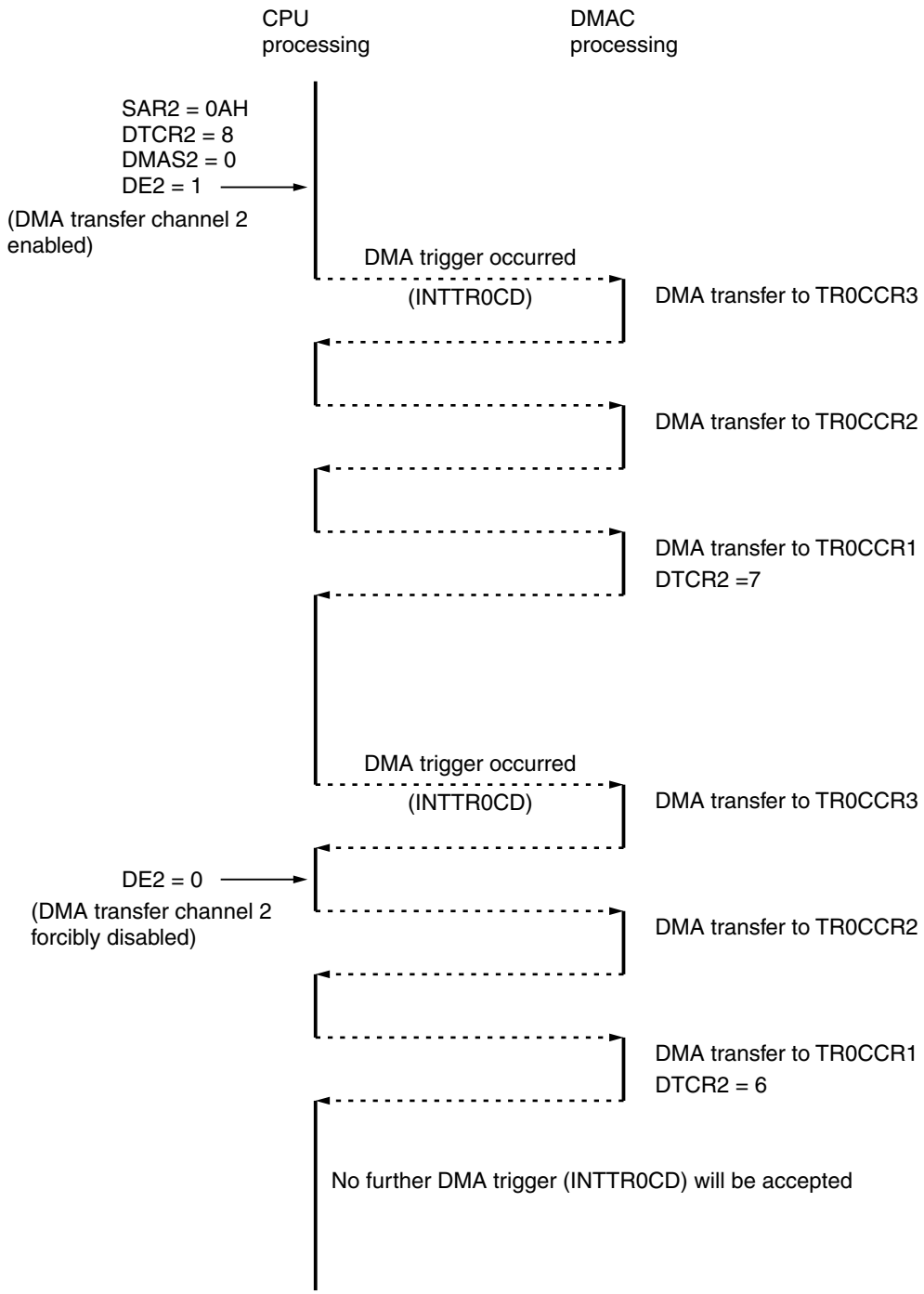
Note: DMA transfer completion interrupt has the same interrupt vector address as the corresponding transmission start interrupt specified by DTFRX register, and replaces that interrupt.

Remark: n = 0, 1 (number of serial interface channel)
x = 6, 7 (number of DMA transfer channel)

6.4.4 Forcible termination of DMA transfer

A once started DMA transfer can be forcible terminated when the corresponding DEn bit in the DMAMC register is cleared (0). However, if the DEn bit is cleared while DMA transferring, an once started data transfer is stopped first after it has been finished (see Figure 6-20).

Figure 6-20: CPU and DMA Controller Processing of DMA Transfer Termination (Example)



6.5 DMA Interrupt Function

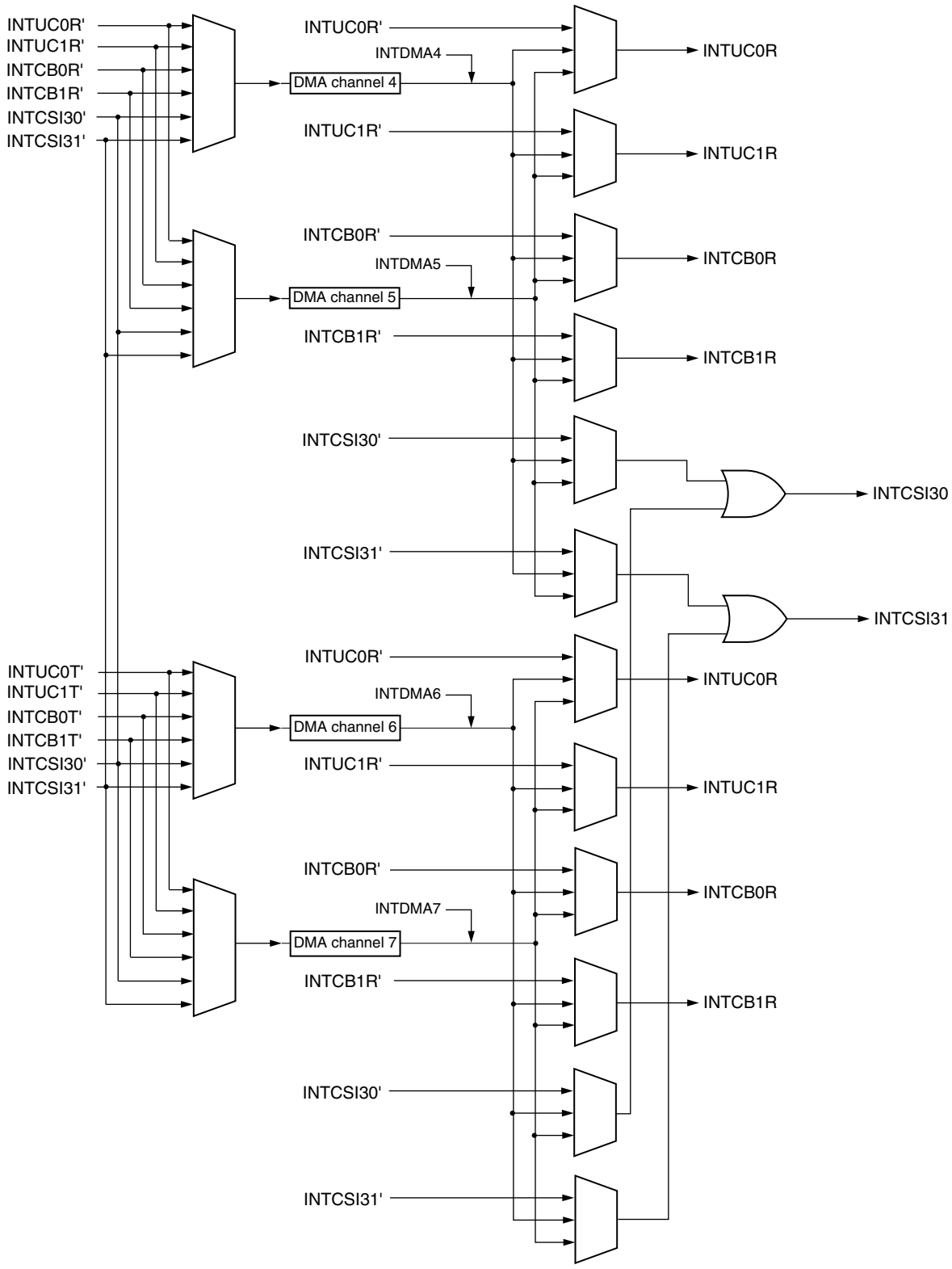
The peripheral I/O interrupts of the A/D converters and the serial interfaces, which serve as DMA trigger factors, are shared with the DMA transfer completion interrupt of the corresponding channel n (INTDMAn) (n = 0, 1, 4 to 7). When a DMA channel is enabled the specified peripheral I/O interrupt is no longer applied to the interrupt controller. Instead of it the corresponding DMA transfer completion interrupt is applied to the appropriate interrupt handler address.
In opposite to the other interrupts serving as DMA trigger factors, the TMR0 interrupts INTTR0OD and INTTR0CD, and the TMR1 interrupts INTTR1OD and INTTR1CD respectively, are not shared with DMA transfer completion interrupt of channel 2 (INTDMA2) and channel 3 (INTDMA3) respectively. These DMA completion interrupts have dedicated entries in the interrupt source list (refer to **Table 7-1: "Interrupt/Exception Source List" on page 207**).
Table 6-4 shows the relations between DMA trigger factors and DMA completion interrupts.

Table 6-4: Relations Between DMA Trigger Factors and DMA Completion Interrupts

DMA channel	DMA trigger factor	DMA completion interrupt			Remark
		Name	Entry	Handler Address	
0	INTAD0	INTDMA0	INTAD0	00000670H	**Note**
1	INTAD1	INTDMA1	INTAD1	00000680H	**Note**
2	INTTR0CD or INTR0OD	INTDMA2	INTDMA2	000006F0H	
3	INTTR1CD or INTR1OD	INTDMA3	INTDMA3	00000700H	
4, 5	INTC30	INTDMA4, INTDMA5	INTC30	000005E0H	**Note**
	INTC31		INTC31	00000600H	**Note**
	INTCB0R		INTCB0R	00000580H	**Note**
	INTCB1R		INTCB1R	000005B0H	**Note**
	INTUC0R		INTUC0R	00000620H	**Note**
	INTUC1R		INTUC1R	00000650H	**Note**
6, 7	INTC30	INTDMA6, INTDMA7	INTC30	000005E0H	**Note**
	INTC31		INTC31	00000600H	**Note**
	INTCB0T		INTCB0T	00000570H	**Note**
	INTCB1T		INTCIB1T	000005A0H	**Note**
	INTUC0T		INTUC0T	00000630H	**Note**
	INTUC1T		INTUC1T	00000660H	**Note**

Note: An interrupt request is not generated for a signal, which serves as DMA trigger factor. Instead of this the defined DMA completion interrupt request is executed on the same interrupt entry address of the DMA trigger factor.

Figure 6-21: Correlation between Serial I/O Interface Interrupts and DMA Completion Interrupts



Remark: Interrupt signals with quote mark (') are signals, which are directly connected from the corresponding serial interface.
Interrupt signals without quote mark are provided to the interrupt controller.

Chapter 7 Interrupt/Exception Processing Function

The V850E/PH2 microcontroller is provided with a dedicated interrupt controller (INTC) for interrupt servicing, which realizes a high-performance interrupt function that can service interrupt requests from a total of 107 sources.
An interrupt is an event that occurs asynchronously (independently of program execution), and an exception is an event that occurs synchronously (dependently on program execution). Generally, an exception takes precedence over an interrupt.
The V850E/PH2 microcontroller can process interrupt requests from the internal peripheral hardware and external sources. Moreover, exception processing can be started (exception trap) by the TRAP instruction (software exception) or by generation of an exception event (fetching of an illegal op code).

7.1 Features

- Interrupts
  - Non-maskable interrupt: 1 source
  - Maskable interrupt: 106 sources
  - 8 levels programmable priorities
  - Mask specification for the interrupt request according to priority
  - Mask can be specified to each maskable interrupt request.
  - Valid edge for detection of external interrupt request signal can be specified.
- Exceptions
  - Software exceptions: 32 sources
  - Exception trap: 1 source (illegal op code exception)

Interrupt/exception sources are listed in Table 7-1.

Table 7-1: Interrupt/Exception Source List (1/5)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Control Register	Generating Source	Gener. Unit				
Reset	Interrupt	RESET	–	$\overline{\text{RESET}}$ input	Pin	–	0000H	00000000H	undefined
Non-maskable	Interrupt	NMI	–	NMI input	Pin	–	0010H	00000010H	nextPC
Software exception	Exception	TRAP0n[Note]	–	TRAP instruction	–	–	004nH[Note]	00000040H	nextPC
	Exception	TRAP1n[Note]	–	TRAP instruction	–	–	005nH[Note]	00000050H	nextPC
Exception trap	Exception	ILGOP/ DBTRAP	–	Illegal opcode/ DBTRAP instruction	–	–	0060H	00000060H	nextPC
Maskable	Interrupt	INTP0	PIC0	INTP0 valid edge input	Pin	0	0080H	00000080H	nextPC
	Interrupt	INTP1	PIC1	INTP1 valid edge input	Pin	1	0090H	00000090H	nextPC
	Interrupt	INTP2	PIC2	INTP2 valid edge input	Pin	2	00A0H	000000A0H	nextPC
	Interrupt	INTP3	PIC3	INTP3 valid edge input	Pin	3	00B0H	000000B0H	nextPC
	Interrupt	INTP4	PIC4	INTP4 valid edge input	Pin	4	00C0H	000000C0H	nextPC

Note: n = 0 to FH

Table 7-1: Interrupt/Exception Source List (2/5)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Control Register	Generating Source	Gener. Unit				
Maskable	Interrupt	INTP5	PIC5	INTP5 valid edge input	Pin	5	00D0H	000000D0H	nextPC
	Interrupt	INTP6	PIC6	INTP6 valid edge input	Pin	6	00E0H	000000E0H	nextPC
	Interrupt	INTP7	PIC7	INTP7 valid edge input	Pin	7	00F0H	000000F0H	nextPC
	Interrupt	INTP8	PIC8	INTP8 valid edge input	Pin	8	0100H	00000100H	nextPC
	Interrupt	INTP9	PIC9	INTP9 valid edge input	Pin	9	0110H	00000110H	nextPC
	Interrupt	INTP10	PIC10	INTP10 valid edge input	Pin	10	0120H	00000120H	nextPC
	Interrupt	INTP11	PIC11	INTP11 valid edge input	Pin	11	0130H	00000130H	nextPC
	Interrupt	INTP12	PIC12	INTP12 valid edge input	Pin	12	0140H	00000140H	nextPC
	Interrupt	INTTR0OV	PIC13	TR0CNT overflow	TMR0	13	0150H	00000150H	nextPC
	Interrupt	INTTR0CC0	PIC14	TR0CCR0 match	TMR0	14	0160H	00000160H	nextPC
	Interrupt	INTTR0CC1	PIC15	TR0CCR1 match	TMR0	15	0170H	00000170H	nextPC
	Interrupt	INTTR0CC2	PIC16	TR0CCR2 match	TMR0	16	0180H	00000180H	nextPC
	Interrupt	INTTR0CC3	PIC17	TR0CCR3 match	TMR0	17	0190H	00000190H	nextPC
	Interrupt	INTTR0CC4	PIC18	TR0CCR4 match	TMR0	18	01A0H	000001A0H	nextPC
	Interrupt	INTTR0CC5	PIC19	TR0CCR5 match	TMR0	19	01B0H	000001B0H	nextPC
	Interrupt	INTTR0CD	PIC20	TR0CNT top reversal	TMR0	20	01C0H	000001C0H	nextPC
	Interrupt	INTTR0OD	PIC21	TR0CNT bottom reversal	TMR0	21	01D0H	000001D0H	nextPC
	Interrupt	INTTR0ER	PIC22	TMR0 error detection	TMR0	22	01E0H	000001E0H	nextPC
	Interrupt	INTTR1OV	PIC23	TR1CNT overflow	TMR1	23	01F0H	000001F0H	nextPC
	Interrupt	INTTR1CC0	PIC24	TIR10 capture input/ TR1CCR0 match	TMR1	24	0200H	00000200H	nextPC
	Interrupt	INTTR1CC1	PIC25	TIR11 capture input/ TR1CCR1 match	TMR1	25	0210H	00000210H	nextPC
	Interrupt	INTTR1CC2	PIC26	TIR12 capture input/ TR1CCR2 match	TMR1	26	0220H	00000220H	nextPC
	Interrupt	INTTR1CC3	PIC27	TIR13 capture input/ TR1CCR3 match	TMR1	27	0230H	00000230H	nextPC
	Interrupt	INTTR1CC4	PIC28	TR1CCR4 match	TMR1	28	0240H	00000240H	nextPC
	Interrupt	INTTR1CC5	PIC29	TR1CCR5 match	TMR1	29	0250H	00000250H	nextPC
	Interrupt	INTTR1CD	PIC30	TR1CNT top reversal	TMR1	30	0260H	00000260H	nextPC
	Interrupt	INTTR1OD	PIC31	TR1CNT bottom reversal	TMR1	31	0270H	00000270H	nextPC
	Interrupt	INTTR1ER	PIC32	TMR1 error detection	TMR1	32	0280H	00000280H	nextPC
	Interrupt	INTT0OV	PIC33	TMT0 overflow	TMT0	33	0290H	00000290H	nextPC
	Interrupt	INTT0CC0	PIC34	TIT00 capture input/ TT0CCR0 match	TMT0	34	02A0H	000002A0H	nextPC
	Interrupt	INTT0CC1	PIC35	TIT01 capture input/ TT0CCR1 match	TMT0	35	02B0H	000002B0H	nextPC
	Interrupt	INTT0EC	PIC36	TMT0 encoder clear	TMT0	36	02C0H	000002C0H	nextPC
	Interrupt	INTT1OV	PIC37	TMT1 overflow	TMT1	37	02D0H	000002D0H	nextPC
	Interrupt	INTT1CC0	PIC38	TIT10 capture input/ TT1CCR0 match	TMT1	38	02E0H	000002E0H	nextPC
	Interrupt	INTT1CC1	PIC39	TIT11 capture input/ TT1CCR1 match	TMT1	39	02F0H	000002F0H	nextPC

Table 7-1: Interrupt/Exception Source List (3/5)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Control Register	Generating Source	Gener. Unit				
Maskable	Interrupt	INTT1EC	PIC40	TMT1 encoder clear	TMT1	40	0300H	00000300H	nextPC
	Interrupt	INTP0OV	PIC41	TMP0 overflow	TMP0	41	0310H	00000310H	nextPC
	Interrupt	INTP0CC0	PIC42	TIP00 capture input/ TP0CCR0 match	TMP0	42	0320H	00000320H	nextPC
	Interrupt	INTP0CC1	PIC43	TIP01 capture input/ TP0CCR1 match	TMP0	43	0330H	00000330H	nextPC
	Interrupt	INTP1OV	PIC44	TMP1 overflow	TMP1	44	0340H	00000340H	nextPC
	Interrupt	INTP1CC0	PIC45	TIP10 pin/ TP1CCR0 match	TMP1	45	0350H	00000350H	nextPC
	Interrupt	INTP1CC1	PIC46	TIP11 capture input/ TP1CCR1 match	TMP1	46	0360H	00000360H	nextPC
	Interrupt	INTP2OV	PIC47	TMP2 overflow	TMP2	47	0370H	00000370H	nextPC
	Interrupt	INTP2CC0	PIC48	TIP20 capture input/ TP2CCR0 match	TMP2	48	0380H	00000380H	nextPC
	Interrupt	INTP2CC1	PIC49	TIP21capture input/ TP2CCR1 match	TMP2	49	0390H	00000390H	nextPC
	Interrupt	INTP3OV	PIC50	TMP3 overflow	TMP3	50	03A0H	000003A0H	nextPC
	Interrupt	INTP3CC0	PIC51	TIP30 capture input/ TP3CCR0 match	TMP3	51	03B0H	000003B0H	nextPC
	Interrupt	INTP3CC1	PIC52	TIP31 capture input/ TP3CCR1 match	TMP3	52	03C0H	000003C0H	nextPC
	Interrupt	INTP4OV	PIC53	TMP4 overflow	TMP4	53	03D0H	000003D0H	nextPC
	Interrupt	INTP4CC0	PIC54	TIP40 capture input/ TP4CCR0 match	TMP4	54	03E0H	000003E0H	nextPC
	Interrupt	INTP4CC1	PIC55	TIP41 capture input/ TP4CCR1 match	TMP4	55	03F0H	000003F0H	nextPC
	Interrupt	INTP5OV	PIC56	TMP5overflow	TMP5	56	0400H	00000400H	nextPC
	Interrupt	INTP5CC0	PIC57	TIP50 capture input/ TP5CCR0 match	TMP5	57	0410H	00000410H	nextPC
	Interrupt	INTP5CC1	PIC58	TIP51 capture input/ TP5CCR1 match	TMP5	58	0420H	00000420H	nextPC
	Interrupt	INTP6OV	PIC59	TMP6 overflow	TMP6	59	0430H	00000430H	nextPC
	Interrupt	INTP6CC0	PIC60	TIP60 capture input/ TP6CCR0 match	TMP6	60	0440H	00000440H	nextPC
	Interrupt	INTP6CC1	PIC61	TIP61 capture input/ TP6CCR1 match	TMP6	61	0450H	00000450H	nextPC
	Interrupt	INTP7OV	PIC62	TMP7 overflow	TMP7	62	0460H	00000460H	nextPC
	Interrupt	INTP7CC0	PIC63	TIP70 capture input/ TP7CCR0 match	TMP7	63	0470H	00000470H	nextPC
	Interrupt	INTP7CC1	PIC64	TIP71 capture input/ TP7CCR1 match	TMP7	64	0480H	00000480H	nextPC
	Interrupt	INTP8OV	PIC65	TMP8 overflow	TMP8	65	0490H	00000490H	nextPC
	Interrupt	INTP8CC0	PIC66	TP8CCR0 match	TMP8	66	04A0H	000004A0H	nextPC
	Interrupt	INTP8CC1	PIC67	TP8CCR1 match	TMP8	67	04B0H	000004B0H	nextPC
	Interrupt	INTBRG0	PIC68	BRG0 match	BRG0	68	04C0H	000004C0H	nextPC
	Interrupt	INTBRG1	PIC69	BRG1 match	BRG1	69	04D0H	000004D0H	nextPC
	Interrupt	INTBRG2	PIC70	BRG2 match	BRG2	70	04E0H	000004E0H	nextPC

Table 7-1: Interrupt/Exception Source List (4/5)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Control Register	Generating Source	Gener. Unit				
Maskable	Interrupt	INTC0ERR	PIC71	FCAN0 error	FCAN0	71	04F0H	000004F0H	nextPC
	Interrupt	INTC0WUP	PIC72	FCAN0 wake up	FCAN0	72	0500H	00000500H	nextPC
	Interrupt	INTC0REC	PIC73	FCAN0 bus reception	FCAN0	73	0510H	00000510H	nextPC
	Interrupt	INTC0TRX	PIC74	FCAN0 bus transmission	FCAN0	74	0520H	00000520H	nextPC
	Interrupt	INTC1ERR	PIC75	FCAN1 error	FCAN1	75	0530H	00000530H	nextPC
	Interrupt	INTC1WUP	PIC76	FCAN1 wake up	FCAN1	76	0540H	00000540H	nextPC
	Interrupt	INTC1REC	PIC77	FCAN1 bus reception	FCAN1	77	0550H	00000550H	nextPC
	Interrupt	INTC1TRX	PIC78	FCAN1 bus transmission	FCAN1	78	0560H	00000560H	nextPC
	Interrupt	INTCB0T	PIC79	CSIB0 transmission enable/ DMA transfer completion	CSIB0/ DMAC	79	0570H	00000570H	nextPC
	Interrupt	INTCB0R	PIC80	CSIB0 reception completion/ DMA transfer completion	CSIB0/ DMAC	80	0580H	00000580H	nextPC
	Interrupt	INTCB0RE	PIC81	CSIB0 receive error	CSIB0	81	0590H	00000590H	nextPC
	Interrupt	INTCB1T	PIC82	CSIB1 transmission enable/ DMA transfer completion	CSIB1/ DMAC	82	05A0H	000005A0H	nextPC
	Interrupt	INTCB1R	PIC83	CSIB1 reception completion/ DMA transfer completion	CSIB1/ DMAC	83	05B0H	000005B0H	nextPC
	Interrupt	INTCB1RE	PIC84	CSIB1 receive error	CSIB1	84	05C0H	000005C0H	nextPC
	Interrupt	INTC30OVF	PIC85	CSI30 overrun	CSI30	85	05D0H	000005D0H	nextPC
	Interrupt	INTC30	PIC86	CSI30 transmission enable/ DMA transfer completion	CSI30/ DMAC	86	05E0H	000005E0H	nextPC
	Interrupt	INTC31OVF	PIC87	CSI31 overrun	CSI31	87	05F0H	000005F0H	nextPC
	Interrupt	INTC31	PIC88	CSI31 transmission enable/ DMA transfer completion	CSI31/ DMAC	88	0600H	00000600H	nextPC
	Interrupt	INTUC0RE	PIC89	UARTC0 receive error	UARTC0	89	0610H	00000610H	nextPC
	Interrupt	INTUC0R	PIC90	UARTC0 reception completion/ DMA transfer completion	UARTC0 / DMAC	90	0620H	00000620H	nextPC
	Interrupt	INTUC0T	PIC91	UARTC0 transmission enable/ DMA transfer completion	UARTC0 / DMAC	91	0630H	00000630H	nextPC
	Interrupt	INTUC1RE	PIC92	UARTC1receive error	UARTC1	92	0640H	00000640H	nextPC
	Interrupt	INTUC1R	PIC93	UARTC1 reception completion/ DMA transfer completion	UARTC1 / DMAC	93	0650H	00000650H	nextPC
	Interrupt	INTUC1T	PIC94	UARTC1 transmission enable/ DMA transfer completion	UARTC1 / DMAC	94	0660H	00000660H	nextPC
	Interrupt	INTAD0	PIC95	ADC0 conversion completion/ DMA transfer completion	ADC0/ DMAC	95	0670H	00000670H	nextPC

Table 7-1: Interrupt/Exception Source List (5/5)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Control Register	Generating Source	Gener. Unit				
Maskable	Interrupt	INTAD1	PIC96	ADC1 conversion completion/ DMA transfer completion	ADC1/ DMAC	96	0680H	00000680H	nextPC
	Interrupt	INTCC10	PIC97	CC10 capture input/ compare match	TMENC1	97	0690H	00000690H	nextPC
	Interrupt	INTCC11	PIC98	CC11capture input/ compare match	TMENC1	98	06A0H	000006A0H	nextPC
	Interrupt	INTCM10	PIC99	CM10 compare match	TMENC1	99	06B0H	000006B0H	nextPC
	Interrupt	INTCM11	PIC100	CM10 compare match	TMENC1	100	06C0H	000006C0H	nextPC
	Interrupt	INTOVF	PIC101	TMENC1 overflow	TMENC1	101	06D0H	000006D0H	nextPC
	Interrupt	INTUDF	PIC102	TMENC1 underflow	TMENC1	102	06E0H	000006E0H	nextPC
	Interrupt	INTDMA2	PIC103	DMA channel 2 transfer completion	DMAC	103	06F0H	000006F0H	nextPC
	Interrupt	INTDMA3	PIC104	DMA channel 3 transfer completion	DMAC	104	0700H	00000700H	nextPC
	Interrupt	INTPERR	PIC105	Internal RAM parity error	iRAM	105	0710H	00000710H	nextPC

Remarks: 1. Default Priority: The priority order that takes precedence when two or more maskable interrupt requests at the same software priority level are present at the same time. The highest priority is 0.

Restored PC: The value of PC saved when an interrupt/exception (other than RESET) occurs is the value of the current PC, which holds the address of the next instruction to be executed when returning from interrupt handling routine. However, if the interrupt request occurs during execution of a divide instruction (DIV, DIVH, DIVU or DIVHU), the value of the PC saved is the address of the divide instruction itself (rather than the address of the instruction following the divide instruction), because the division is cancelled in this case, and restarted completely after interrupt servicing.

nextPC: The PC value that proceeds the processing following interrupt/ exception processing.

2. The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC - 4).

7.2 Non-maskable Interrupt

A non-maskable interrupt request is acknowledged unconditionally, even when interrupts are in the interrupt disabled (DI) status. A NMI is not subject to priority control and takes precedence over all the other interrupts.
A non-maskable interrupt request is input from the NMI pin. When the valid edge specified by ESN0, ESN1 bits of the interrupt mode register 0 (INTM0) is detected at the NMI pin, the interrupt occurs.
While the service program of the non-maskable interrupt is being executed (PSW.NP = 1), the acknowledgment of another non-maskable interrupt request is held pending. The pending NMI is acknowledged after the original service program of the non-maskable interrupt under execution has been terminated (by the RETI instruction). Note that if two or more NMI requests are input during the execution of the service program for a NMI, the number of NMIs that will be acknowledged after PSW.NP is cleared to 0 is only one.

Remark: PSW.NP: The NP bit of the PSW register.

7.2.1 Operation

If a non-maskable interrupt is generated, the CPU performs the following processing, and transfers control to the handler routine:

(1) Saves the restored PC to FEPC.
(2) Saves the current PSW to FEPSW.
(3) Writes exception code 0010H to the higher half-word (FECC) of ECR.
(4) Sets the NP and ID bits of the PSW and clears the EP bit.
(5) Sets the handler address (00000010H) corresponding to the non-maskable interrupt to the PC, and transfers control.

The processing configuration of a non-maskable interrupt is shown in Figure 7-1.

Figure 7-1: Processing Configuration of Non-Maskable Interrupt

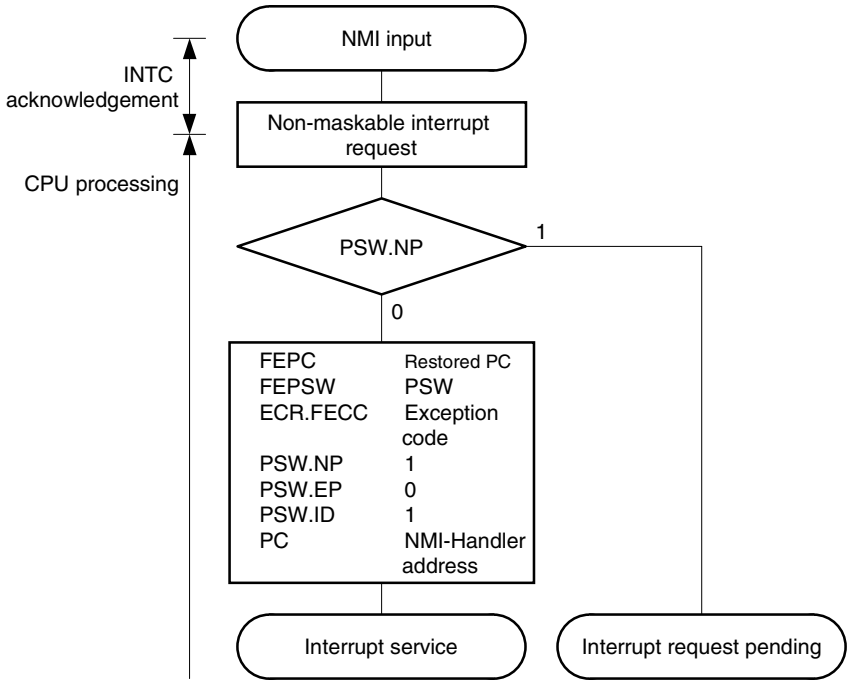
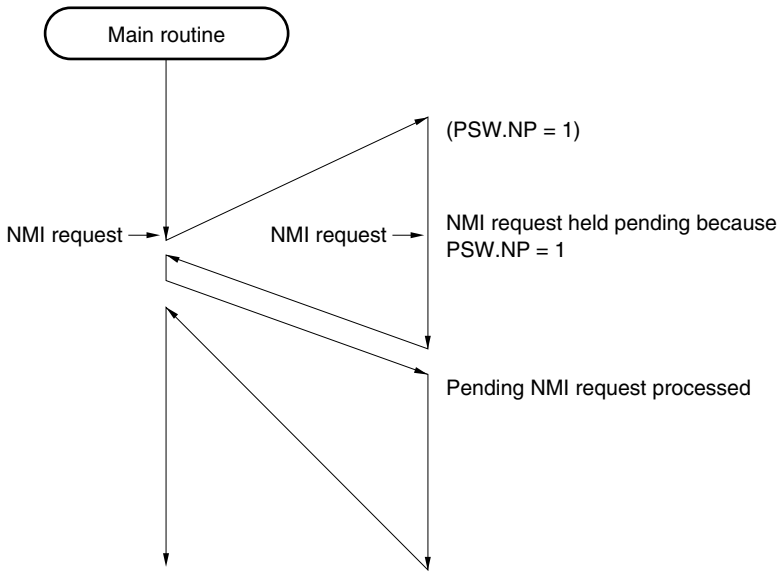
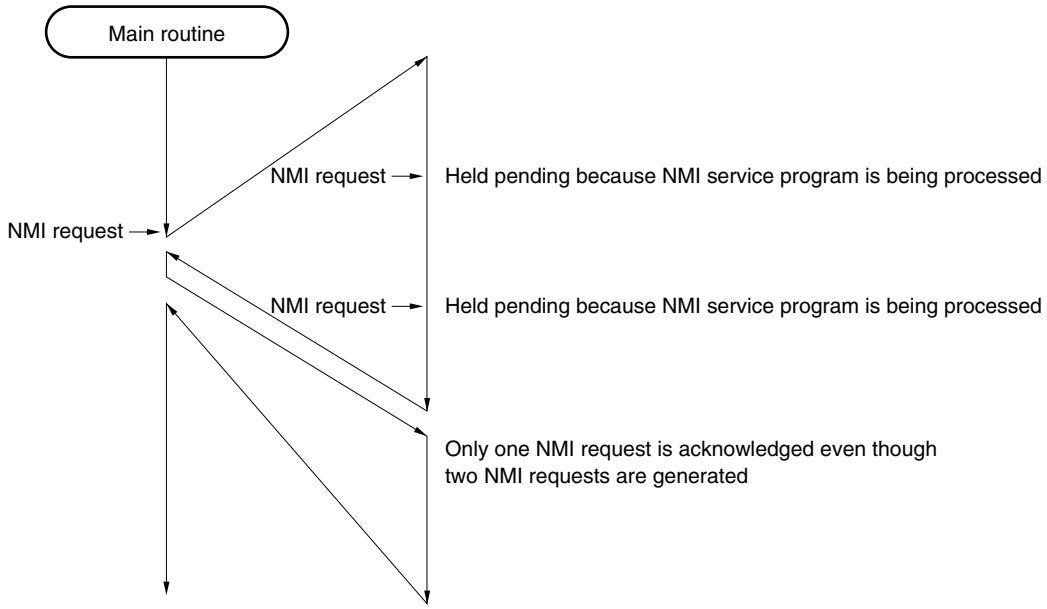


Figure 7-2: Acknowledging Non-Maskable Interrupt Request

(a) If a new NMI request is generated while a NMI service program is being executed



(b) If a new NMI request is generated twice while a NMI service program is being executed



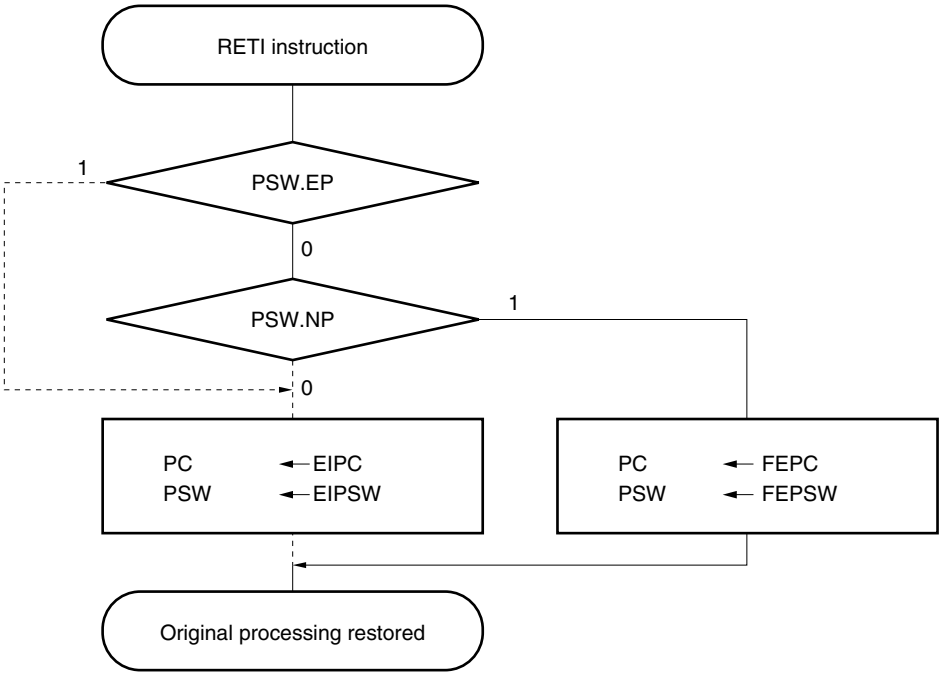
7.2.2 Restore

Execution is restored from the non-maskable interrupt (NMI) processing by the RETI instruction. When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<1> Restores the values of the PC and the PSW from FEPC and FEPSW, respectively, because the EP bit of the PSW is 0 and the NP bit of the PSW is 1.
<2> Transfers control back to the address of the restored PC and PSW.

Figure 7-3 illustrates how the RETI instruction is processed.

Figure 7-3: RETI Instruction Processing



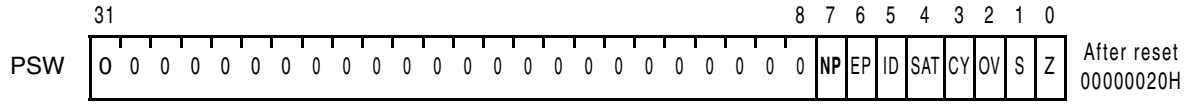
Caution: When the PSW.EP bit and PSW.NP bit are changed by the LDSR instruction during non-maskable interrupt processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark: The solid line indicates the CPU processing flow.

7.2.3 Non-maskable interrupt status flag (NP)

The NP flag is a status flag that indicates that non-maskable interrupt (NMI) processing is under execution.
This flag is set when a NMI interrupt has been acknowledged, and masks all interrupt requests and exceptions to prohibit multiple interrupts from being acknowledged.

Figure 7-4: Non-maskable Interrupt Status Flag (NP)



NP	NMI Servicing Status
0	No NMI interrupt servicing
1	NMI interrupt currently servicing

7.2.4 Edge Detection Function

The behaviour of the non-maskable interrupt (NMI) can be specified by the interrupt mode register 0 (INTM0). The valid edge of the external NMI pin input can be specified by the ESN0 and ESN1 bits.
The INTM0 register can be read/written in 8-bit or 1-bit units.

Figure 7-5: NMI Edge Detection Specification: Interrupt Mode Register 0 (INTM0)

After reset: 00H R/W Address: FFFFF880H

	7	6	5	4	3	2	1	0
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	ESN1	ESN0

ESN1	ESN0	Valid Edge Specification of NMI pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

7.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The V850E/PH2 has 106 maskable interrupt sources.
If two or more maskable interrupt requests are generated at the same time, they are acknowledged according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers (programmable priority control).
When an interrupt request has been acknowledged, the acknowledgement of other maskable interrupt requests is disabled and the interrupt disabled (DI) status is set.
When the EI instruction is executed in an interrupt processing routine, the interrupt enabled (EI) status is set, which enables servicing of interrupts having a higher priority than the interrupt request in progress (specified by the interrupt control register). Note that only interrupts with a higher priority will have this capability; interrupts with the same priority level cannot be nested.
However, if multiple interrupts are executed, the following processing is necessary.

(1) Save EIPC and EIPSW in memory or a general-purpose register before executing the EI instruction.
(2) Execute the DI instruction before executing the RETI instruction, then reset EIPC and EIPSW with the values saved in (1).

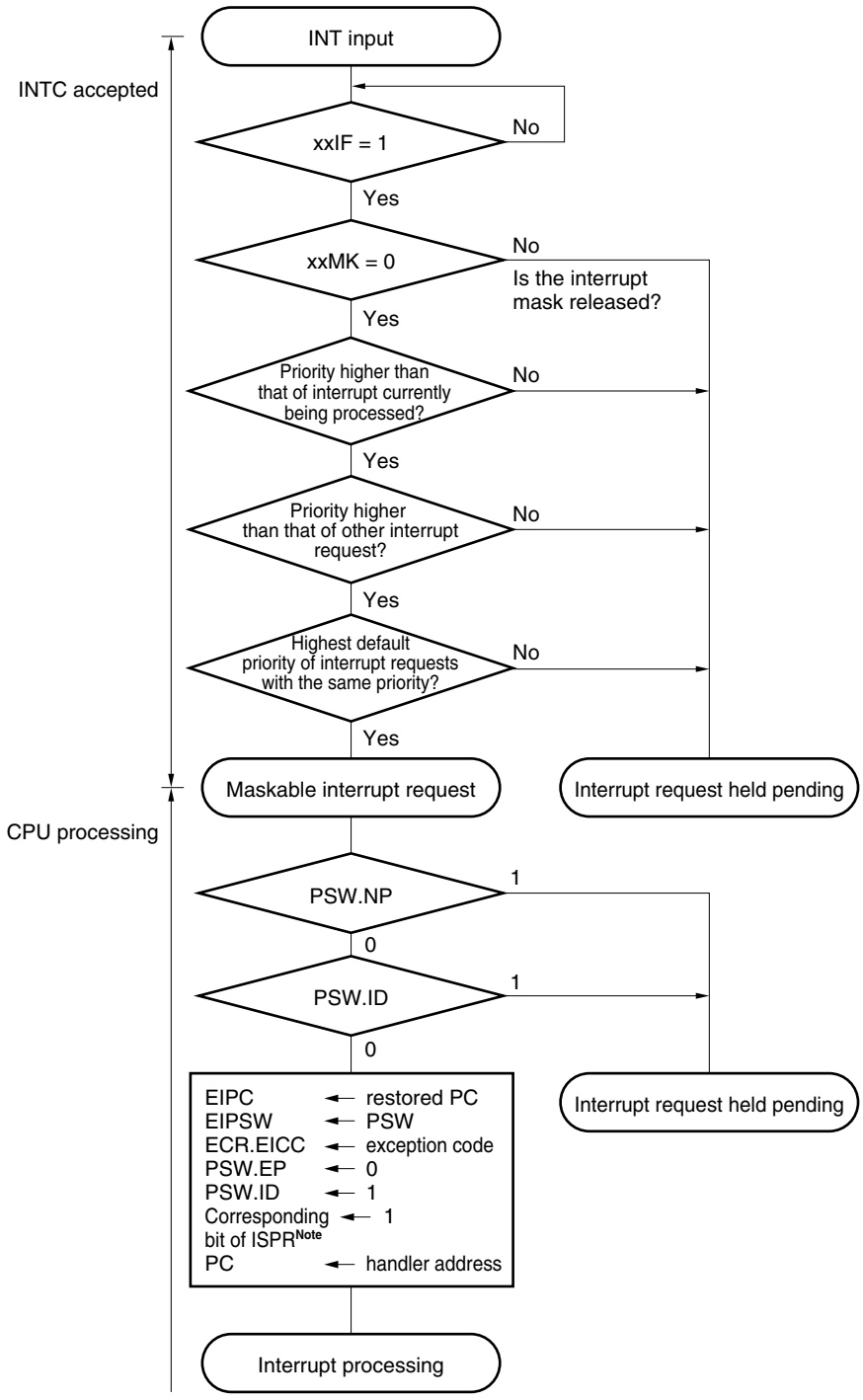
7.3.1 Operation

If a maskable interrupt occurs by INT input, the CPU performs the following processing, and transfers control to a handler routine:

(1) Saves the restored PC to EIPC.
(2) Saves the current PSW to EIPSW.
(3) Writes an exception code to the lower half-word of ECR (EICC).
(4) Sets the ID bit of the PSW and clears the EP bit.
(5) Sets the handler address corresponding to each interrupt to the PC, and transfers control.

The processing configuration of a maskable interrupt is shown in Figure 7-6.

Figure 7-6: Maskable Interrupt Processing



Note: For the ISPR register, see **7.3.6 "In-service priority register (ISPR)" on page 230**.

An INT input masked by the interrupt controllers and an INT input that occurs while another interrupt is being processed (when PSW.NP = 1 or PSW.ID = 1) are held pending internally by the interrupt controller. In such case, if the interrupts are unmasked, or when PSW.NP = 0 and PSW.ID = 0 as set by the RETI and LDSR instructions, input of the pending INT starts the new maskable interrupt processing.

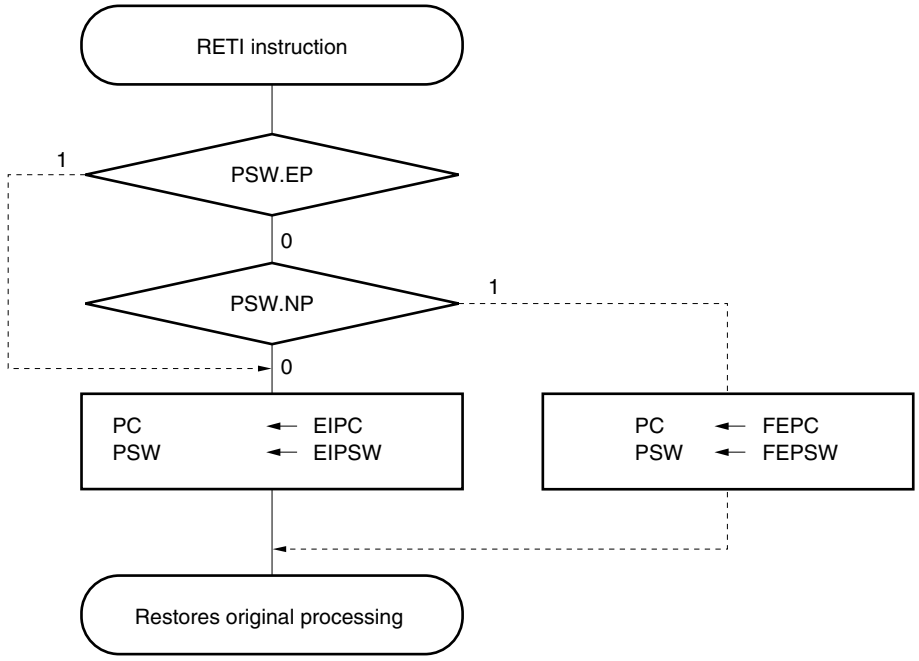
7.3.2 Restore

Recovery from maskable interrupt processing is carried out by the RETI instruction.
When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

(1) Restores the values of the PC and the PSW from EIPC and EIPSW because the EP bit of the PSW is 0 and the NP bit of the PSW is 0.
(2) Transfers control to the address of the restored PC and PSW.

Figure 7-7 illustrates the processing of the RETI instruction.

Figure 7-7: RETI Instruction Processing



Note: For the ISPR register, see **7.3.6 "In-service priority register (ISPR)" on page 230**.

Caution: When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during maskable interrupt processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark: The solid lines show the CPU processing flow.

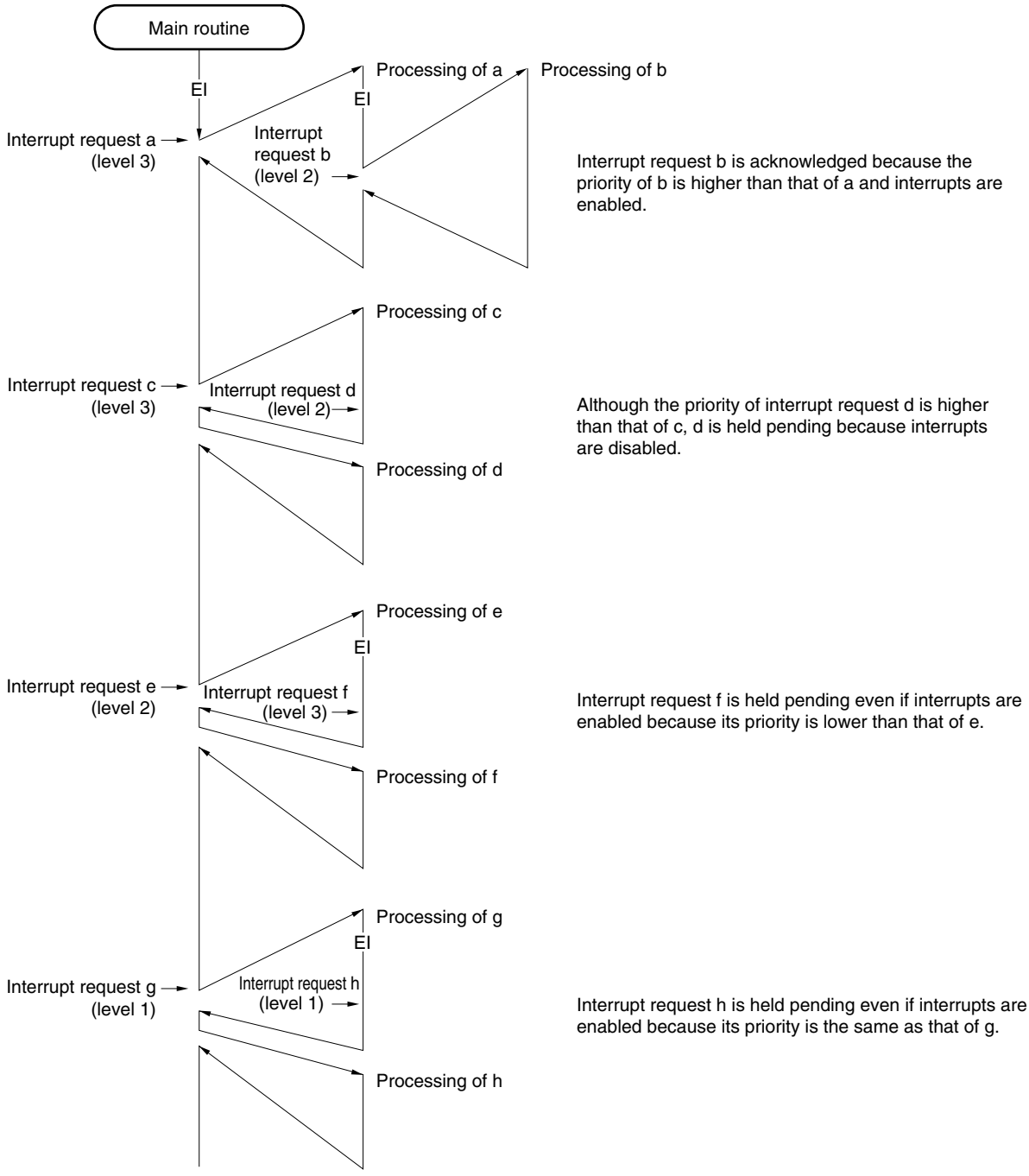
7.3.3 Priorities of maskable interrupts

The V850E/PH2 provides multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.
There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels that are specified by the interrupt priority level specification bit (PRn) of the interrupt control register (PICn). When two or more interrupts having the same priority level specified by the PRn bit are generated at the same time, interrupts are serviced in order depending on the priority level allocated to each interrupt request type (default priority level) beforehand. For more information, refer to **Table 7-1, "Interrupt/Exception Source List," on page 207**. The programmable priority control customizes interrupt requests into eight levels by setting the priority level specification flag.

Note that when an interrupt request is acknowledged, the ID flag of PSW is automatically set to 1. Therefore, when multiple interrupts are to be used, clear the ID flag to 0 beforehand (for example, by placing the EI instruction in the interrupt service program) to set the interrupt enable mode.

Remark: n = 0 to 105 (number of interrupt)

Figure 7-8: Example of Processing in which Another Interrupt Request Is Issued while an Interrupt is being Processed (1/2)

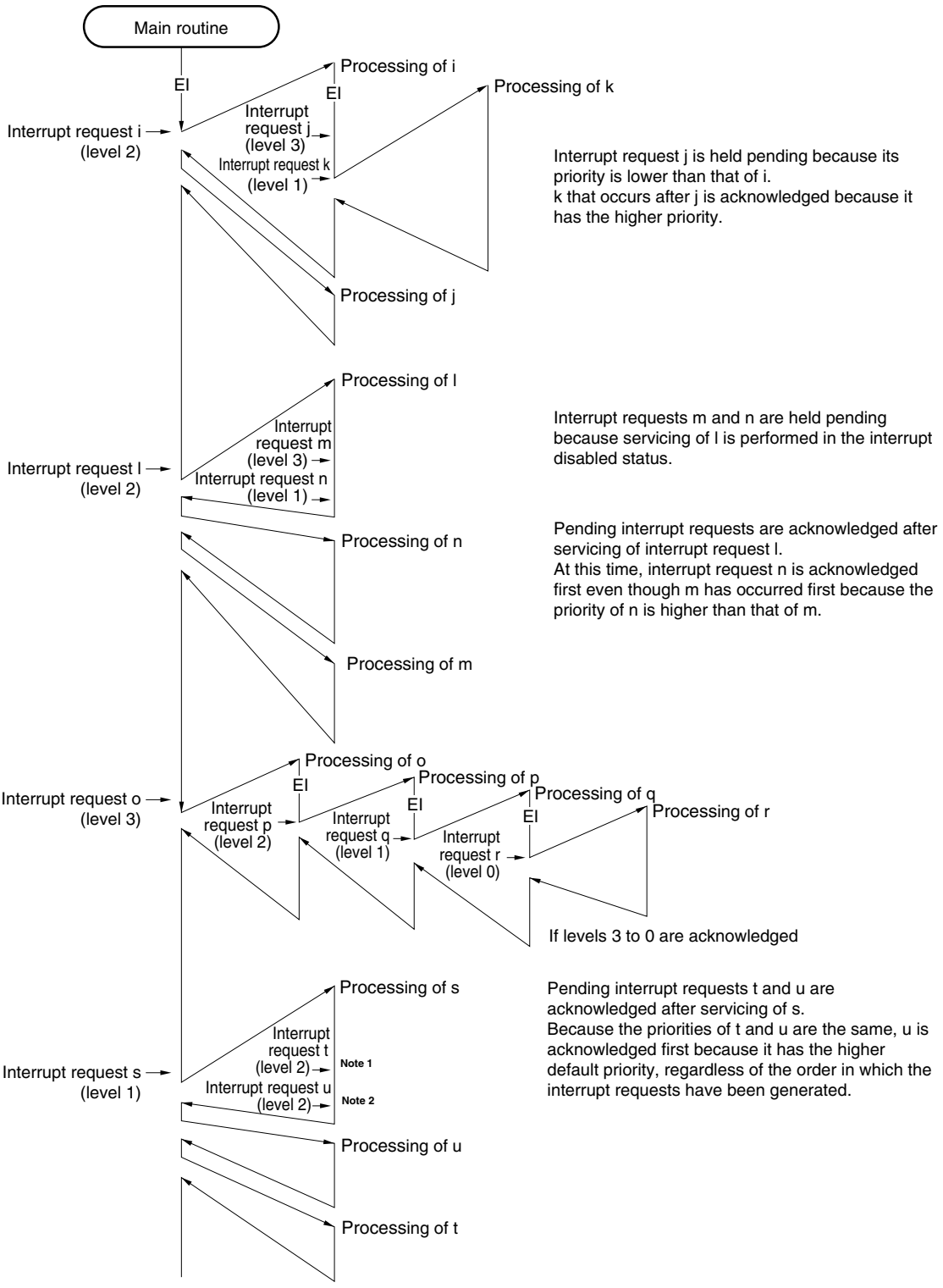


Caution: The values of the EIPC and EIPSW registers must be saved before executing multiple interrupts. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks: 1. **a** to **u** in the figure are the temporary names of interrupt requests shown for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt requests.

Figure 7-8: Example of Processing in which Another Interrupt Request Is Issued while an Interrupt is being Processed (2/2)

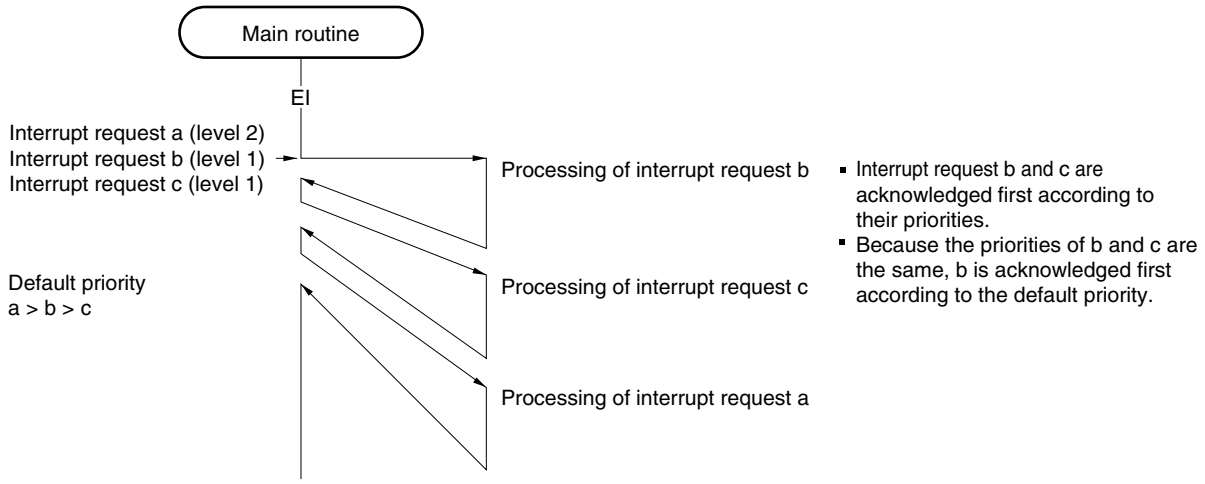


Notes: 1. Lower default priority

2. Higher default priority

Caution: The values of the EIPC and EIPSW registers must be saved before executing multiple interrupts. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Figure 7-9: Example of Processing Interrupt Requests Simultaneously Generated



Caution: The values of the EIPC and EIPSW registers must be saved before executing multiple interrupts. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

7.3.4 Interrupt control register (PICn)

An interrupt control register is assigned to each interrupt request (maskable interrupt) and sets the control conditions for each maskable interrupt request.
This register can be read/written in 8-bit or 1-bit units.

Figure 7-10: Interrupt Control Register (PICn)

After reset: 47H R/W Address: Refer to Table 7-2

	7	6	5	4	3	2	1	0
PICn	IFn	MKn	0	0	0	PRn2	PRn1	PRn0

IFn	Interrupt Request Flag n[Note]
0	Interrupt request is not issued
1	Interrupt request issued

MKn	Interrupt Mask Flag n
0	Interrupt servicing enabled
1	Interrupt servicing disabled (IFn flag hold pending)

PRn2	PRn1	PRn0	Interrupt Priority Specification n
0	0	0	Specifies level 0 (highest)
0	0	1	Specifies level 1
0	1	0	Specifies level 2
0	1	1	Specifies level 3
1	0	0	Specifies level 4
1	0	1	Specifies level 5
1	1	0	Specifies level 6
1	1	1	Specifies level 7 (lowest)

Note: Automatically reset by hardware when interrupt request is acknowledged.

Remark: n = 0 to 105 (see **Table 7-2: Addresses and Bits of Interrupt Control Registers**)

Table 7-2: Addresses and Bits of Interrupt Control Registers (1/3)

Address	Register	Bit								Associated Interrupt
		7	6	5	4	3	2	1	0	
FFFFF110H	PIC0	IF0	MK0	0	0	0	PR02	PR01	PR00	INTP0
FFFFF112H	PIC1	IF1	MK1	0	0	0	PR12	PR11	PR10	INTP1
FFFFF114H	PIC2	IF2	MK2	0	0	0	PR22	PR21	PR20	INTP2
FFFFF116H	PIC3	IF3	MK3	0	0	0	PR32	PR31	PR30	INTP3
FFFFF118H	PIC4	IF4	MK4	0	0	0	PR42	PR41	PR40	INTP4
FFFFF11AH	PIC5	IF5	MK5	0	0	0	PR52	PR51	PR50	INTP5
FFFFF11CH	PIC6	IF6	MK6	0	0	0	PR62	PR61	PR60	INTP6
FFFFF11EH	PIC7	IF7	MK7	0	0	0	PR72	PR71	PR70	INTP7
FFFFF120H	PIC8	IF8	MK8	0	0	0	PR82	PR81	PR80	INTP8
FFFFF122H	PIC9	IF9	MK9	0	0	0	PR92	PR91	PR90	INTP9
FFFFF124H	PIC10	IF10	MK10	0	0	0	PR102	PR101	PR100	INTP10
FFFFF126H	PIC11	IF11	MK11	0	0	0	PR112	PR111	PR110	INTP11
FFFFF128H	PIC12	IF12	MK12	0	0	0	PR122	PR121	PR120	INTP12
FFFFF12AH	PIC13	IF13	MK13	0	0	0	PR132	PR131	PR130	INTTR0OV
FFFFF12CH	PIC14	IF14	MK14	0	0	0	PR142	PR141	PR140	INTTR0CC0
FFFFF12EH	PIC15	IF15	MK15	0	0	0	PR152	PR151	PR150	INTTR0CC1
FFFFF130H	PIC16	IF16	MK16	0	0	0	PR162	PR161	PR160	INTTR0CC2
FFFFF132H	PIC17	IF17	MK17	0	0	0	PR172	PR171	PR170	INTTR0CC3
FFFFF134H	PIC18	IF18	MK18	0	0	0	PR182	PR181	PR180	INTTR0CC4
FFFFF136H	PIC19	IF19	MK19	0	0	0	PR192	PR191	PR190	INTTR0CC5
FFFFF138H	PIC20	IF20	MK20	0	0	0	PR202	PR201	PR200	INTTR0CD
FFFFF13AH	PIC21	IF21	MK21	0	0	0	PR212	PR211	PR210	INTTR0OD
FFFFF13CH	PIC22	IF22	MK22	0	0	0	PR222	PR221	PR220	INTTR0ER
FFFFF13EH	PIC23	IF23	MK23	0	0	0	PR232	PR231	PR230	INTTR1OV
FFFFF140H	PIC24	IF24	MK24	0	0	0	PR242	PR241	PR240	INTTR1CC0
FFFFF142H	PIC25	IF25	MK25	0	0	0	PR252	PR251	PR250	INTTR1CC1
FFFFF144H	PIC26	IF26	MK26	0	0	0	PR262	PR261	PR260	INTTR1CC2
FFFFF146H	PIC27	IF27	MK27	0	0	0	PR272	PR271	PR270	INTTR1CC3
FFFFF148H	PIC28	IF28	MK28	0	0	0	PR282	PR281	PR280	INTTR1CC4
FFFFF14AH	PIC29	IF29	MK29	0	0	0	PR292	PR291	PR290	INTTR1CC5
FFFFF14CH	PIC30	IF30	MK30	0	0	0	PR302	PR301	PR300	INTTR1CD
FFFFF14EH	PIC31	IF31	MK31	0	0	0	PR312	PR311	PR310	INTTR1OD
FFFFF150H	PIC32	IF32	MK32	0	0	0	PR322	PR321	PR320	INTTR1ER
FFFFF152H	PIC33	IF33	MK33	0	0	0	PR332	PR331	PR330	INTT0OV
FFFFF154H	PIC34	IF34	MK34	0	0	0	PR342	PR341	PR340	INTT0CC0
FFFFF156H	PIC35	IF35	MK35	0	0	0	PR352	PR351	PR350	INTT0CC1
FFFFF158H	PIC36	IF36	MK36	0	0	0	PR362	PR361	PR360	INTT0EC
FFFFF15AH	PIC37	IF37	MK37	0	0	0	PR372	PR371	PR370	INTT1OV
FFFFF15CH	PIC38	IF38	MK38	0	0	0	PR382	PR381	PR380	INTT1CC0
FFFFF15EH	PIC39	IF39	MK39	0	0	0	PR392	PR391	PR390	INTT1CC1

Table 7-2: Addresses and Bits of Interrupt Control Registers (2/3)

Address	Register	Bit								Associated Interrupt
		7	6	5	4	3	2	1	0	
FFFFF160H	PIC40	IF40	MK40	0	0	0	PR402	PR401	PR400	INTT1EC
FFFFF162H	PIC41	IF41	MK41	0	0	0	PR412	PR411	PR410	INTP0OV
FFFFF164H	PIC42	IF42	MK42	0	0	0	PR422	PR421	PR420	INTP0CC0
FFFFF166H	PIC43	IF43	MK43	0	0	0	PR432	PR431	PR430	INTP0CC1
FFFFF168H	PIC44	IF44	MK44	0	0	0	PR442	PR441	PR440	INTP1OV
FFFFF16AH	PIC45	IF45	MK45	0	0	0	PR452	PR451	PR450	INTP1CC0
FFFFF16CH	PIC46	IF46	MK46	0	0	0	PR462	PR461	PR460	INTP1CC1
FFFFF16EH	PIC47	IF47	MK47	0	0	0	PR472	PR471	PR470	INTP2OV
FFFFF170H	PIC48	IF48	MK48	0	0	0	PR482	PR481	PR480	INTP2CC0
FFFFF172H	PIC49	IF49	MK49	0	0	0	PR492	PR491	PR490	INTP2CC1
FFFFF174H	PIC50	IF50	MK50	0	0	0	PR502	PR501	PR500	INTP3OV
FFFFF176H	PIC51	IF51	MK51	0	0	0	PR512	PR511	PR510	INTP3CC0
FFFFF178H	PIC52	IF52	MK52	0	0	0	PR522	PR521	PR520	INTP3CC1
FFFFF17AH	PIC53	IF53	MK53	0	0	0	PR532	PR531	PR530	INTP4OV
FFFFF17CH	PIC54	IF54	MK54	0	0	0	PR542	PR541	PR540	INTP4CC0
FFFFF17EH	PIC55	IF55	MK55	0	0	0	PR552	PR551	PR550	INTP4CC1
FFFFF180H	PIC56	IF56	MK56	0	0	0	PR562	PR561	PR560	INTP5OV
FFFFF182H	PIC57	IF57	MK57	0	0	0	PR572	PR571	PR570	INTP5CC0
FFFFF184H	PIC58	IF58	MK58	0	0	0	PR582	PR581	PR580	INTP5CC1
FFFFF186H	PIC59	IF59	MK59	0	0	0	PR592	PR591	PR590	INTP6OV
FFFFF188H	PIC60	IF60	MK60	0	0	0	PR602	PR601	PR600	INTP6CC0
FFFFF18AH	PIC61	IF61	MK61	0	0	0	PR612	PR611	PR610	INTP6CC1
FFFFF18CH	PIC62	IF62	MK62	0	0	0	PR622	PR621	PR620	INTP7OV
FFFFF18EH	PIC63	IF63	MK63	0	0	0	PR632	PR631	PR630	INTP7CC0
FFFFF190H	PIC64	IF64	MK64	0	0	0	PR642	PR641	PR640	INTP7CC1
FFFFF192H	PIC65	IF65	MK65	0	0	0	PR652	PR651	PR650	INTP8OV
FFFFF194H	PIC66	IF66	MK66	0	0	0	PR662	PR661	PR660	INTP8CC0
FFFFF196H	PIC67	IF67	MK67	0	0	0	PR672	PR671	PR670	INTP8CC1
FFFFF198H	PIC68	IF68	MK68	0	0	0	PR682	PR681	PR680	INTBRG0
FFFFF19AH	PIC69	IF69	MK69	0	0	0	PR692	PR691	PR690	INTBRG1
FFFFF19CH	PIC70	IF70	MK70	0	0	0	PR702	PR701	PR700	INTBRG2
FFFFF19EH	PIC71	IF71	MK71	0	0	0	PR712	PR711	PR710	INTC0ERR
FFFFF1A0H	PIC72	IF72	MK72	0	0	0	PR722	PR721	PR720	INTC0WUP
FFFFF1A2H	PIC73	IF73	MK73	0	0	0	PR732	PR731	PR730	INTC0REC
FFFFF1A4H	PIC74	IF74	MK74	0	0	0	PR742	PR741	PR740	INTC0TRX
FFFFF1A6H	PIC75	IF75	MK75	0	0	0	PR752	PR751	PR750	INTC1ERR
FFFFF1A8H	PIC76	IF76	MK76	0	0	0	PR762	PR761	PR760	INTC1WUP
FFFFF1AAH	PIC77	IF77	MK77	0	0	0	PR772	PR771	PR770	INTC1REC
FFFFF1ACH	PIC78	IF78	MK78	0	0	0	PR782	PR781	PR780	INTC1TRX
FFFFF1AEH	PIC79	IF79	MK79	0	0	0	PR792	PR791	PR790	INTCB0T

Table 7-2: Addresses and Bits of Interrupt Control Registers (3/3)

Address	Register	Bit								Associated Interrupt
		7	6	5	4	3	2	1	0	
FFFFF1B0H	PIC80	IF80	MK80	0	0	0	PR802	PR801	PR800	INTCB0R
FFFFF1B2H	PIC81	IF81	MK81	0	0	0	PR812	PR811	PR810	INTCB0RE
FFFFF1B4H	PIC82	IF82	MK82	0	0	0	PR822	PR821	PR820	INTCB1T
FFFFF1B6H	PIC83	IF83	MK83	0	0	0	PR832	PR831	PR830	INTCB1R
FFFFF1B8H	PIC84	IF84	MK84	0	0	0	PR842	PR841	PR840	INTCB1RE
FFFFF1BAH	PIC85	IF85	MK85	0	0	0	PR852	PR851	PR850	INTC30OVF
FFFFF1BCH	PIC86	IF86	MK86	0	0	0	PR862	PR861	PR860	INTC30
FFFFF1BEH	PIC87	IF87	MK87	0	0	0	PR872	PR871	PR870	INTC31OVF
FFFFF1C0H	PIC88	IF88	MK88	0	0	0	PR882	PR881	PR880	INTC31
FFFFF1C2H	PIC89	IF89	MK89	0	0	0	PR892	PR891	PR890	INTUC0RE
FFFFF1C4H	PIC90	IF90	MK90	0	0	0	PR902	PR901	PR900	INTUC0R
FFFFF1C6H	PIC91	IF91	MK91	0	0	0	PR912	PR911	PR910	INTUC0T
FFFFF1C8H	PIC92	IF92	MK92	0	0	0	PR922	PR921	PR920	INTUC1RE
FFFFF1CAH	PIC93	IF93	MK93	0	0	0	PR932	PR931	PR930	INTUC1R
FFFFF1CCH	PIC94	IF94	MK94	0	0	0	PR942	PR941	PR940	INTUC1T
FFFFF1CEH	PIC95	IF95	MK95	0	0	0	PR952	PR951	PR950	INTAD0
FFFFF1D0H	PIC96	IF96	MK96	0	0	0	PR962	PR961	PR960	INTAD1
FFFFF1D2H	PIC97	IF97	MK97	0	0	0	PR972	PR971	PR970	INTCC10
FFFFF1D4H	PIC98	IF98	MK98	0	0	0	PR982	PR981	PR980	INTCC11
FFFFF1D6H	PIC99	IF99	MK99	0	0	0	PR992	PR991	PR990	INTCM10
FFFFF1D8H	PIC100	IF100	MK100	0	0	0	PR1002	PR1001	PR1000	INTCM11
FFFFF1DAH	PIC101	IF101	MK101	0	0	0	PR1012	PR1011	PR1010	INTOVF
FFFFF1DCH	PIC102	IF102	MK102	0	0	0	PR1022	PR1021	PR1020	INTUDF
FFFFF1DEH	PIC103	IF103	MK103	0	0	0	PR1032	PR1031	PR1030	INTDMA2
FFFFF1E0H	PIC104	IF104	MK104	0	0	0	PR1042	PR1041	PR1040	INTDMA3
FFFFF1E2H	PIC105	IF105	MK105	0	0	0	PR1052	PR1051	PR1050	INTPERR

7.3.5 Interrupt mask registers 0 to 6 (IMR0 to IMR6)

The IMR0 to IMR6 registers set the interrupt mask state for the maskable interrupts. The IMK0 to IMK104 bits are equivalent to the MKn bit in the corresponding PICn register.
The IMRm register (m = 0 to 6) can be read or written in 16-bit units.
If the higher 8 bits of the IMRm register are used as the IMRmH register and the lower 8 bits as the IMRmL register, these registers can be read or written in 8-bit or 1-bit units.
Reset input sets these registers to FFFFH.
Bits 15 to 9 of the IMR6 register (bits 7 to 1 of the IMR6H register) are fixed to 1. If these bits are not 1, the operation cannot be guaranteed.

Caution: The device file defines the MKn bit as a reserved word. If a bit is manipulated using the name of MKn, the contents of the PICn register, instead of the IMRm register, are rewritten (as a result, the contents of the IMRm register are also rewritten).

Figure 7-11: Interrupt Mask Registers 0 to 2 (IMR0 to IMR2)

After reset: FFFFH R/W Address: IMR0 FFFFF100H
IMR0L FFFFF100H, IMR0H FFFFF101H

IMR0

15	14	13	12	11	10	9	8
MK15	MK14	MK13	MK12	MK11	MK10	MK9	MK8
7	6	5	4	3	2	1	0
MK7	MK6	MK5	MK4	MK3	MK2	MK1	MK0

After reset: FFFFH R/W Address: IMR1 FFFFF102H
IMR1L FFFFF102H, IMR1H FFFFF103H

IMR1

15	14	13	12	11	10	9	8
MK31	MK30	MK29	MK28	MK27	MK26	MK25	MK24
7	6	5	4	3	2	1	0
MK23	MK22	MK21	MK20	MK19	MK18	MK17	MK16

After reset: FFFFH R/W Address: IMR2 FFFFF104H
IMR2L FFFFF104H, IMR2H FFFFF105H

IMR2

15	14	13	12	11	10	9	8
MK47	MK46	MK45	MK44	MK43	MK42	MK41	MK40
7	6	5	4	3	2	1	0
MK39	MK38	MK37	MK36	MK35	MK34	MK33	MK32

IMKn	Interrupt Mask Flag
0	Enable interrupt servicing
1	Disable interrupt servicing (pending)

Remark: n = 0 to 105 (see **Table 7-1**)

Figure 7-12: Interrupt Mask Registers 3 to 6 (IMR3 to IMR6)

After reset: FFFFH R/W Address: IMR3 FFFFF106H
IMR3L FFFFF106H, IMR3H FFFFF107H

	15	14	13	12	11	10	9	8
IMR3	MK63	MK62	MK61	MK60	MK59	MK58	MK57	MK56
	MK55	MK54	MK53	MK52	MK51	MK50	MK49	MK48

After reset: FFFFH R/W Address: IMR4 FFFFF108H
IMR4L FFFFF108H, IMR4H FFFFF109H

	15	14	13	12	11	10	9	8
IMR4	MK79	MK78	MK77	MK76	MK75	MK74	MK73	MK72
	MK71	MK70	MK69	MK68	MK67	MK66	MK65	MK64

After reset: FFFFH R/W Address: IMR5 FFFFF10AH
IMR5L FFFFF10AH, IMR5H FFFFF10BH

	15	14	13	12	11	10	9	8
IMR5	MK95	MK94	MK93	MK92	MK91	MK90	MK89	MK88
	MK87	MK86	MK85	MK84	MK83	MK82	MK81	MK80

After reset: FFFFH R/W Address: IMR6 FFFFF10CH
IMR6L FFFFF10CH, IMR6H FFFFF10DH

	15	14	13	12	11	10	9	8
IMR6	1	1	1	1	1	1	MK105	MK104
	MK103	MK102	MK101	MK100	MK99	MK98	MK97	MK96

IMKn	Interrupt Mask Flag
0	Enable interrupt servicing
1	Disable interrupt servicing (pending)

Remark: n = 0 to 105 (see **Table 7-1**)

7.3.6 In-service priority register (ISPR)

The ISPR register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request is acknowledged, the bit of this register corresponding to the priority level of that interrupt request is set to 1 and remains set while the interrupt is serviced.
When the RETI instruction is executed, the bit corresponding to the interrupt request having the highest priority is automatically reset to 0 by hardware. However, it is not reset to 0 when execution is returned from non-maskable interrupt servicing or exception processing.
Reset input clears this register to 00H.
This register is read-only, in 8-bit or 1-bit units.

Caution: In the interrupt enabled (EI) state, if an interrupt is acknowledged during the reading of the ISPR register, the value of the ISPR register may be read after the bit is set (1) by this interrupt acknowledgment. To read the value of the ISPR register properly before interrupt acknowledgment, read it in the interrupt disabled (DI) state.

Figure 7-13: Interrupt Service Priority Register (ISPR)

After reset: 00H R Address: FFFFF1FAH

	7	6	5	4	3	2	1	0
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

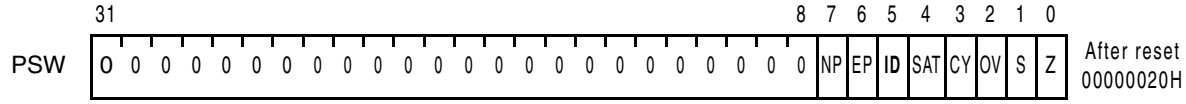
ISPRn	Priority of Interrupt Currently Being Acknowledged
0	Interrupt request with priority n is not acknowledged
1	Interrupt request with priority n is being acknowledged

Remark: n = 0 to 7 (priority level)

7.3.7 Maskable interrupt status flag (ID)

The ID flag is bit 5 of the PSW and controls the maskable interrupt's operating state, and stores control information regarding enabling or disabling of interrupt requests.

Figure 7-14: Maskable interrupt status flag (ID)



ID	Maskable Interrupt Servicing Specification[Note]
0	Maskable interrupt request acknowledgment enabled
1	Maskable interrupt request acknowledgment disabled (pending)

Note: Interrupt disable flag (ID) function

- This flag is set to 1 by the DI instruction and reset to 0 by the EI instruction. Its value is also modified by the RETI instruction or LDSR instruction when referencing the PSW.
- Non-maskable interrupt and exceptions are acknowledged regardless of this flag. When a maskable interrupt is acknowledged, the ID flag is automatically set to 1 by hardware.
- The interrupt request generated during the acknowledgement disabled period (ID = 1) can be acknowledged when the IFn bit of the interrupt control register PICn is set to 1, and the ID flag is reset to 0.

7.3.8 Interrupt trigger mode selection

The valid edge of the maskable external interrupt input pin (INTPn) can be selected by program (n = 0 to 12).
The edge that can be selected as the valid edge is one of the following.

- Rising edge
- Falling edge
- Both, the rising and falling edges

The edge-detected INTPn signal becomes an interrupt source.
The valid edge is specified by interrupt mode registers 0 to 3 (INTM0 to INTM3)

(1) Interrupt mode register 0 (INTM0)

The behaviour of the external interrupt input pins INTP0 to INTP2 can be specified by the interrupt mode register 0 (INTM0).

The INTM0 register can be read/written in 8-bit or 1-bit units.

Figure 7-15: Interrupt Mode Register 0 (INTM0)

After reset: 00H R/W Address: FFFFF880H

	7	6	5	4	3	2	1	0
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	ESN1	ESN0

ES21	ES20	Valid Edge Specification of INTP2 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

ES11	ES10	Valid Edge Specification of INTP1 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

ES01	ES00	Valid Edge Specification of INTP0 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

ESN1	ESN0	Valid Edge Specification of NMI pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

Caution: Changing the state of interrupt mode configuration registers ESn0/ESn1 may trigger an unintended interrupt event for the respective interrupt channels. Be sure to mask the respective interrupt channel and clear the interrupt status flag after changing the bits ESn0/ESn1 of the interrupt channel (n = 0 to 2).

(2) Interrupt mode register 1 (INTM1)

The behaviour of the external interrupt input pins INTP3 to INTP6 can be specified by the interrupt mode register 1 (INTM1).

The INTM1 register can be read/written in 8-bit or 1-bit units.

Figure 7-16: Interrupt Mode Register 1 (INTM1)

After reset: 00H R/W Address: FFFFF882H

	7	6	5	4	3	2	1	0
INTM1	ES61	ES60	ES51	ES50	ES41	ES40	ES31	ES30

ES61	ES60	Valid Edge Specification of INTP6 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

ES51	ES50	Valid Edge Specification of INTP5 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

ES41	ES40	Valid Edge Specification of INTP4 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

ES31	ES30	Valid Edge Specification of INTP4 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

Caution: Changing the state of interrupt mode configuration registers ESn0/ESn1 may trigger an unintended interrupt event for the respective interrupt channels. Be sure to mask the respective interrupt channel and clear the interrupt status flag after changing the bits ESn0/ESn1 of the interrupt channel (n = 3 to 6).

(3) Interrupt mode register 2 (INTM2)

The behaviour of the external interrupt input pins INTP7 to INTP10 can be specified by the interrupt mode register 2 (INTM2).
The INTM2 register can be read/written in 8-bit or 1-bit units.

Figure 7-17: Interrupt Mode Register 2 (INTM2)

After reset: 00H R/W Address: FFFFF882H

	7	6	5	4	3	2	1	0
INTM1	ES101	ES100	ES91	ES90	ES81	ES80	ES71	ES70

ES101	ES100	Valid Edge Specification of INTP10 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

ES91	ES90	Valid Edge Specification of INTP9 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

ES81	ES80	Valid Edge Specification of INTP8 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

ES71	ES70	Valid Edge Specification of INTP7 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

Caution: Changing the state of interrupt mode configuration registers ESn0/ESn1 may trigger an unintended interrupt event for the respective interrupt channels. Be sure to mask the respective interrupt channel and clear the interrupt status flag after changing the bits ESn0/ESn1 of the interrupt channel (n = 7 to 10).

(4) Interrupt mode register 3 (INTM3)

The behaviour of the external interrupt input pins INTP11 and INTP12 can be specified by the interrupt mode register 3 (INTM3).

The INTM3 register can be read/written in 8-bit or 1-bit units.

Figure 7-18: Interrupt Mode Register 3 (INTM3)

After reset: 00H R/W Address: FFFFF882H

	7	6	5	4	3	2	1	0
INTM1	0	0	0	0	ES121	ES120	ES111	ES110

ES121	ES120	Valid Edge Specification of INTP12 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

ES111	ES110	Valid Edge Specification of INTP11 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

Caution: Changing the state of interrupt mode configuration registers ESn0/ESn1 may trigger an unintended interrupt event for the respective interrupt channels. Be sure to mask the respective interrupt channel and clear the interrupt status flag after changing the bits ESn0/ESn1 of the interrupt channel (n = 11, 12).

7.4 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and is always accepted.

For details of the instruction function, refer to the **V850 Family User's Manual Architecture.**

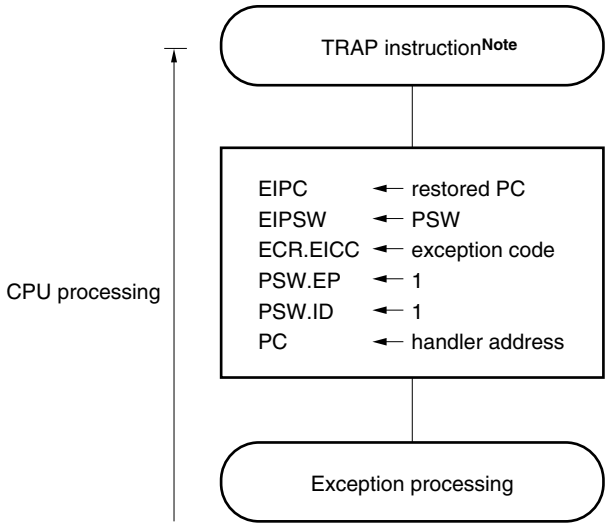
7.4.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine:

<1> Saves the current PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
<4> Sets the EP and ID bits of PSW.
<5> Loads the handler address (00000040H or 00000050H) of the software exception routine in the PC, and transfers control.

The processing of a software exception is shown below.

Figure 7-19: Software Exception Processing



Note: TRAP instruction format: TRAP vector (the vector is a value from 0 to 1FH.)

The handler address is determined by the TRAP instruction's operand (vector). If the vector is 0 to 0FH, it becomes 00000040H, and if the vector is 10H to 1FH, it becomes 00000050H.

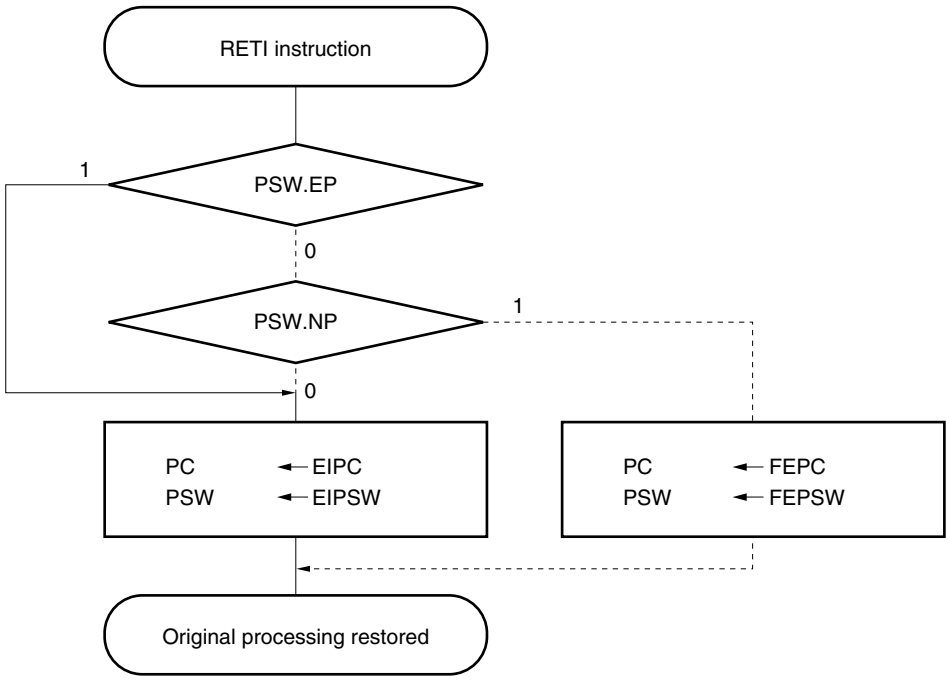
7.4.2 Restore

Recovery from software exception processing is carried out by the RETI instruction.
By executing the RETI instruction, the CPU carries out the following processing and shifts control to the restored PC's address.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the PSW.EP bit is 1.
<2> Transfers control to the address of the restored PC and PSW.

The processing of the RETI instruction is shown below.

Figure 7-20: RETI Instruction Processing



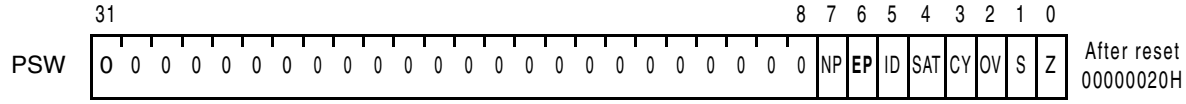
Caution: When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during the software exception process, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark: The solid line shows the CPU processing flow.

7.4.3 Exception status flag (EP)

The EP flag is bit 6 of the PSW, and is a status flag used to indicate that exception processing is in progress. This flag is set when an exception occurs.

Figure 7-21: Exception Status Flag (EP)



EP	Exception Processing Status
0	Exception processing not in progress
1	Exception processing in progress

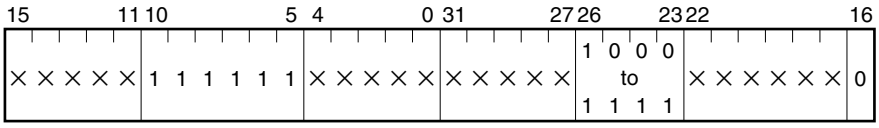
7.5 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850E/PH2, an illegal opcode trap (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

7.5.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B, a sub-opcode (bits 26 to 23) of 1000B to 1111B, and a sub-opcode (bit 16) of 0B. An exception trap is generated when an instruction applicable to this illegal instruction is executed.

Figure 7-22: Illegal Opcode



Caution: Caution Since it is possible that this instruction may be assigned to an illegal opcode in the future, it is recommended that it not be used.

Remark: x: don't care

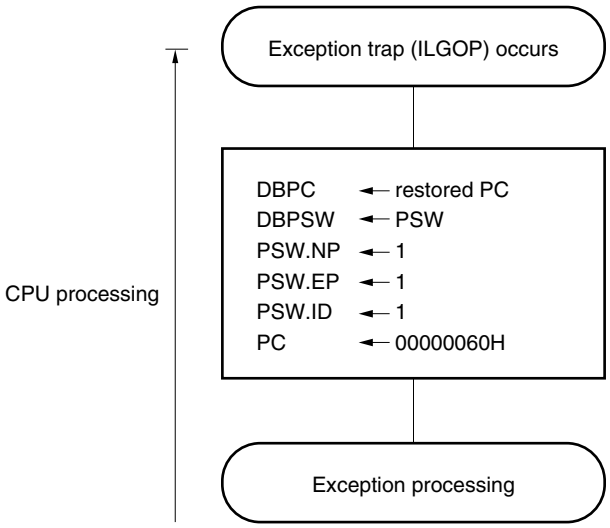
(1) Operation

If an exception trap occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to DBPC.
<2> Saves the current PSW to DBPSW.
<3> Sets the PSW.NP, PSW.EP, and PSW.ID bits.
<4> Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

Figure 7-23 illustrates the processing of the exception trap.

Figure 7-23: Exception Trap Processing



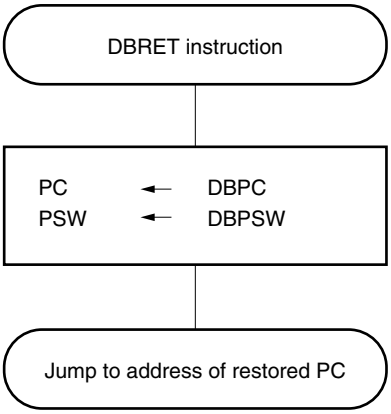
(2) Restore

Recovery from an exception trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

Figure 7-24 illustrates the restore processing from an exception trap.

Figure 7-24: Restore Processing from Exception Trap



7.6 Periods in Which CPU Does Not Acknowledge Interrupts

The CPU acknowledges an interrupt while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt request non-sample instruction and the next instruction (interrupt is held pending).
The interrupt request non-sample instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- The store instruction for the command register (PRCMD)
- The store, or bit manipulation instructions excluding the tst1 instruction for the following interrupt-related registers:
  - Interrupt control register (PICn)
  - Interrupt mask registers 0 to 3 (IMR0 to IMR3)

Remark: n = 0 to 105 (see **Table 7-2, “Addresses and Bits of Interrupt Control Registers,” on page 225**)

Chapter 8 Clock Generator

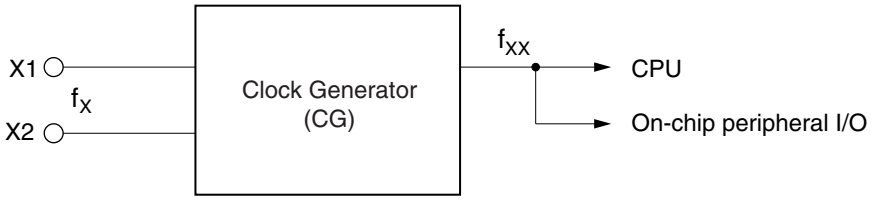
The clock generator (CG) generates and controls the internal system clock (f_{XX}) that is supplied to each internal unit, such as the CPU.

8.1 Features

- Multiplier function using a phase locked loop (PLL) synthesizer ($f_{XX} = 4 \times f_X$)
  - Crystal frequency: f_X = 16 MHz
  - Internal system clock: f_{XX} = 64 MHz
- Power saving mode: HALT mode

8.2 Configuration

Figure 8-1: Clock Generator



Remark: f_X: External resonator or external clock frequency
f_{XX}: Internal system clock

An external resonator or crystal is connected to X1 and X2 pins, whose frequency is multiplied by the PLL synthesizer. By this an internal system clock (f_{XX}) is generated that is 4 times the frequency (f_X) of the external resonator or crystal.
The clock controller enables PLL automatically and starts clock supply to the system after oscillation stabilization time has passed.

Internal System Clock Frequency (f_{XX})	External Resonator or Crystal Frequency (f_X)
64.000 MHz	16.0000 MHz

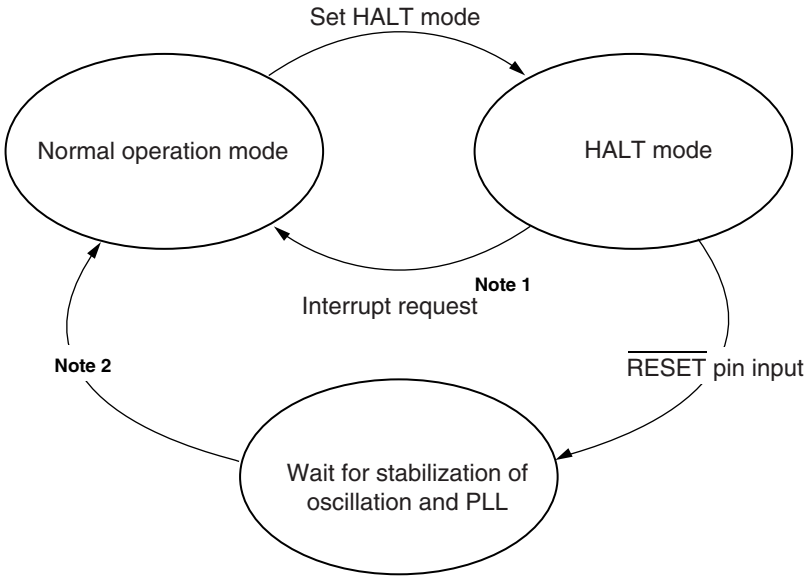
8.3 Power Save Control

8.3.1 Overview

The power save function of V850E/PH2 supports the HALT mode only. In this mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but the CPU's operation clock stops. Since the supply of clocks to on-chip peripheral functions other than the CPU continues,
operation continues. The power consumption of the overall system can be reduced by intermittent operation that is achieved due to a combination of HALT mode and normal operation mode.
The system is switched to HALT mode by a specific instruction (the HALT instruction).

Figure 8-2 shows the operation of the clock generator in normal operation mode and HALT mode.

Figure 8-2: Power Save Mode State Transition Diagram



Notes: 1. Non-maskable interrupt request signal (NMI) or unmasked maskable interrupt request signal.

2. The oscillation stabilization time is necessary after release of reset because the PLL is initialized by a reset. The stabilization time is determined by default.

8.3.2 HALT mode

(1) Setting and operation status

The HALT mode is set when a dedicated instruction (HALT) is executed in the normal operation mode.

When HALT mode is set, clock supply is stopped to the CPU only. The clock generator and PLL continue operating. Clock supply to the other on-chip peripheral functions continues.

As a result, program execution is stopped, and the internal RAM retains the contents before the HALT mode was set. The on-chip peripheral functions that are independent of instruction processing by the CPU continue operating.

Table 18-3 shows the operation status in the HALT mode.

The average power consumption of the system can be reduced by using the HALT mode in combination with the normal operation mode for intermittent operation.

Cautions: 1. Insert five or more NOP instructions after the HALT instruction.

2. If the HALT instruction is executed while an interrupt request is being held pending, the HALT mode is set but is released immediately by the pending interrupt request.

Table 8-1: Operation Status in HALT Mode

Function	Operation Status
Clock generator	Operating
Internal system clock (f_{XX})	Supplied
CPU	Stopped
DMA	Operating
Interrupt controller	Operating
Ports	Maintained
On-chip peripheral I/O (excluding ports)	Operating
Internal data	All internal data such as CPU registers, states, data, and the contents of internal RAM are retained in the state they were before HALT mode was set.
A0 to A21	Operating
D0 to D31	
$\overline{RD}$	
$\overline{WR}$	
$\overline{BEN0}$ to $\overline{BEN3}$	
$\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$, $\overline{CS4}$	
BCLK, STST, STNXT	
$\overline{WAIT}$	

(2) Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request signal (NMI), an unmasked maskable interrupt request signal, or RESET pin input.
After the HALT mode has been released, the normal operation mode is restored.

(a) Releasing HALT mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The HALT mode is released by a non-maskable interrupt request signal (INTWDT) or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

- If an interrupt request signal with a priority lower than or same as the interrupt currently being serviced is generated, the HALT mode is released, but the newly generated interrupt request signal is not acknowledged. The interrupt request signal itself is retained.
- If an interrupt request signal with a priority higher than that of the interrupt currently being serviced is issued (including a non-maskable interrupt request signal), the HALT mode is released and that interrupt request signal is acknowledged.

Table 8-2: Operation After Releasing HALT Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address	
Unmasked maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(b) Releasing HALT mode by RESET pin input or WDTRES signal generation

The same operation as the normal reset operation is performed.

Chapter 9 16-Bit Timer/Event Counter P

9.1 Features

Timer P (TMP) is a 16-bit timer/event counter that can be used in various ways.
TMP can perform the following operations.

- PWM output
- Interval timer
- External event counter (operation not possible when clock is stopped)
- One-shot pulse output
- Pulse width measurement

9.2 Function Outline

- Capture trigger input signal × 2
- External trigger input signal × 1
- Clock select × 8
- External event count input × 1
- Readable counter × 1
- Capture/compare reload register × 2
- Capture/compare match interrupt × 2
- Timer output (TOPn0, TOPn1) × 2

9.3 Configuration

TMP includes the following hardware.

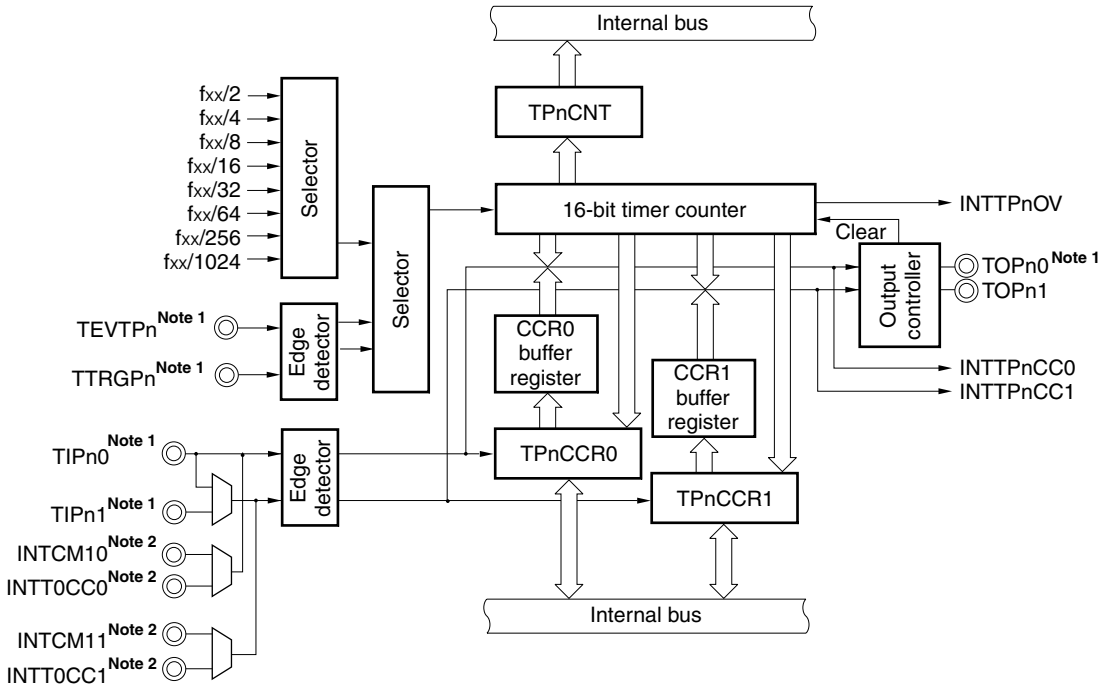
Table 9-1: Configuration of TMP0 to TMP8

Item	Configuration
Timer register	16-bit counter
Registers	TMPn capture/compare registers 0, 1 (TPnCCR0, TPnCCR1) TMPn counter register (TPnCNT) CCR0 buffer register, CCR1 buffer register
Timer input	2 × 8 (TIPm0, TIPm1, TTRGPm, TEVTPm)**Note**
Timer output	2 × 8 (TOPm0, TOPm1)**Note** 1 × 1 (TOP81)
Control registers	TMPn control registers 0, 1 (TPnCTL0, TPnCTL1) TMPn I/O control registers 0 to 2 (TPnIOC0 to TPnIOC2) TMPn option registers 0, 1 (TPnOPT0, TPnOPT1)

Note: TIPm0 and TIPm1 captures inputs are shared with external trigger inputs TTRGPm, and external event inputs TEVTPm, and the corresponding TOPm0 and TOPm1 outputs.

Remark: n = 0 to 8
m = n for n= 0 to 7

Figure 9-1: Block Diagram of Timer P



Notes: 1. External pin is not available for TMP8.

2. Internal signal inputs (INTTT0CC0 and INTTT0CC1 of TMT0, or INTCM10 and INTCM11 of TMENC1) available on TMP8 only. (ref. to **9.4 (9) TMP input control register 2 (TPIC2)**).

(1) TMPn capture/compare register 0 (TPnCCR0)

The TPnCCR0 register is a 16-bit register that functions both as a capture register and as a compare register.
Whether this register functions as a capture register or as a compare register can be controlled with the TPnCCS0 bit of the TPnOPT0 register, but only in the free-running mode.
In the pulse width measurement mode, this register can be used as a dedicated capture register (the compare function cannot be used.)
In modes other than the free-running mode and pulse width measurement mode, this register is used as a dedicated compare register.
In the initial setting, the TPnCCR0 register is a compare register.
This register can be read or written in 16-bit units.
Reset input clears this register to 0000H.

Figure 9-2: TMPn Capture/Compare Register 0 (TPnCCR0)

After reset: 0000H R/W Address: TP0CCR0 FFFFF606H, TP1CCR0 FFFFF616H, TP2CCR0 FFFFF626H, TP3CCR0 FFFFF636H, TP4CCR0 FFFFF646H, TP5CCR0 FFFFF656H, TP6CCR0 FFFFF666H, TP7CCR0 FFFFF676H, TP8CCR0 FFFFF686H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR0																

(n = 0 to 8)

(a) Use as compare register

TPnCCR0 can be rewritten when TPnCE = 1
The timing at which the TPnCCR0 rewrite values become valid when TPnCE = 1 is as follows.

TMP Operation Mode	Method of Writing TPnCCR0 Register
PWM mode, external trigger pulse output mode	Reload
Free-running mode, external event count mode, one-shot pulse output mode, interval timer mode	Anytime write
Pulse width measurement mode	Cannot be used because dedicated capture register

(b) Use as capture register

- **TMP0 to TMP7**
  The counter value is saved to TPnCCR0 upon capture trigger (TIPn0) input edge detection.

- **TMP8**
  Since TMP8 has no external input pin, the capture function can only be used internally for capturing the interrupt signal (INTTT0CC0 of TMT0, or INTCM10 of TMENC1) specified by the TPIC22 bit of TPIC2 register (ref. to **9.4 (9) TMP input control register 2 (TPIC2)**).

(2) TMPn capture/compare register 1 (TPnCCR1)

The TPnCCR1 register is a 16-bit register that functions both as a capture register and as a compare register.
Whether this register functions as a capture register or as a compare register can be controlled with the TPnCCS1 bit of the TPnOPT0 register, but only in the free-running mode.
In the pulse width measurement mode, this register can be used as a dedicated capture register (the compare function cannot be used.)
In modes other than the free-running mode and pulse width measurement mode, this register is used as a dedicated compare register.
In the initial setting, the TPnCCR1 register is a reload register.
This register can be read or written in 16-bit units.
Reset input clears this register to 0000H.

Figure 9-3: TMPn Capture/Compare Register 1 (TPnCCR1)

After reset: 0000H R/W Address: TP0CCR1 FFFFF608H, TP1CCR1 FFFFF618H, TP2CCR1 FFFFF628H, TP3CCR1 FFFFF638H, TP4CCR1 FFFFF648H, TP5CCR1 FFFFF658H, TP6CCR1 FFFFF668H, TP7CCR1 FFFFF678H, TP8CCR1 FFFFF688H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR1 (n = 0 to 8)																

(a) Use as compare register

TPnCCR1 can be rewritten when TPnCE = 1
The timing at which the TPnCCR1 rewrite values become valid when TPnCE = 1 is as follows.

TMP Operation Mode	Method of Writing TPnCCR0 Register
PWM mode, external trigger pulse output mode	Reload
Free-running mode, external event count mode, one-shot pulse output mode, interval timer mode	Anytime write
Pulse width measurement mode	Cannot be used because dedicated capture register

(b) Use as capture register

- **TMP0 to TMP7**
  The counter value is saved to TPnCCR1 upon capture trigger (TIPn1) input edge detection.

- **TMP8**
  Since TMP8 has no external input pin, the capture function can only be used internally for capturing the interrupt signal (INTTT0CC1 of TMT0, or INTCM11 of TMENC1) specified by the TPIC22 bit of TPIC2 register (ref. to **9.4 (9) TMP input control register 2 (TPIC2)**).

(3) TMPn counter register (TPnCNT)

The TPnCNT register is a read buffer register that can read 16-bit counter values.
This register is read-only, in 16-bit units.
Reset input clears this register to 0000H, as the TPnCE bit is cleared to 0.

Figure 9-4: TMPn Counter Register (TPnCNT)

After reset: 0000H R Address: TP0CNT FFFFF60AH, TP1CNT FFFFF61AH, TP2CNT FFFFF62AH, TP3CNT FFFFF63AH, TP4CNT FFFFF64AH, TP5CNT FFFFF65AH, TP6CNT FFFFF66AH, TP7CNT FFFFF67AH, TP8CNT FFFFF68AH

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

(n = 0 to 8)

Remark: The value of the TPnCNT register is cleared to 0000H when the TPnCE bit = 0. If the TPnCNT register is read at this time, the value of the 16-bit counter (FFFFH) is not read, but 0000H is read.

9.4 Control Registers

(1) TMPn control register 0 (TPnCTL0)

The TPnCTL0 register is an 8-bit register that controls the operation of timer P.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.
The same value can always be written to the TPnCTL0 register by software.

Figure 9-5: TMPn Control Register 0 (TPnCTL0)

After reset: 00H R/W Address: TP0CTL0 FFFFF600H, TP1CTL0 FFFFF610H, TP2CTL0 FFFFF620H, TP3CTL0 FFFFF630H, TP4CTL0 FFFFF640H, TP5CTL0 FFFFF650H, TP6CTL0 FFFFF660H, TP7CTL0 FFFFF670H, TP8CTL0 FFFFF680H

	7	6	5	4	3	2	1	0
TPnCTL0 (n = 0 to 8)	TPnCE	0	0	0	0	TPnCKS2	TPnCKS1	TPnCKS0

TPnCE	Timer Pn Operation Control
0	Internal operating clock operation disabled (TMPn reset asynchronously)
1	Internal operating clock operation enabled

- Internal operating clock control and TMPn asynchronous reset are performed with the TPnCE bit. When the TPnCE bit is cleared to 0, the internal operating clock of TMPn stops (fixed to low level) and TMPn is reset asynchronously.
- When the TPnCE bit is set to 1, the internal operating clock is enabled and count-up operation starts within 2 input clocks after the TPnCE bit was set to 1

TPnCKS2	TPnCKS1	TPnCKS0	Internal Count Clock Selection
0	0	0	$f_{XX}/2$
0	0	1	$f_{XX}/4$
0	1	0	$f_{XX}/8$
0	1	1	$f_{XX}/16$
1	0	0	$f_{XX}/32$
1	0	1	$f_{XX}/64$
1	1	0	$f_{XX}/256$
1	1	1	$f_{XX}/1024$

Caution: Set the TPnCKS2 to TPnCKS0 bits when TPnCE = 0. When the value of the TPnCE bit is changed from 0 to 1, the TPnCKS2 to TPnCKS0 bits can be set simultaneously.

Remark: n = 0 to 8

(2) TMPn control register 1 (TPnCTL1)

The TPnCTL1 register is an 8-bit register that controls the operation of timer P.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Figure 9-6: TMPn Control Register 1 (TPnCTL1) (1/2)

After reset: 00H R/W Address: TP0CTL1 FFFFF601H, TP1CTL1 FFFFF611H, TP2CTL1 FFFFF621H, TP3CTL1 FFFFF631H, TP4CTL1 FFFFF641H, TP5CTL1 FFFFF651H, TP6CTL1 FFFFF661H, TP7CTL1 FFFFF671H, TP8CTL1 FFFFF681H

	7	6	5	4	3	2	1	0
TPnCTL1 (n = 0 to 8)	TPnSYE	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

TPnSYE	Synchronous Mode Selection
0	Timer Pn operates in single operation mode
1	Timer Pn operates in synchronous operation mode[Note]

- This bit supports synchronous operation of two or more timer P.
- Two groups of timers exist, which can be synchronized: TMP0 to TMP3 with TMP0 as master, and TMP4 to TMP7 with TMP4 as master.

Note: Synchronous operation mode is not available for TMP8 (n = 8).

TPnEST	Software Trigger Control
0	No operation
1	In one-shot pulse mode: One-shot pulse software trigger In external trigger pulse output mode: Pulse output software trigger

- The TPnEST bit functions as a software trigger in the one-shot pulse mode and the external trigger pulse output mode[Note 1], if it is set to 1 when TPnCE = 1. Therefore, be sure to set TPnEST to 1 after setting TPnCE to 1.
- TTRGPn pin is used as the external trigger input of TMPn.[Note 2]
- The read value of the TPnEST bit is always 0.

Notes: 1. The TRnEST bit is invalid even if it is controlled in any other mode.

2. External trigger input pin is not available for TMP8 (n = 8)

Cautions: 1. Always clear the TPnSYE bit for the master timers TMP0 and TMP4.

2. Always clear the TPnSYE bit for TMP8. Do not operate TMP8 in synchronous mode.

Remark: n = 0 to 8

Figure 9-6: TMPn Control Register 1 (TPnCTL1) (2/2)

TPnEEE	Count Clock Selection
0	Use the internal clock (selected by bits TPnCKS2 to TPnCKS0)
1	Use external clock input (TEVTPn input edge)[Note]

- When TPnEEE = 1 (external clock input TEVTPn), the valid edge is specified by bits TPnEES1 and TPnEES0.

Note: External clock input pin is not available for TMP8 (n = 8).

TPnMD2	TPnMD1	TPnMD0	Timer Mode Selection
0	0	0	Interval timer mode [Note 1, 2]
0	0	1	External event count mode [Note 1, 2, 3]
0	1	0	External trigger pulse output mode [Note 2, 3]
0	1	1	One-shot pulse mode [Note 2]
1	0	0	PWM mode [Note 2]
1	0	1	Free-running mode
1	1	0	Pulse width measurement mode [Note 1, 2]
1	1	1	Setting prohibited

Notes: 1. Setting prohibited for TMP0 and TMP4, when synchronous operation function is enabled (TPnSYE = 1).

2. Setting prohibited for TMP1 to TMP3, and TMP5 to TMP7, when synchronous operation function is enabled (TPnSYE = 1).

3. Setting prohibited for TMP8.

Cautions: 1. Rewrite the TPnEEE and TPnMD2 to TPnMD0 bits only when TPnCE = 0. (The same value can be written when TPnCE = 1.) The operation is not guaranteed if rewriting is performed when TPnCE = 1. If rewriting was mistakenly performed, set TPnCE = 0 and then set the bits again.

2. Set TP8EEE bit of the TR0CTL1 register always to 0, because TMP8 does not incorporate an external clock input. In case of TP8EEE = 1 operation of TMP8 is not guaranteed.

Remark: n = 0 to 8

(3) TMPn I/O control register 0 (TPnIOC0)

The TPnIOC0 register is an 8-bit register that controls the timer output (TOPn0, TOPn1).
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Figure 9-7: TMPn I/O Control Register 0 (TPnIOC0)

After reset: 00H R/W Address: TP0IOC0 FFFFF602H, TP1IOC0 FFFFF612H, TP2IOC0 FFFFF622H, TP3IOC0 FFFFF632H, TP4IOC0 FFFFF642H, TP5IOC0 FFFFF652H, TP6IOC0 FFFFF662H, TP7IOC0 FFFFF672H, TP8IOC0 FFFFF682H

	7	6	5	4	3	2	1	0
TPnIOC0 (n = 0 to 8)	0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0

TPnOL1	Timer Output Level Setting (TOPn1 pin)
0	Normal output (Low level, when output is inactive.)
1	Inverted output (High level, when output is inactive.)

TPnOE1	Timer Output Control (TOPn1 pin)
0	Timer output prohibited (TOPn1 pin output is fixed to inactive level.)
1	Timer output enabled (A pulse can be output from the TOPn1 pin.)

TPnOL0	Timer Output Level Setting (TOPn0 pin)[Note]
0	Normal output (Low level, when output is inactive.)
1	Inverted output (High level, when output is inactive.)

TPnOE0	Timer Output Control (TOPn0 pin)[Note]
0	Timer output prohibited (TOPn0 pin output is fixed to inactive level.)
1	Timer output enabled (A pulse can be output from the TOPn0 pin.)

Note: TOPn0 output pin is not available for TMP8.

Caution: Rewrite the TPnOL1, TPnOE1, TPnOl0, and TPnOE0 bits only when TPnCE = 0. (The same value can be written when TPnCE = 1.) If rewriting was mistakenly performed, set TPnCE = 0 and then set the bits again.

Remark: n = 0 to 8

(4) TMPn I/O control register 1 (TPnIOC1)

The TPnIOC1 register is an 8-bit register that controls the valid edge for the external input signals (TIPn0, TIPn1).

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

Reset input clears this register to 00H.

Figure 9-8: TMPn I/O Control Register 1 (TPnIOC1)

After reset: 00H R/W Address: TP0IOC1 FFFFF603H, TP1IOC1 FFFFF613H, TP2IOC1 FFFFF623H, TP3IOC1 FFFFF633H, TP4IOC1 FFFFF643H, TP5IOC1 FFFFF653H, TP6IOC1 FFFFF663H, TP7IOC1 FFFFF673H, TP8IOC1 FFFFF683H

	7	6	5	4	3	2	1	0
TPnIOC1 (n = 0 to 8)	0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0

TPnIS3	TPnIS2	Capture Input (TIPn1) Valid Edge Setting[Note]
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection

TPnIS1	TPnIS0	Capture Input (TIPn0) Valid Edge Setting[Note]
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection

Note: TIPn0 and TIPn1 input pins are not available for TMP8. These inputs are only connected internally to capture the interrupt signals INTTT0CC0 and INTT0CC1 of TMT0, or INTCM10 and INTCM11 of TMENC1, specified by the TPIC22 bit of TPIC2 register (ref. to **9.4 (9) TMP input control register 2 (TPIC2)**).

Cautions: 1. Rewrite the TPnIS3 to TPnIS0 bits only when TPnCE = 0. (The same value can be written when TPnCE = 1.) If rewriting was mistakenly performed, set TPnCE = 0 and then set the bits again.

2. The TPnIS3 to TPnIS0 bits are valid only in the free-running mode and the pulse width measurement mode. In all other modes, a capture operation is not possible.

Remark: n = 0 to 8

(5) TMPn I/O control register 2 (TPnIOC2)

The TPnIOC2 register is an 8-bit register that controls the valid edge of the external event count input signal (TEVTPn) and external trigger input signal (TTRGPn).
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Figure 9-9: TMPn I/O Control Register 2 (TPnIOC2)

After reset: 00H R/W Address: TP0IOC2 FFFFF604H, TP1IOC2 FFFFF614H, TP2IOC2 FFFFF624H, TP3IOC2 FFFFF634H, TP4IOC2 FFFFF644H, TP5IOC2 FFFFF654H, TP6IOC2 FFFFF664H, TP7IOC2 FFFFF674H, TP8IOC2 FFFFF684H

	7	6	5	4	3	2	1	0
TPnIOC2	0	0	0	0	TPnEES1	TPnEES0	TPnETS1	TPnETS0

(n = 0 to 7)

TP1EES1	TP1EES0	External Event Counter Input (TEVTPn) Valid Edge Setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection

TP1ETS1	TP1ETS0	External Trigger Input (TTRGPn) Valid Edge Setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection

Cautions: 1. Rewrite the TPnEES1, TPnEES0, TPnEST1, and TPnEST0 bits only when TPnCE = 0. (The same value can be written when TPnCE = 1.) If rewriting was mistakenly performed, set TPnCE = 0 and then set the bits again.

2. The TPnEES1 and TPnEES0 bits are valid only when TPnEEE = 1 or when the external event count mode (TPnMD2 to TPnMD0 = 001B of the TPnCTL1 register) has been set.

Remark: n = 0 to 7

(6) TMPn option register 0 (TPnOPT0)

The TPnOPT0 register is an 8-bit register used to set the capture/compare operation and detect overflow.

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

Reset input clears this register to 00H.

Figure 9-10: TMPn Option Register 0 (TPnOPT0)

After reset: 00H R/W Address: TP0OPT0 FFFFF605H, TP1OPT0 FFFFF615H, TP2OPT0 FFFFF625H, TP3OPT0 FFFFF635H, TP4OPT0 FFFFF645H, TP5OPT0 FFFFF655H, TP6OPT0 FFFFF665H, TP7OPT0 FFFFF675H, TP8OPT0 FFFFF685H

	7	6	5	4	3	2	1	0
TPnOPT0 (n = 0 to 8)	0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF

TPnCCS1	TPnCCR1 register capture/compare selection
0	Compare register selection
1	Capture register selection
The TPnCCS1 bit settings are valid only in the free-running mode.	

TPnCCS0	TPnCCR0 register capture/compare selection
0	Compare register selection
1	Capture register selection
The TPnCCS0 bit settings are valid only in the free-running mode.	

TPnOVF	Timer P overflow detection flag
0	No overflow occurrence after timer restart or flag reset
1	Overflow occurrence
• The TPnOVF flag is set when the 16-bit counter value overflows from FFFFH to 0000H in the free-running mode or the pulse measurement mode. • An interrupt request signal (INTTPnOV) is generated at the same time that the TPnOVF flag is set (1). The INTTPnOV signal is not generated in modes other than the free-running mode or the pulse measurement mode. • The TPnOVF flag is not cleared even when the TPnOVF flag and the TPnOPT0 register are read. • The TPnOVF flag can be both read and written, but only reset (0) is accepted. Writing 1 has no influence on the operation of timer P.	

Caution: Rewrite the TPnCCS1 and TPnCCS0 bits only when TPnCE = 0. (The same value can be written when TPnCE = 1.) If rewriting was mistakenly performed, set TPnCE = 0 and then set the bits again.

Remark: n = 0 to 8

(7) TMP input control register 0 (TPIC0)

The TPIC0 register is an 8-bit register that controls the external input pin source of the capture register 1 of TMP0 to TMP3.

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

Reset input clears this register to 00H.

Figure 9-11: TMPn Input Control Register 0 (TPIC0)

After reset: 00H R/W Address: FFFFF6F0H

	7	6	5	4	3	2	1	0
TPIC0	0	0	0	0	TPIC03	TPIC02	TPIC01	TPIC00

TPIC03	TP3CCR1 Register Capture Source Input Selection
0	Capture source input is pin P17/TIP31
1	Capture source input is pin P16/TIP30

TPIC02	TP2CCR1 Register Capture Source Input Selection
0	Capture source input is pin P15/TIP21
1	Capture source input is pin P14/TIP20

TPIC01	TP1CCR1 Register Capture Source Input Selection
0	Capture source input is pin P13/TIP11
1	Capture source input is pin P12/TIP10

TPIC00	TP0CCR1 Register Capture Source Input Selection
0	Capture source input is pin P11/TIP01
1	Capture source input is pin P10/TIP00

(8) TMP input control register 1 (TPIC1)

The TPIC1 register is an 8-bit register that controls the external input pin source of the capture register 1 of TMP4 to TMP7, as well as the internal time trigger source from the FCAN controllers of both capture registers 0 and 1 of TMP7.
This register can be read or written in 8-bit units.
Reset input clears this register to 00H.

Figure 9-12: TMP Input Control Register 1 (TPIC1)

After reset: 00H R/W Address: FFFFF6F2H

	7	6	5	4	3	2	1	0
TPIC1	0	0	TIP15	TIP14	TPIC13	TPIC12	TPIC11	TPIC10

TPIC15	TIPC14	TIPC13	Capture Source Input Selection of	
			TP7CCR0	TP7CCR1
0	0	0	Pin P26/TIP70	Pin P27/TIP71
0	0	1		Pin P26/TIP70
0	1	0	FCAN0 time trigger	Pin P27/TIP71
0	1	1		Pin P26/TIP70
1	0	0	Pin P26/TIP70	FCAN1 time trigger
1	0	1		
1	1	0	FCAN0 time trigger	
1	1	1		

TPIC12	TP6CCR1 Register Capture Source Input Selection
0	Capture source input is pin P25/TIP61
1	Capture source input is pin P24/TIP60

TPIC11	TP5CCR1 Register Capture Source Input Selection
0	Capture source input is pin P23/TIP51
1	Capture source input is pin P22/TIP50

TPIC10	TP4CCR1 Register Capture Source Input Selection
0	Capture source input is pin P21/TIP41
1	Capture source input is pin P20/TIP40

(9) TMP input control register 2 (TPIC2)

The TPIC2 register is an 8-bit register that controls the external input pin source of the capture register 1 of TMT0 and TMT1, as well as the internal source of both capture registers 0 and 1 of TMP8.

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

Reset input clears this register to 00H.

Figure 9-13: TMP Input Control Register 1 (TPIC1)

After reset: 00H R/W Address: FFFFF6F4H

	7	6	5	4	3	2	1	0
TPIC2	0	0	0	0	0	TPIC22	TPIC21	TPIC20

TIPC22	Capture Source Input Selection of	
	TP8CCR0	TP8CCR1
0	INTTT0CC0 signal of TMT0	INTTT1CC1 signal of TMT0
1	INTCM10 signal of TMENC1	INTCM11 signal of TMENC1

TPIC21	TT1CCR1 Register Capture Source Input Selection
0	Capture source input is pin P74/TIT11
1	Capture source input is pin P73/TIT10

TPIC20	TT0CCR1 Register Capture Source Input Selection
0	Capture source input is pin P71/TIT01
1	Capture source input is pin P70/TIT00

9.5 Operation

Timer P can perform the following operations.

Operation	TPnEST (Software Trigger Bit)	TTRGPn0 (External Trigger Input)	Capture/Compare Mode	Compare Register Rewriting Method
Interval timer mode	Invalid	Invalid	Compare only	Anytime rewrite
External event count mode Note 1	Invalid	Invalid	Compare only	Anytime rewrite
External trigger pulse output mode Note 2	Valid	Valid	Compare only	Reload
One-shot pulse output mode Note 2	Valid	Valid	Compare only	Anytime rewrite
PWM mode	Invalid	Invalid	Compare only	Reload
Free-running mode	Invalid	Invalid	Capture/compare selectable	Anytime rewrite
Pulse width measurement mode Note 2	Invalid	Invalid	Capture only	Not applicable

Notes: 1. To use the external event count function, specify that the edge of the capture input TIPn1 or TIPn0 respectively, shared with event input TEVTPn is not detected (by clearing the TPSn3, TPSn2 bits or TPnIS1, TPnIS0 bits of the TPnIOC1 register respectively to “00B”) (n = 0 to 7).

2. When using the external trigger pulse output mode, one-shot pulse mode, and pulse width measurement mode, select a count clock (by clearing the TPnEEE bit of the TPnCTL1 register to 0).

Remark: n = 0 to 7

9.5.1 Anytime rewrite and reload

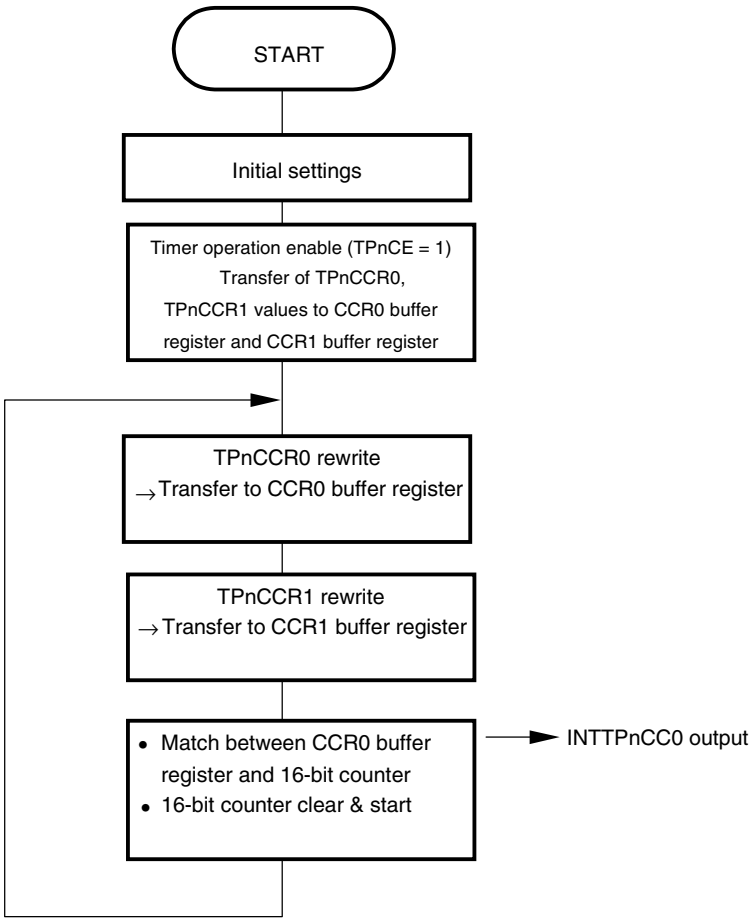
TPnCCR0 and TPnCCR1 register rewrite is possible for timer P during timer operation (TPnCE = 1), but the write method (anytime rewrite, reload) differs depending on the mode.

(1) Anytime rewrite

When the TPnCCRm register is written during timer operation, the write data is transferred at that time to the CCRm buffer register and used as the 16-bit counter comparison value.

Remark: n = 0 to 8
m = 0, 1

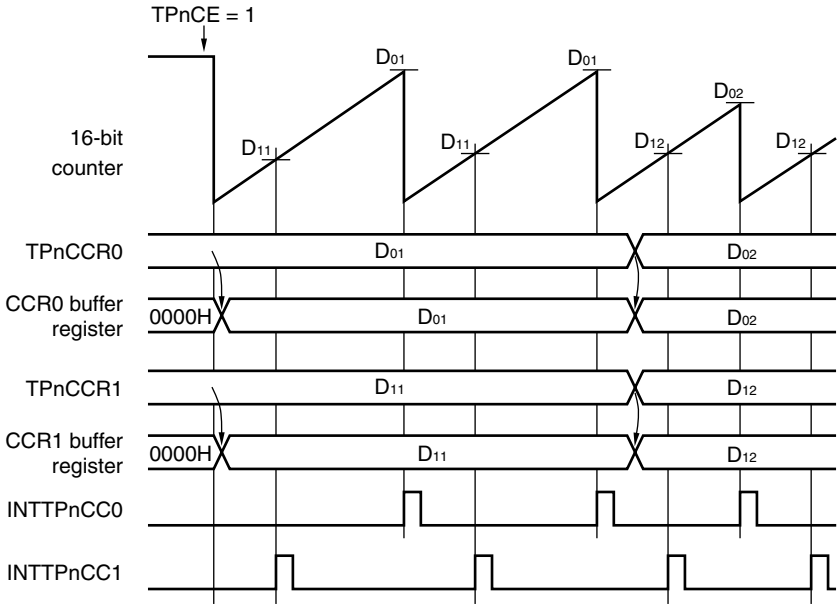
Figure 9-14: Basic Operation Flow for Anytime Write



Remarks: 1. The above flowchart illustrates an example of the operation in the interval timer mode.

2. n = 0 to 8

Figure 9-15: Timing Diagram for Anytime Write



Remarks: 1. D_{01}, D_{02}: Setting values of TPnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Setting values of TPnCCR1 register (0000H to FFFFH)

2. The above timing chart illustrates an example of the operation in the interval timer mode.

3. n = 0 to 8

(2) Reload method (Batch Rewrite)

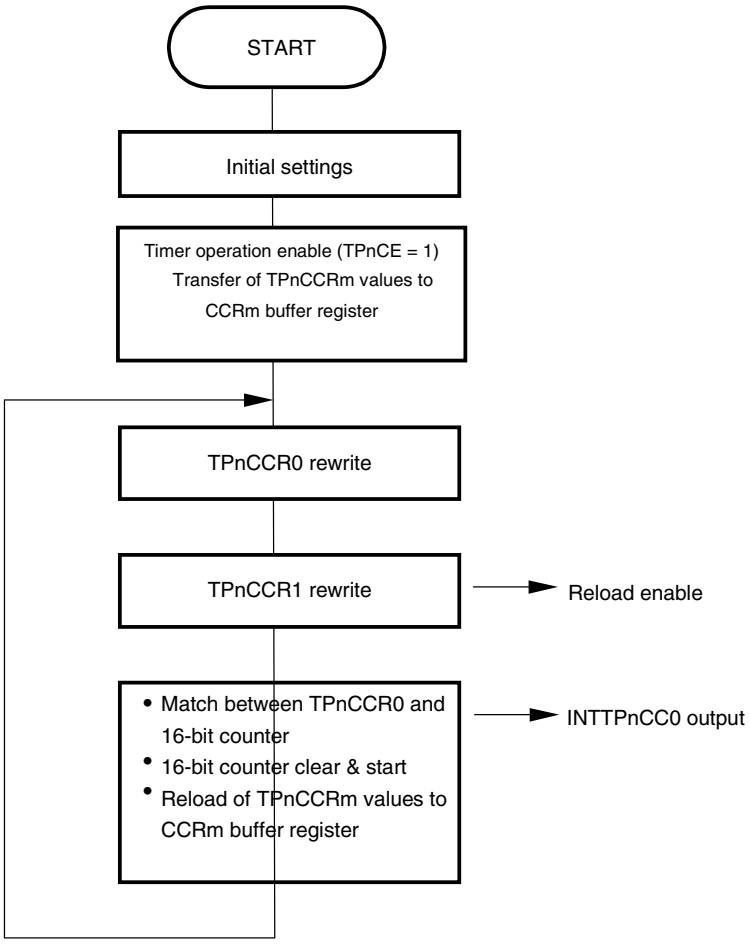
When the TPnCCR0 and TPnCCR1 registers are written during timer operation via the CCRm buffer register, the write data is used as the 16-bit counter comparison value. The TPnCCR0 register and the TPnCCR1 register can be rewritten when TPnCE = 1.

In order for the setting value when the TPnCCR0 register and the TPnCCR1 register are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be reloaded to the CCRm buffer register), it is necessary to rewrite TPnCCR0 and then write to the TPnCCR1 register before the 16-bit counter value and the TPnCCR0 register value match.

Thereafter, the values of the TPnCCR0 and the TPnCCR1 register are reloaded upon TPnCCR0 register match.

Whether to enable or disable the next reload timing is controlled by writing to the TPnCCR1 register. Thus even when wishing only to rewrite the value of the TPnCCR0 register, also write the same value to the TPnCCR1 register.

Figure 9-16: Basic Operation Flow for Reload (Batch Rewrite)

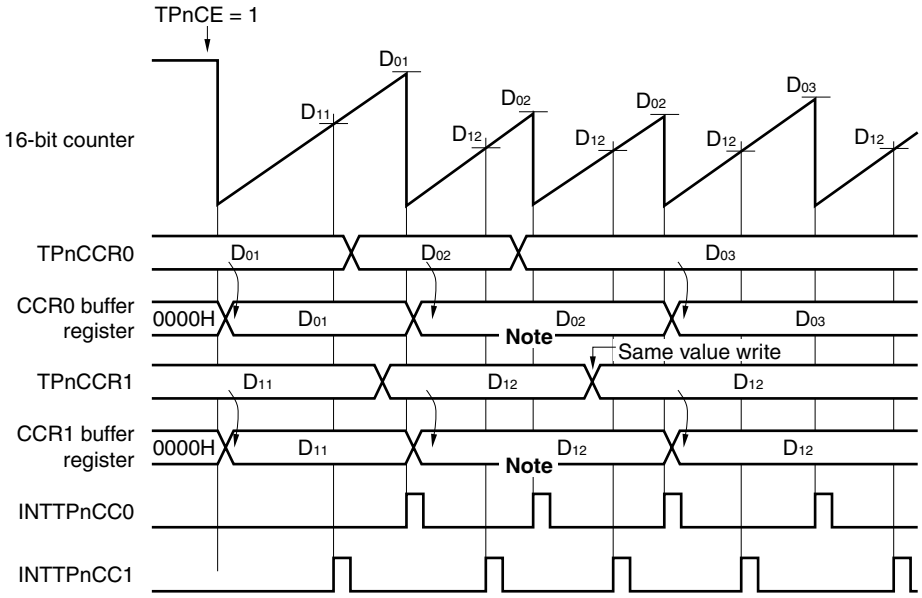


Caution: Writing to the TPnCCR1 register includes enabling of reload. Thus, rewrite the TPnCCR1 register after rewriting the TPnCCR0 register.

Remarks: 1. The above flowchart illustrates an example of the operation in the PWM mode.

2. n = 0 to 8
m = 0, 1

Figure 9-17: Timing Chart for Reload



Note: Reload is not performed because the TPnCCR1 register was not rewritten.

Remarks: **1.** D_{01}, D_{02}, D_{03}: Setting value of TPnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Setting value of TPnCCR1 register (0000H to FFFFH)

2. The above timing chart illustrates the operation in the PWM mode as an example.

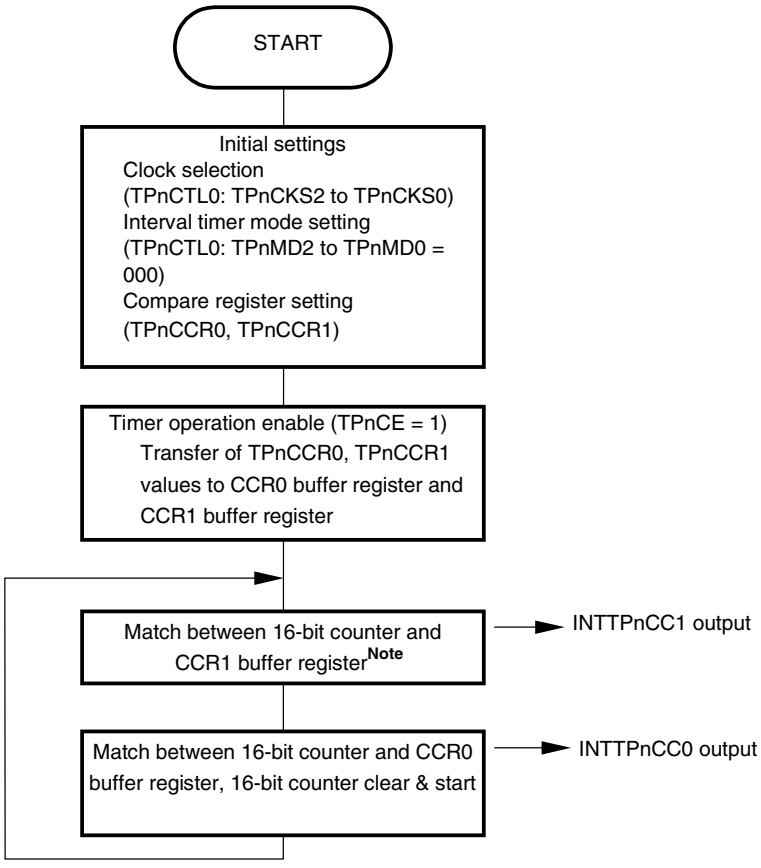
3. n = 0 to 8

9.5.2 Interval timer mode (TPnMD2 to TPnMD0 = 000B)

In the interval timer mode, an interrupt request signal (INTTPnCC0) is output upon a match between the setting value of the TPnCCR0 register and the value of the 16-bit counter, and the 16-bit counter is cleared. The TPnCCR0 register can be rewritten when TPnCE = 1, and when a value is set to the TPnCCR0 register with a write instruction from the CPU, it is transferred to the CCR0 buffer register through anytime write, and is used as the value for comparison with the 16-bit counter value.
In the interval timer mode, the 16-bit counter is cleared only upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.
16-bit counter clearing using the TPnCCR1 register is not performed. However, the setting value of the TPnCCR1 register is transferred to the CCR1 buffer register and compared with the value of the 16-bit counter, and an interrupt request (INTTPnCC1) is output if these values match.
Moreover, TOPnm pin output is also possible by setting the TPnOEm bit to 1.
When the TPnCCR1 register is not used, it is recommended to set FFFFH as the setting value for the TPnCCR1 register.

Remark: n = 0 to 8
m = 0, 1

Figure 9-18: Flowchart of Basic Operation in Interval Timer Mode

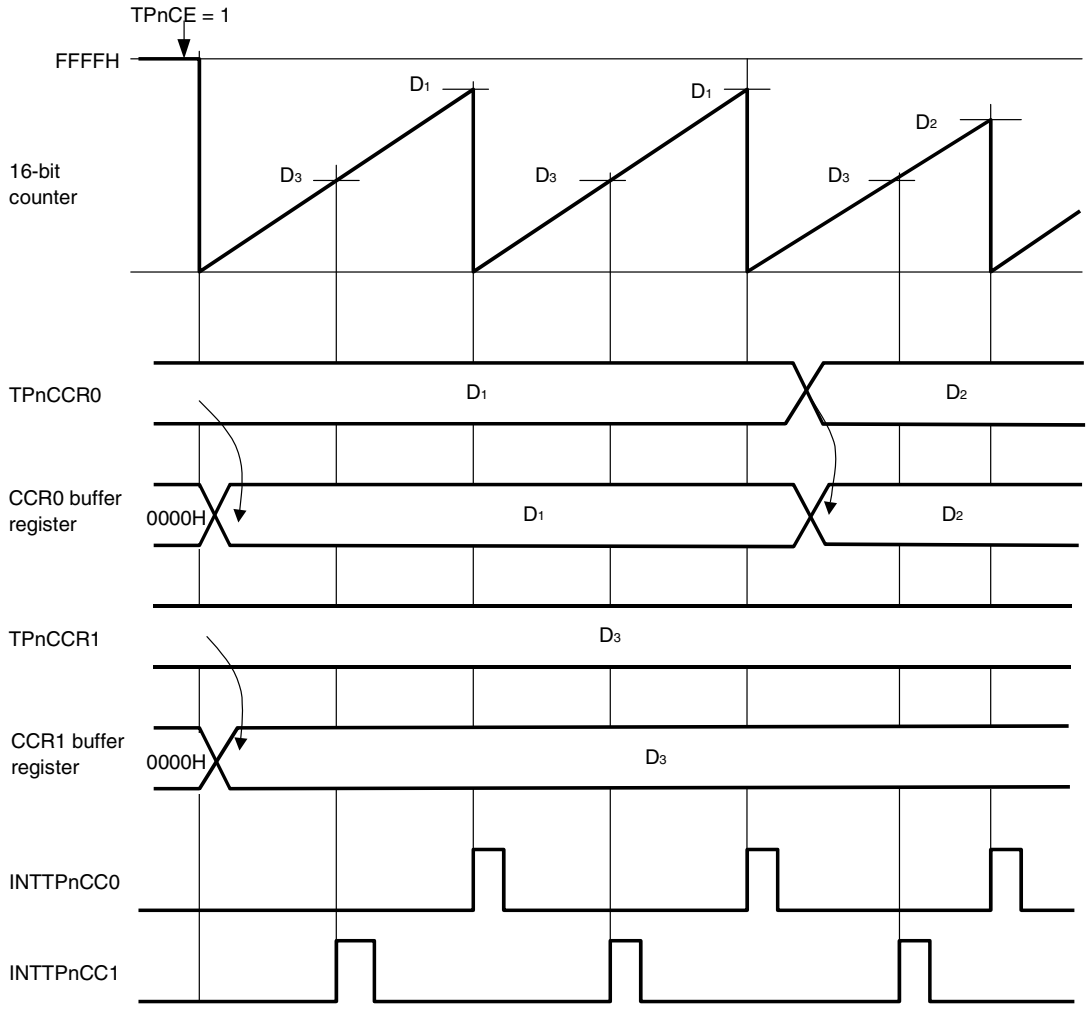


Note: The 16-bit counter is not cleared upon a match between the 16-bit counter and TPnCCR1.

Remark: n = 0 to 8

Figure 9-19: Basic Operation Timing in Interval Timer Mode (1/2)

(a) $D_1 > D_2 > D_3$; rewrite of TPnCCR0 register only; no TOPn0, TOPn1 output



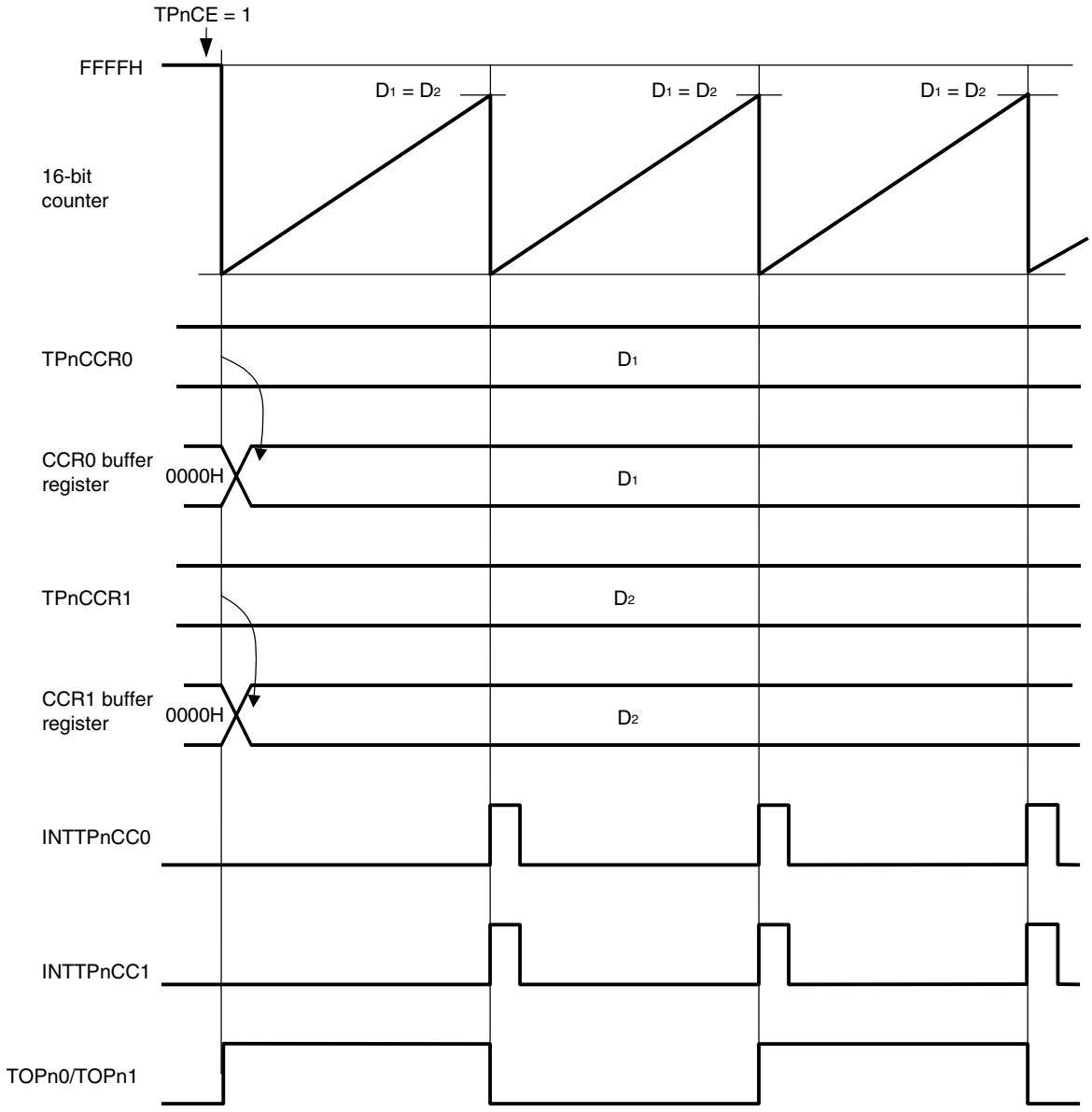
Remarks: **1.** D_1, D_2: Setting values of TPnCCR0 register (0000H to FFFFH)
D_3: Setting value of TPnCCR1 register (0000H to FFFFH)

2. Interval time = (Dn + 1) × (count clock cycle)

3. n = 0 to 8

Figure 9-19: Basic Operation Timing in Interval Timer Mode (2/2)

(b) $D_1 = D_2$; no TPnCCR0, TPnCCR1 rewrite; TOPn1 output



Remarks: 1. D_1: Setting value of TPnCCR0 register (0000H to FFFFH)
D_2: Setting value of TPnCCR1 register (0000H to FFFFH)

2. Interval time = (Dn + 1) × (count clock cycle)

3. n = 0 to 8

9.5.3 External event count mode (TPnMD2 to TPnMD0 = 001B)

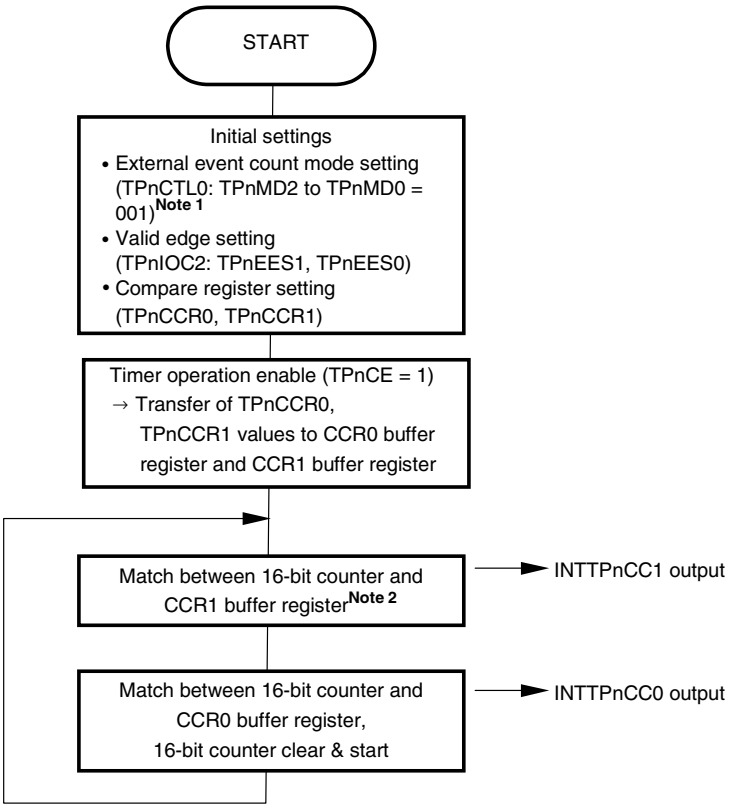
In the external event count mode, external event count input (TEVTPn pin input) is used as a count-up signal. When the external event count mode is set, count-up is performed using external event count input (TEVTPn pin input), regardless of the setting of the TPnEEE bit of the TPnCTL0 register.
In the external event count mode, a match interrupt request (INTTPnCC0) is output upon a match between the setting value of the TPnCCR0 register and the value of the 16-bit counter, and the 16-bit counter is cleared.
When a value is set to the TPnCCR0 register with a write instruction from the CPU, it is transferred to the CCR0 buffer register through anytime write, and is used as the value for comparison with the 16-bit counter value.
In the external event count mode, the 16-bit counter is cleared only upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.
16-bit counter clearing using the TPnCCR1 register is not performed. However, the setting value of the TPnCCR1 register is transferred to the CCR1 buffer register and compared with the value of the 16-bit counter, and an interrupt request (INTTPnCC1) is output if these values match.
Moreover, TOPnm pin output is also possible by setting the TPnOEm bit to 1.
The TPnCCR0 register can be rewritten when TPnCE = 1. When the TPnCCR1 register is not used, it is recommended to set FFFFH as the setting value for the TPnCCR1 register.

Cautions: 1. In external event count mode, when the setting value of the TRnCCR0 register is set to m, the number of TEVTPn pin input edge detection times is m+1.

2. In external event count mode, do not set TPnCCR0 register to 0000H.

Remark: n = 0 to 7
m = 0, 1

Figure 9-20: Flowchart of Basic Operation in External Event Count Mode



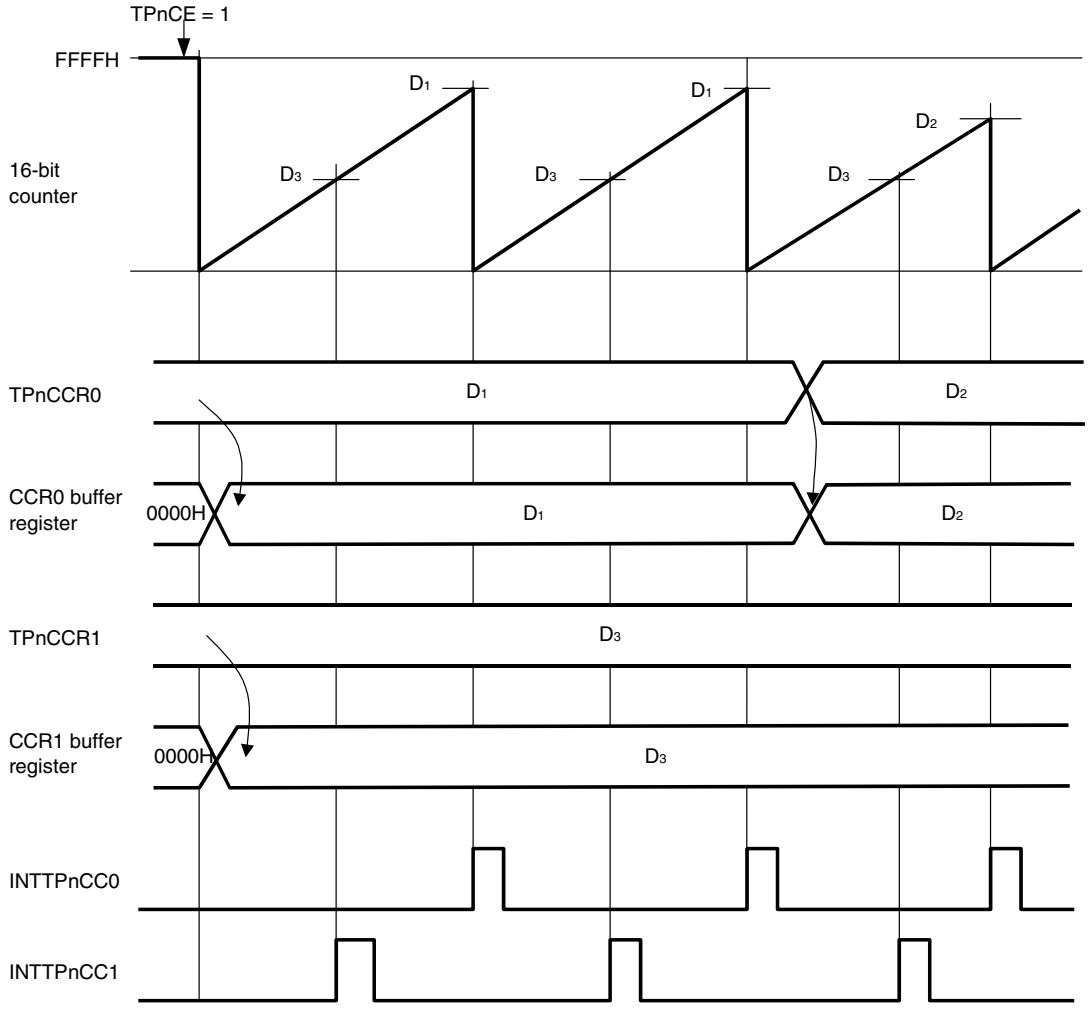
Notes: 1. Selection of the TPnEEE bit has no influence.

2. The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.

Remark: n = 0 to 7

Figure 9-21: Basic Operation Timing in External Event Count Mode (1/2)

(a) D1 > D2 > D3; rewrite of TPnCCR0 only; no TOPn0, TOPn1 output



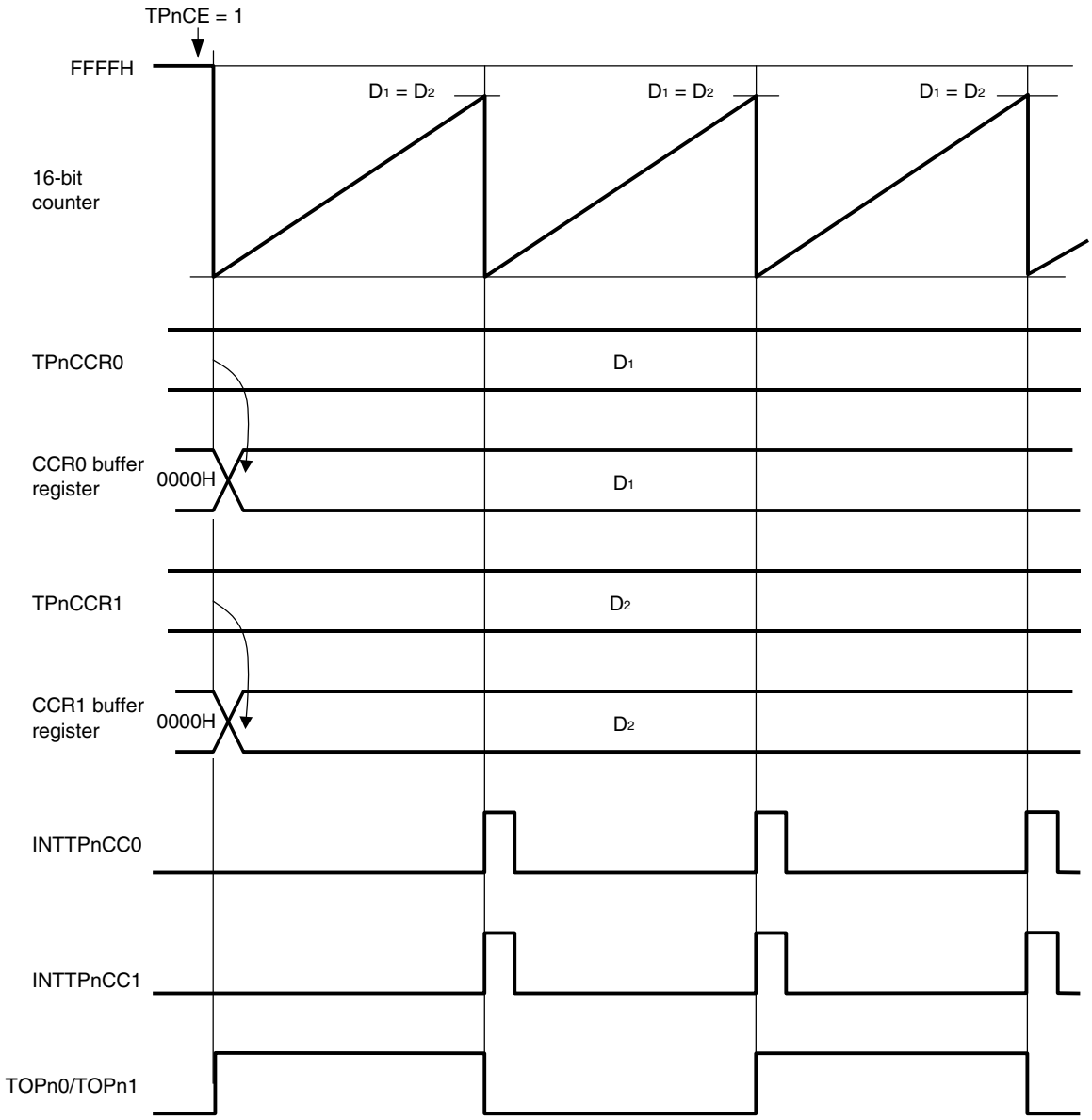
Remarks: 1. D_1, D_2: Setting values of TPnCCR0 register (0000H to FFFFH)
D_3: Setting value of TPnCCR1 register (0000H to FFFFH)

2. Event count = (Dn + 1)

3. n = 0 to 7

Figure 9-21: Basic Operation Timing in External Event Count Mode (1/2)

(b) D1 = D2; no TPnCCR0, TPnCCR1 rewrite; TOPn0, TOPn1 output



Remarks: 1. D1: Setting value of TPnCCR0 register (0000H to FFFFH)
D2: Setting value of TPnCCR1 register (0000H to FFFFH)

2. Event count = (Dn + 1)

3. n = 0 to 7

9.5.4 External trigger pulse output mode (TPnMD2 to TPnMD0 = 010B)

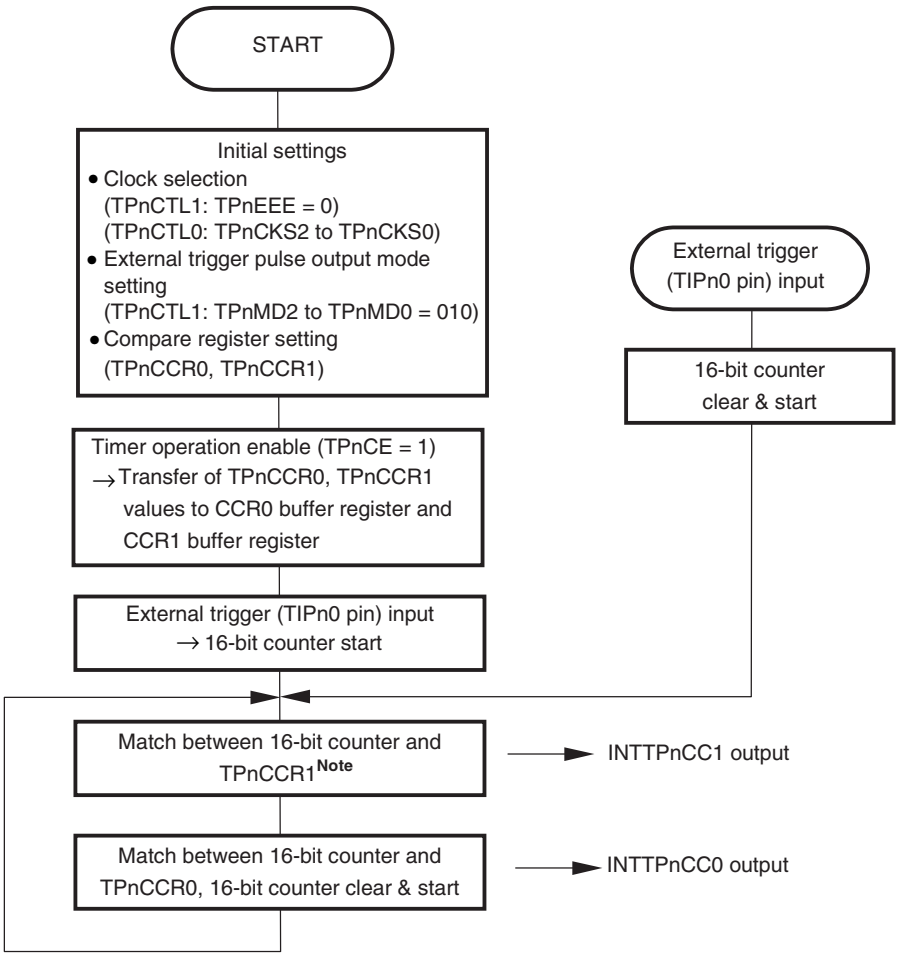
In the external trigger pulse output mode, setting TPnCE = 1 causes external trigger input (TTRGPn pin input) wait with the 16-bit counter stopped at FFFFH. The count-up operation starts upon detection of the external trigger input (TTRGPn pin input) edge.
Regarding TOPn1 output control, the reload register (TPnCCR1) is used as the duty setting register and the compare register (TPnCCR0) is used as the cycle setting register.
The TPnCCR0 register and the TPnCCR1 register can be rewritten when TPnCE = 1.
In order for the setting value when the TPnCCR0 register and the TPnCCR1 register are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be reloaded to the CCRm buffer register), it is necessary to rewrite TPnCCR0 and then write to the TPnCCR1 register before the 16-bit counter value and the TPnCCR0 register value match. Thereafter, the values of the TPnCCR0 and the TPnCCR1 register are reloaded upon a TPnCCR0 register match.
Whether to enable or disable the next reload timing is controlled by writing to the TPnCCR1 register. Thus even when wishing only to rewrite the value of the TPnCCR0 register, also write the same value to the TPnCCR1 register.
Reload is disabled even when only the TPnCCR0 register is rewritten. To stop timer P, set TPnCE = 0. If the external trigger (TTRGPn pin input) edge is detected several times in the external trigger pulse mode, the 16-bit counter is cleared at the edge detection timing and count-up starts.
To realize the same function (software trigger pulse mode) as external trigger pulse mode using a software trigger instead of external trigger input (TTRGPn pin input), set the TPnEST bit of the TPnCTL1 register to 1 so that the software trigger is output. The external trigger pulse waveform is output from TOPn1. The TOPn0 pin performs toggle output upon a match between the TPnCCR0 register and the 16-bit counter.
Since the TPnCCR0 register and the TPnCCR1 register have their function fixed to that of a compare register in the external trigger pulse mode, they cannot be used for capture operation in this mode.

Caution: In the external trigger pulse output mode, select the internal clock (TPnEEE bit of TPnCTL1 register = 0) for the count clock.

Remarks: 1. For the reload operation when TPnCCR0 and TPnCCR1 are rewritten during timer operation, refer to **9.5.6 PWM mode (TPnMD2 to TPnMD0 = 100B)**.

2. n = 0 to 7
m = 0, 1

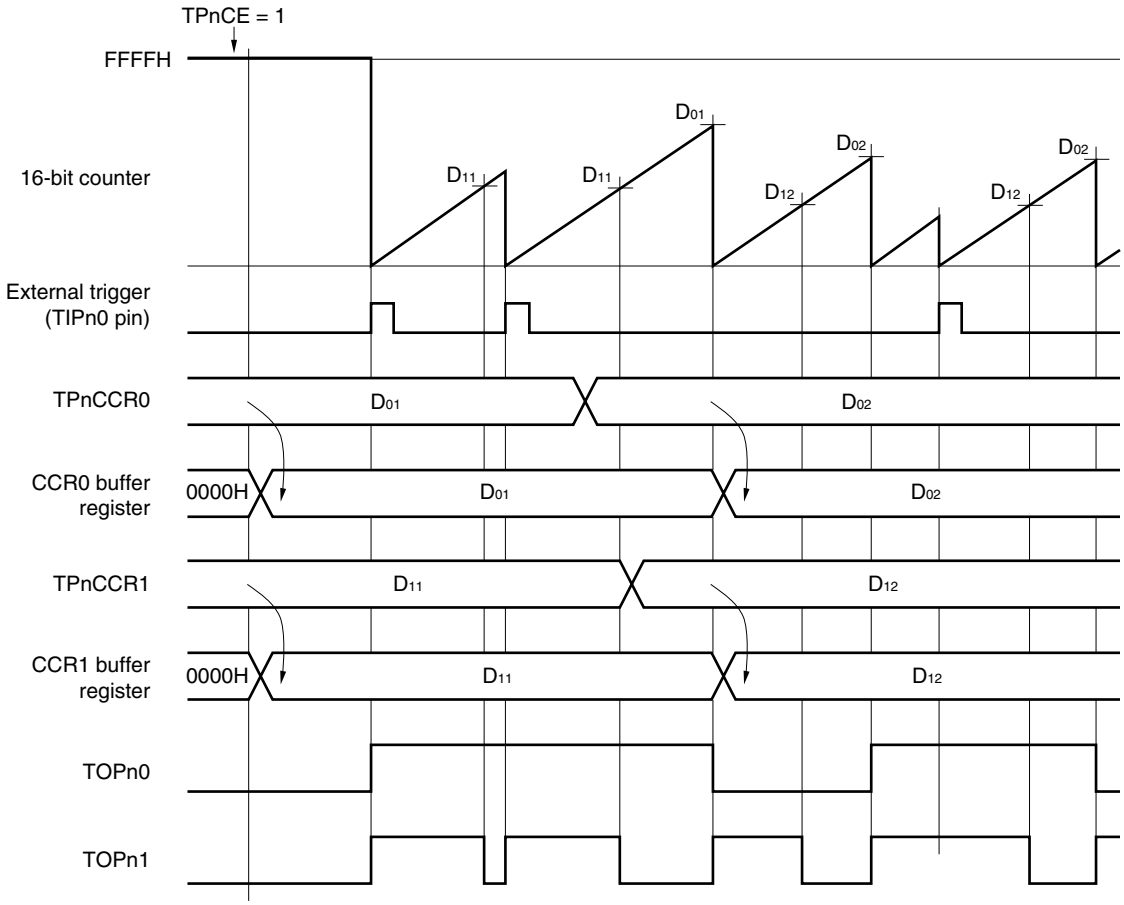
Figure 9-22: Flowchart of Basic Operation in External Trigger Pulse Output Mode



Note: The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.

Remark: n = 0 to 7

Figure 9-23: Basic Operation Timing in External Trigger Pulse Output Mode



Remarks: 1. D01, D02: Setting value of TPnCCR0 register (0000H to FFFFH)
D11, D12: Setting value of TPnCCR1 register (0000H to FFFFH)

2. TOPn1 output duty = (Setting value of TPnCCR1 register) / (Setting value of TP0CCR0 register)
TOPn1 output cycle = (Setting value of TPnCCR0 register) × (Count clock cycle)

3. n = 0 to 7

9.5.5 One-shot pulse mode (TPnMD2 to TPnMD0 = 011B)

In the one-shot pulse mode, setting TPnCE = 1 causes waiting on TPnEST bit setting (1) or TTRGPn pin edge detection trigger[Note 1] with the 16-bit counter held at FFFFH. The 16-bit counter starts counting up upon trigger input, and upon a match between the value of the 16-bit counter and the value of the CCR1 buffer register transferred from the TPnCR1 register, TOPn1 becomes high level; Upon a match between the value of the 16-bit counter and the value of the CCR0 register transferred from the TPnCCR0 register, TOPn1 becomes low level and the 16-bit counter is cleared to 0000H and stops. Any trigger input past the first one during 16-bit counter operation is ignored. Be sure to input the second and subsequent triggers when the 16-bit counter has stopped at 0000H. In the one-shot pulse mode, the TPnCCR0 and TPnCCR1 registers can be rewritten when TPnCE = 1. The setting values rewritten to the TPnCCR0 and TPnCCR1 registers become valid following execution of a write instruction from the CPU, at which time they are transferred to the CCR0 buffer register and the CCR0 buffer register through anytime write, and become the values for comparison with the 16-bit counter value. The one-shot pulse waveform is output from the TOPn1 pin. The TOPn0 pin performs toggle output upon a match between the 16-bit counter and the TPnCCR0 register[Note 2].
Since the TPnCCR0 and TPnCCR1 registers have their function fixed to that of a compare register in the one-shot pulse mode, they cannot be used for capture operation in this mode.

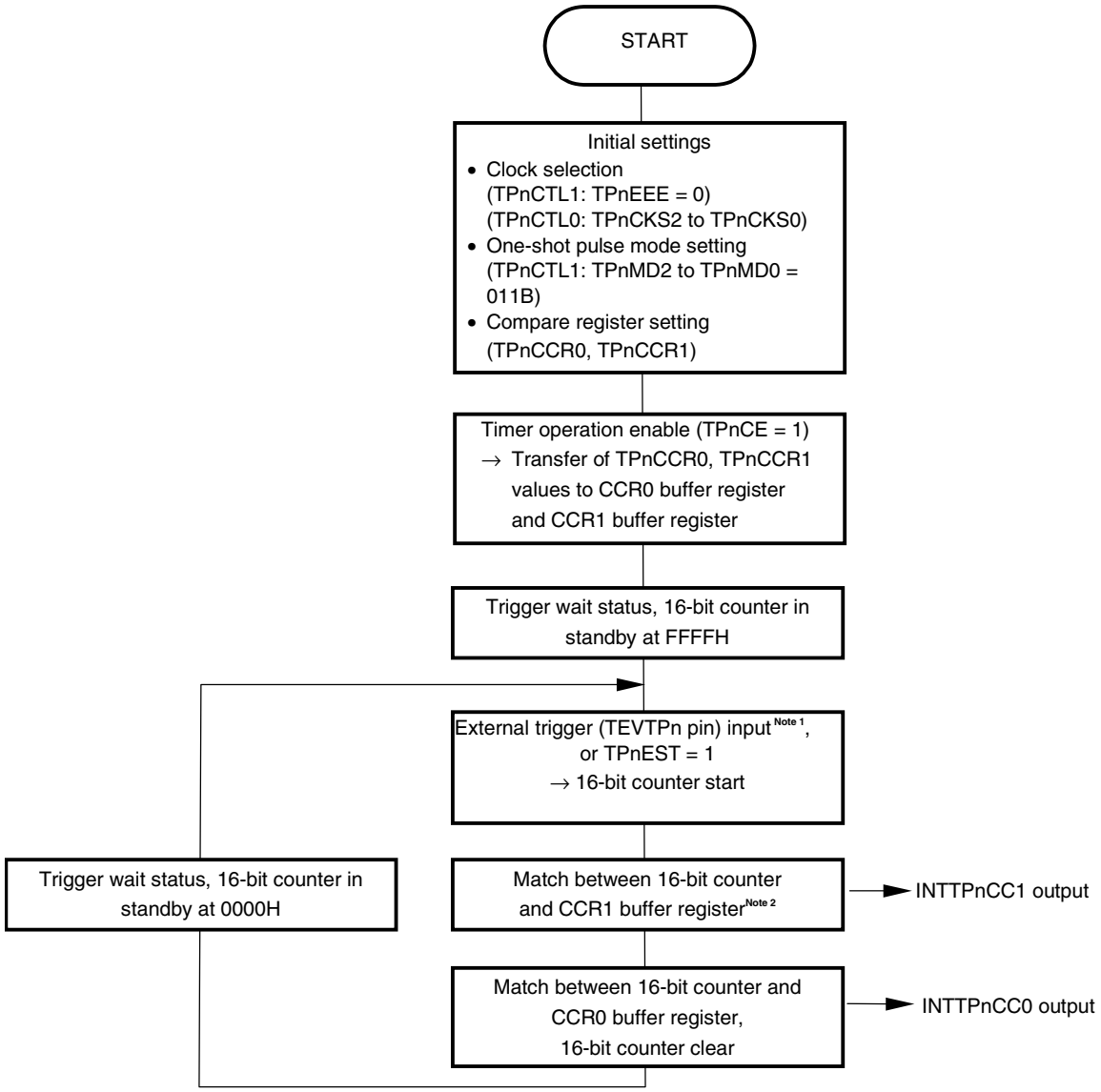
Notes: 1. External trigger input pin (TTRGPn) is not available for TMP8 (n = 8).

2. Output pin (TOPn0) is not available for TMP8 (n = 8).

Caution: In the one-shot pulse mode, select the internal clock (TPnEEE bit of TPnCTL1 register = 0) for the count clock.

Remark: n = 0 to 8

Figure 9-24: Flowchart of Basic Operation in One-Shot Pulse Mode



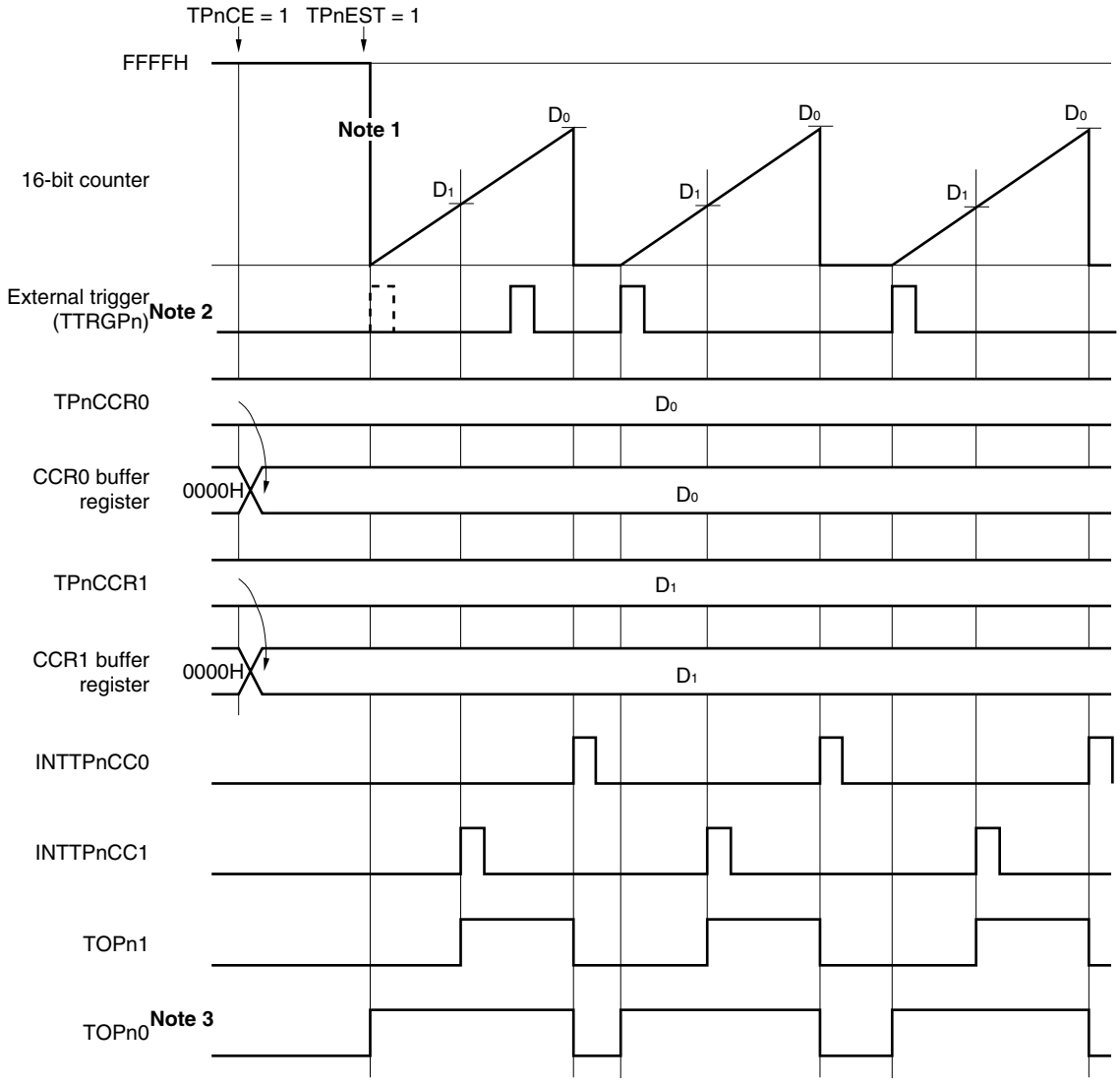
Notes: 1. External trigger input (TTRGPn) is not available for TMP8 (n = 8).

2. The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.

Caution: The 16-bit counter is not cleared and trigger input is ignored even if trigger input is performed during the count-up operation of the 16-bit counter.

Remark: n = 0 to 8

Figure 9-25: Timing of Basic Operation in One-Shot Pulse Mode



Notes: 1. The 16-bit counter starts counting up when either TPnEST = 1 is set or TEVTPn is input.

2. External trigger input pin (TTRGPn) is not available for TMP8 (n = 8).

3. Output pin (TOPn0) is not available for TMP8 (n = 8).

Remarks: 1. D0: Setting value of TPnCCR0 register (0000H to FFFFH)
D1: Setting value of TPnCCR1 register (0000H to FFFFH)

2. Delay time of one-shot pulse output (TOPn1) when external pin edge detection trigger is used:
(TPnCCR1 value + 1) (Selected count clock) + $2/(f_{XX})$ + (TTRGPn input filter delay)

3. n = 0 to 8

9.5.6 PWM mode (TPnMD2 to TPnMD0 = 100B)

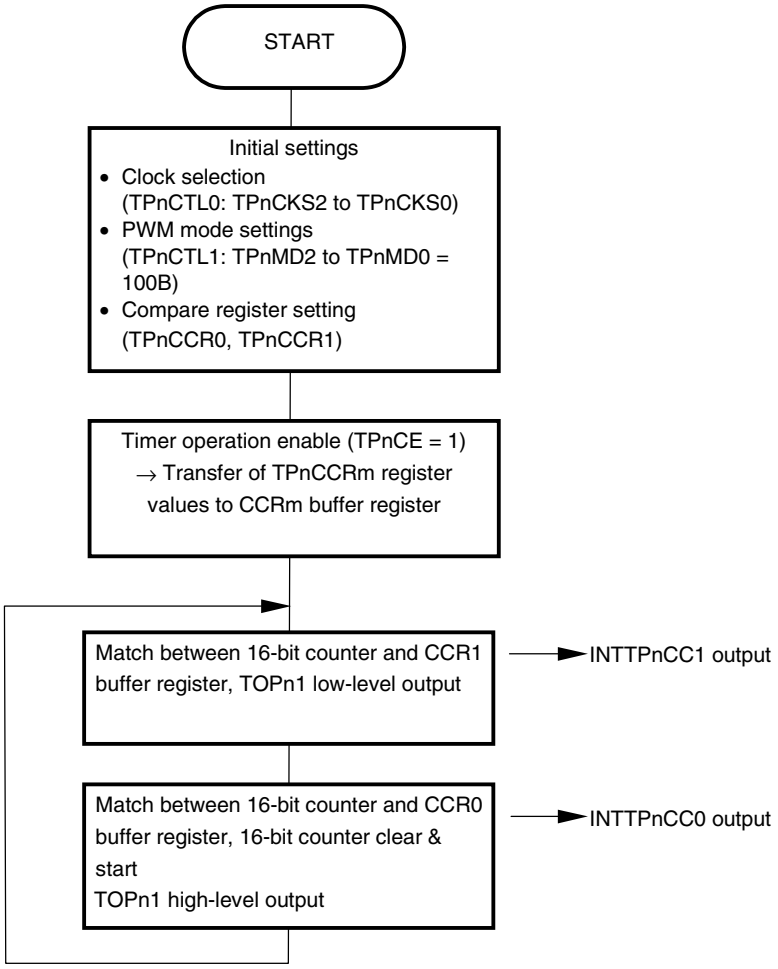
In the PWM mode, TMPn capture/compare register 1 (TPnCCR1) is used as the duty setting register and TMPn capture/compare register 0 (TPnCCR0) is used as the cycle setting register.
Variable duty PWM is output by setting these two registers and operating the timer.
The TPnCCR0 register and the TPnCCR1 register can be rewritten when TPnCE = 1.
In order for the setting value when the TPnCCR0 register and the TPnCCR1 register are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be reloaded to CCR0 buffer register or CCR1 buffer register), it is necessary to rewrite TPnCCR0 and then write to the TPnCCR1 register before the 16-bit counter value and the TPnCCR0 register value match. Thereafter, the values of the TPnCCR0 register and the TPnCCR1 register are reloaded upon a TPnCCR0 register match.
Whether to enable or disable the next reload timing is controlled by writing to the TPnCCR1 register. Thus even when wishing only to rewrite the value of the TPnCCR0 register, also write the same value to the TPnCCR1 register.
Reload is disabled even when only the TPnCCR0 register is rewritten. To stop timer P, set TPnCE = 0.
PWM waveform output is performed from the TOPn1 pin. The TOPn0 pin**Note** performs toggle output upon a match between the 16-bit counter and the TPnCCR0 register.
Since the TPnCCR0 and TPnCCR1 registers have their function fixed that of a compare register in the PWM mode, they cannot be used for capture operation in this mode.

Note: TOPn0 output pin is not available for TMP8 (n = 8).

Remark: n = 0 to 8

Figure 9-26: Flowchart of Basic Operation in PWM Mode (1/2)

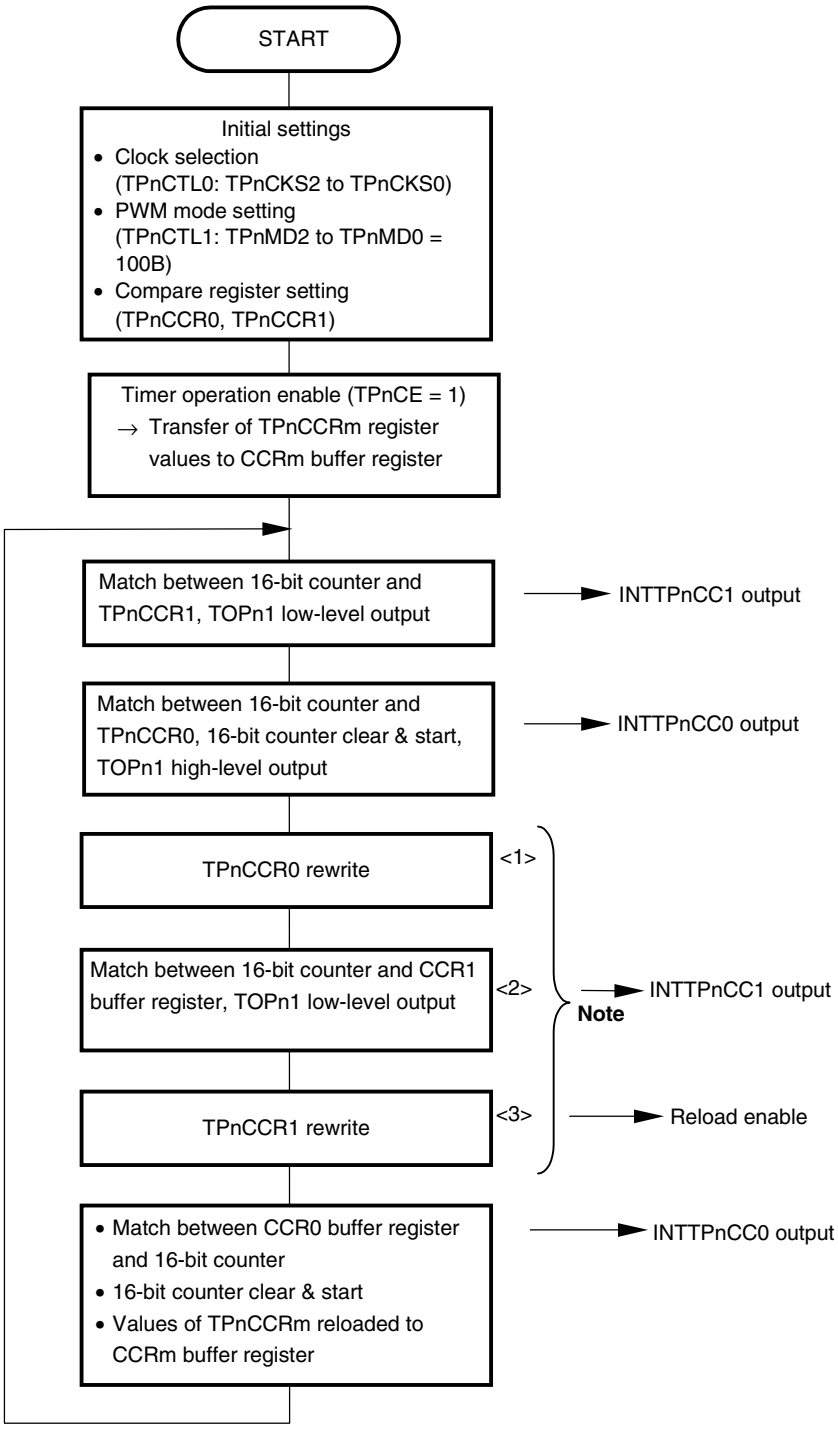
(a) Values of TPnCCR0, TPnCCR1 registers not rewritten during timer operation



Remark: n = 0 to 8
m = 0, 1

Figure 9-26: Flowchart of Basic Operation in PWM Mode (2/2)

(b) Values of TPnCCR0, TPnCCR1 registers rewritten during timer operation

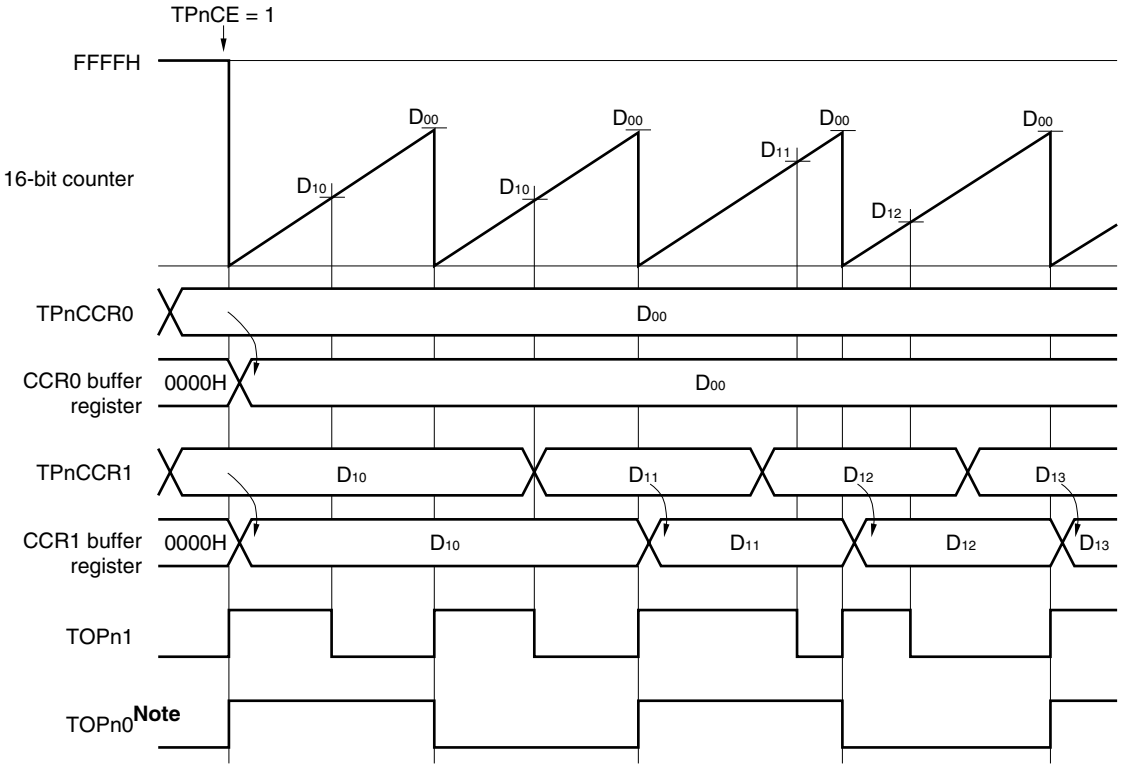


Note: The timing of <2> in the above flowchart may differ depending on the rewrite timing of steps <1> and <3> and the value of TPnCCR1, but make sure that step <3> comes after step <1>.

Remark: n = 0 to 8
m = 0, 1

Figure 9-27: Basic Operation Timing in PWM Mode (1/2)

(a) TPnCCR1 value rewritten



Note: TOPn0 output pin is not available for TMP8 (n = 8).

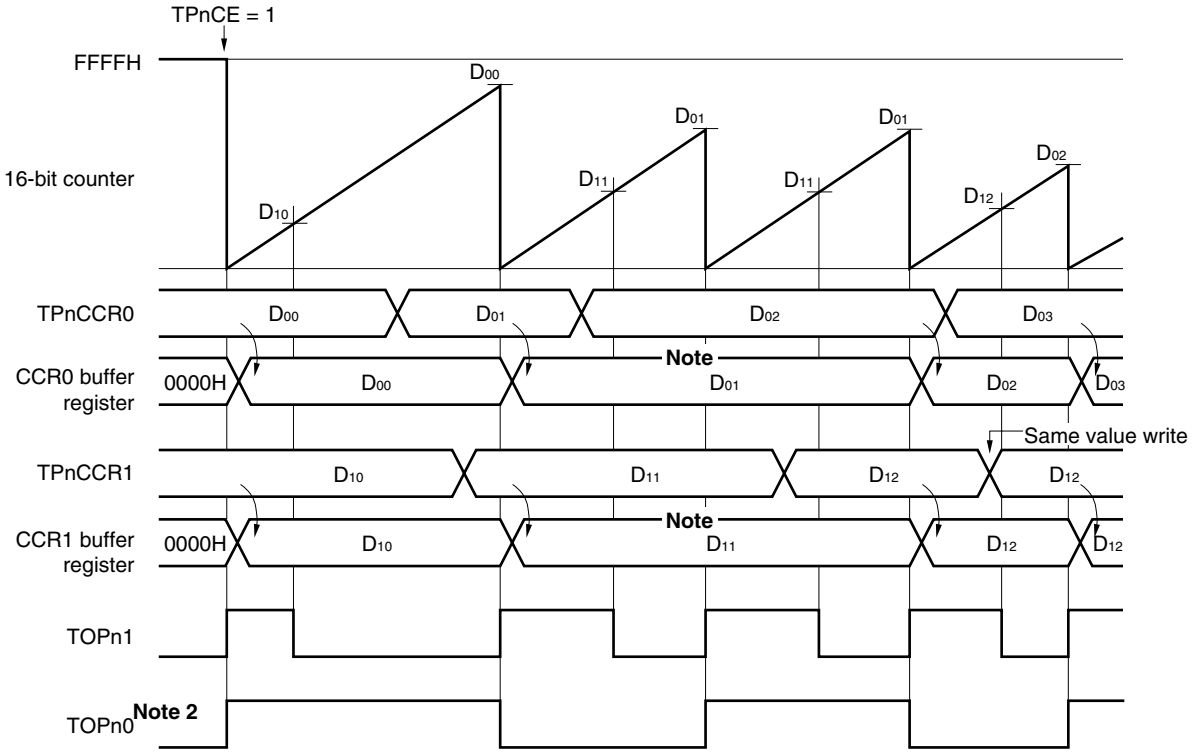
Remarks: 1. D_{00}: Setting value of TPnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}, D_{13}: Setting values of TPnCCR1 register (0000H to FFFFH)

2. TOPn1 output duty factor = (Setting value of TPnCCR1 register) / (Setting value of TP0CCR0 register + 1)
TOPn1 output cycle = (Setting value of TPnCCR0 register + 1) × (Count clock cycle)
TOPn0 output toggle width = (Setting value of TPnCCR0 register + 1) × (Count clock cycle)

3. n = 0 to 8

Figure 9-27: Basic Operation Timing in PWM Mode (2/2)

(b) TPnCCR0, TPnCCR1 values rewritten



Notes: 1. Reload is not performed because the TPnCCR1 register was not rewritten.

2. TOPn0 output pin is not available for TMP8 (n = 8).

Remarks: 1. D_{00}, D_{01}, D_{02}, D_{03}: Setting values of TPnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}, D_{13}: Setting values of TPnCCR1 register (0000H to FFFFH)

2. TOPn1 output duty factor = (Setting value of TPnCCR1 register) / (Setting value of TP0CCR0 register + 1)
TOPn1 output cycle = (Setting value of TPnCCR0 register + 1) × (Count clock cycle)
TOPn0 output toggle width = (Setting value of TPnCCR0 register + 1) × (Count clock cycle)

3. n = 0 to 8

9.5.7 Free-running mode (TPnMD2 to TPnMD0 = 101B)

In the free-running mode, both the interval function and the compare function can be realized by operating the 16-bit counter as a free-running counter and selecting capture/compare operation with the TPnCCS1 and TPnCCS0 bits.
The settings of the TPnCCS1 and TPnCCS0 bits of the TPnOPT0 register are valid only in the free-running mode.

TPnCCS1	Operation
0	Use TPnCCR1 register as compare register
1	Use TPnCCR1 register as capture register

TPnCCS0	Operation
0	Use TPnCCR0 register as compare register
1	Use TPnCCR0 register as capture register

(a) Using TPnCCR1 register as compare register

An interrupt is output upon a match between the 16-bit counter and the CCR1 buffer register in the free-running mode (interval function).
Rewrite during compare timer operation is enabled and performed with anytime write. (Once the compare value has been written, synchronization with the internal clock is done and this value is used as the 16-bit counter comparison value.)
When timer output (TOPn1) has been enabled, TOPn1 performs toggle output upon a match between the 16-bit counter and the CCR1 buffer register.

(b) Using TPnCCR1 register as capture register

The value of the 16-bit counter is saved to the TPnCCR1 register upon TIPn1 pin[Note 1] edge detection.

(c) Using TPnCCR0 register as compare register

An interrupt is output upon a match between the 16-bit counter and the CCR0 buffer register in the free-running mode (interval function).
Rewrite during compare timer operation is enabled and performed with anytime rewrite.
When timer output (TOPn0) has been enabled, TOPn0[Note 2] performs toggle output upon a match between the 16-bit counter and the CCR0 buffer register.

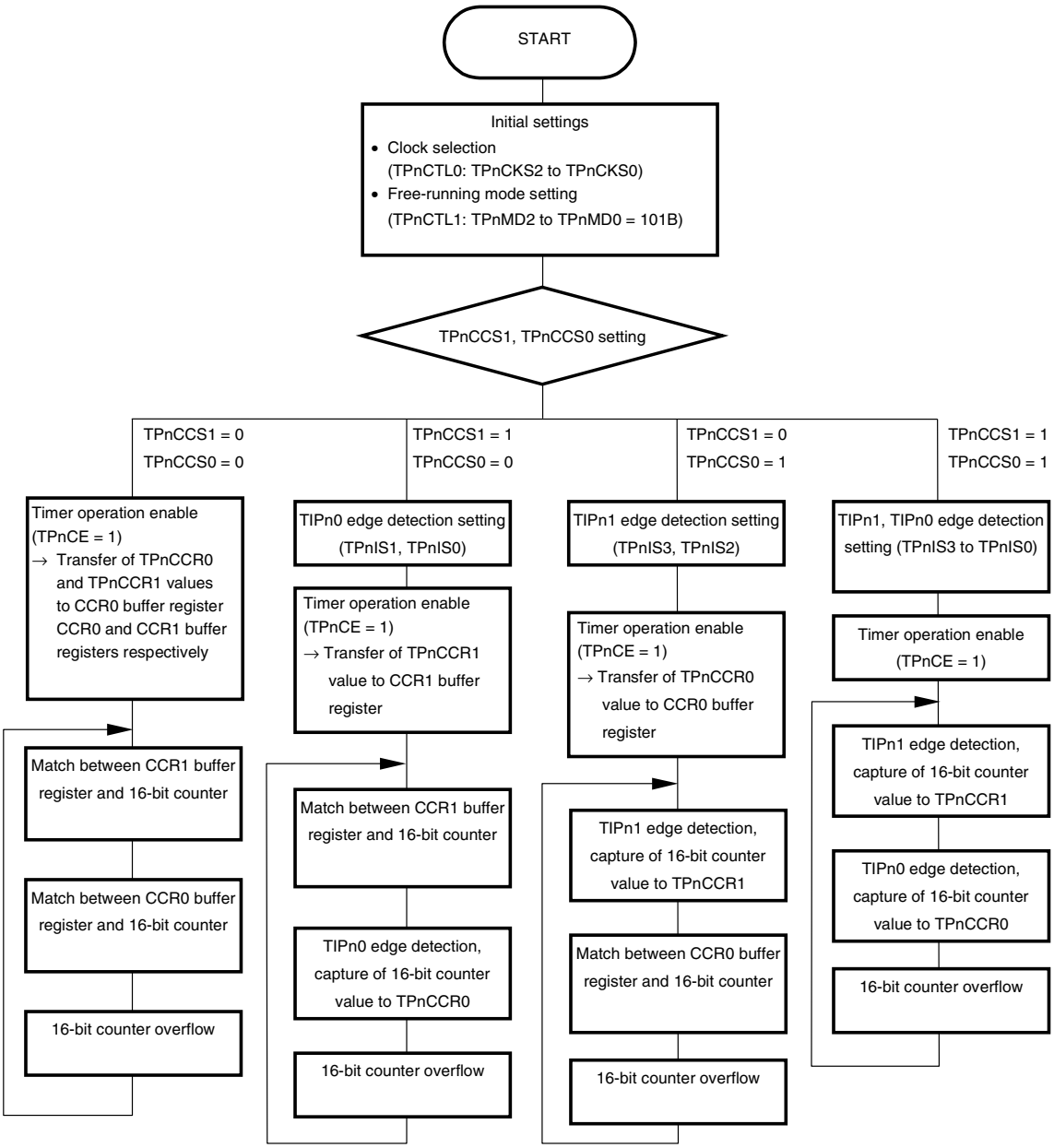
(d) Using TPnCCR0 register as capture register

The value of the 16-bit counter is saved to the TPnCCR0 register upon TIPn0 pin[Note 1] edge detection.

Notes: 1. Since TMP8 has no external input pin, the capture function can only be used internally for capturing the interrupt signal INTTT0CC0 of TMT0, or INTCM10 of TMENC1, into the TP8CCR0 register, or the interrupt signal INTTT0CC1 of TMT0, or INTCM11 of TMENC1 into the TP8CCR1 register respectively, which is specified by the TPIC22 bit of TPIC2 register (refer to **9.4 (9) TMP input control register 2 (TPIC2)**).

2. TOPn0 output pin is not available for TMP8 (n = 8).

Figure 9-28: Flowchart of Basic Operation in Free-Running Mode

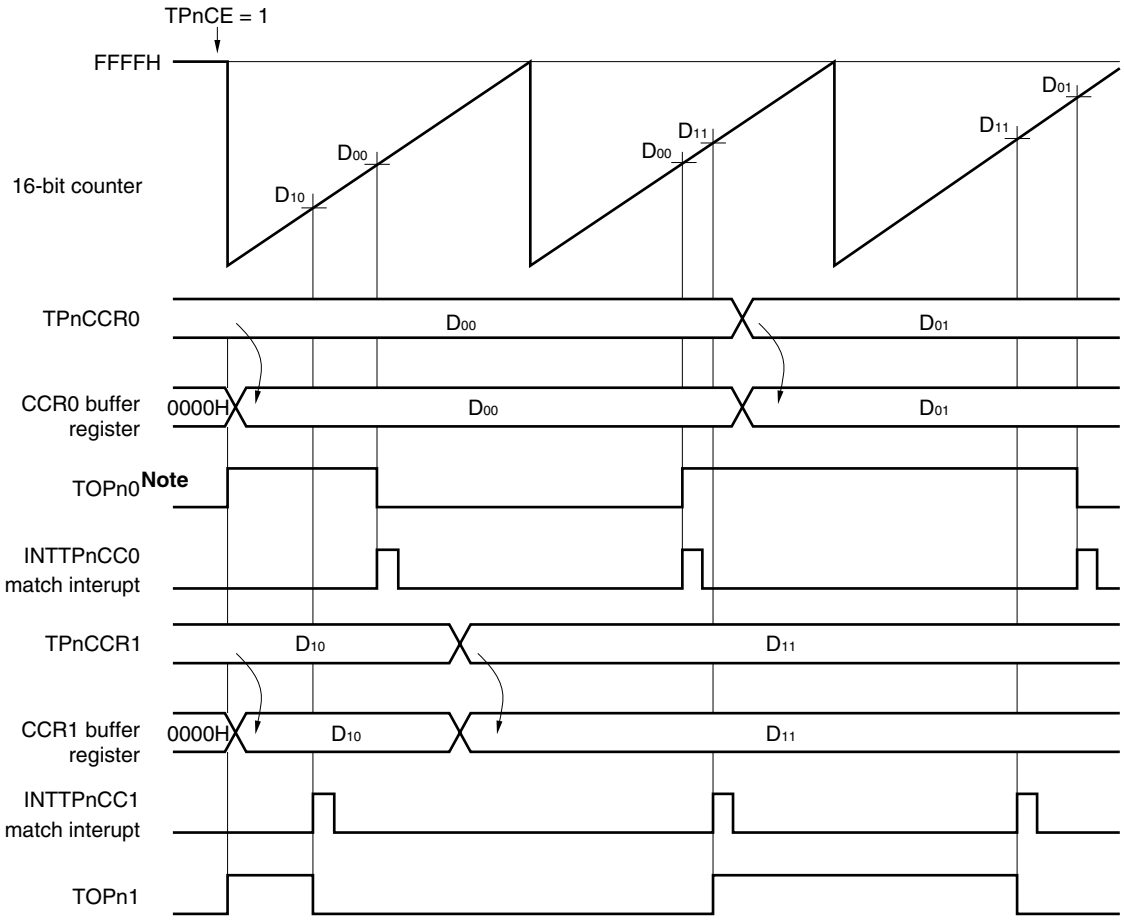


Remark: n = 0 to 8

(1) TPnCCS1 = 0, TPnCCS0 = 0 settings (interval function description)

When TPnCE = 1 is set, the 16-bit counter counts from 0000H to FFFFH and the free-running count-up operation continues until TPnCE = 0 is set. In this mode, when a value is written to the TPnCCR0 and TPnCCR1 registers, they are transferred to the CCR0 buffer register and the CCR1 buffer register (anytime write). In this mode, no one-shot pulse is output even when an one-shot pulse trigger is input. Moreover, when TPnOEm = 1 is set, TOPnm performs toggle output upon a match between the 16-bit counter and the CCRm buffer register.

Figure 9-29: Basic Operation Timing in Free-Running Mode (TPnCCS1 = 0, TPnCCS0 = 0)



Note: TOPn0 output pin is not available for TMP8 (n = 8).

Remarks: 1. D_{00}, D_{01}: Setting values of TPnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}: Setting values of TPnCCR1 register (0000H to FFFFH)

2. TOPnm output rises to the high level when counting is started.

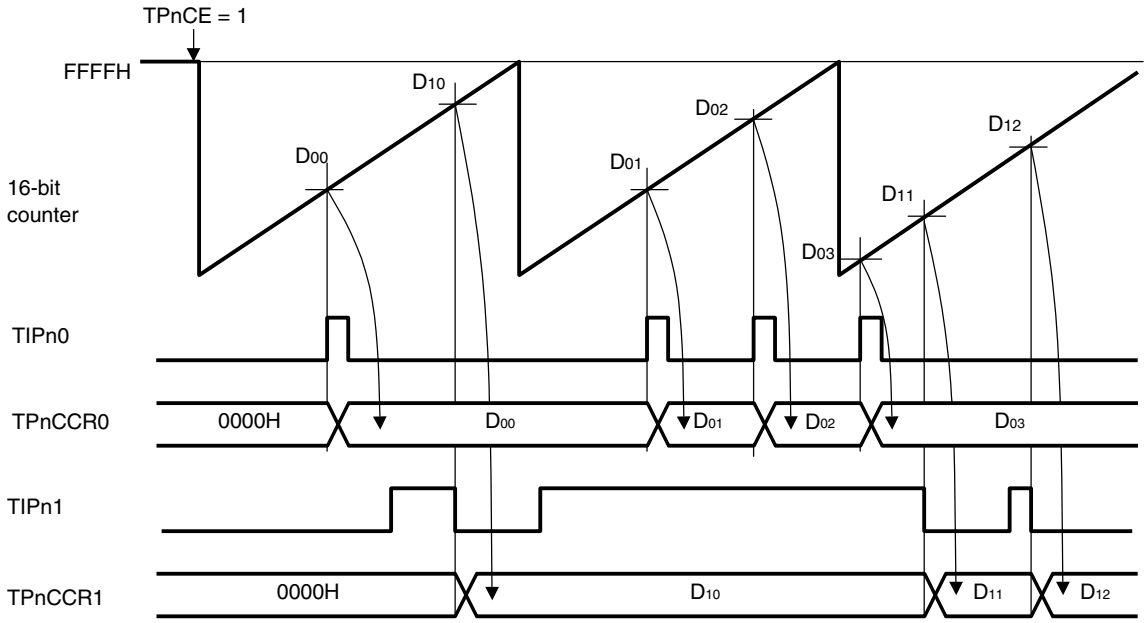
3. n = 0 to 8
m = 0, 1

(2) TPnCCS1 = 1, TPnCCS0 = 1 settings (capture function description)

When TPnCE = 1, the 16-bit counter counts from 0000H to FFFFH and free-running count-up operation continues until TPnCE = 0 is set. During this time, values are captured by capture trigger operation and are written to the TPnCCR0 and TPnCCR1 registers.

Regarding capture in the vicinity of overflow (FFFFH), judgment is made using the overflow flag (TPnOVF). However, if overflow occurs twice (2 or more free-running cycles), the capture trigger interval cannot be judged with the TPnOVF flag. In this case, the system should be revised.

Figure 9-30: Basic Operation Timing in Free-Running Mode (TPnCCS1 = 1, TPnCCS0 = 1)



Remarks: 1. D_{00}, D_{01}: Values captured to TPnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}: Values captured to TPnCCR1 register (0000H to FFFFH)

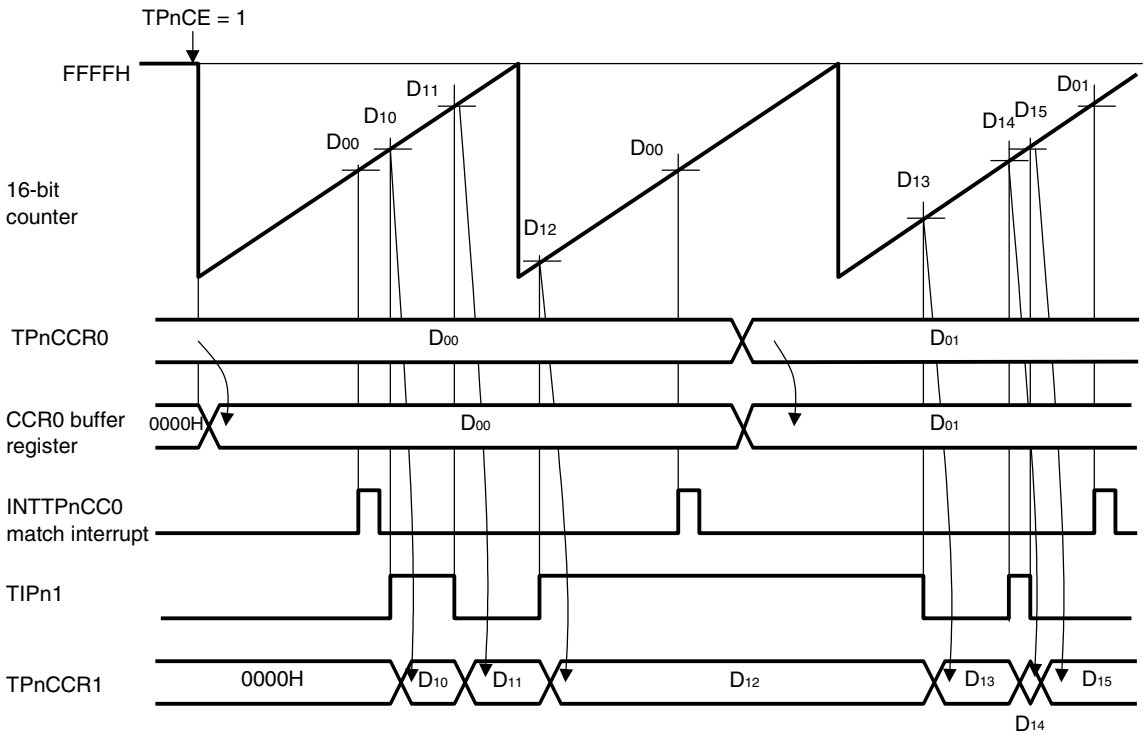
2. TIPn0: Set to rising edge detection (TPnIS1, TPnIS0 = 01B)
TIPn1: Set to falling edge detection (TPnIS3, TPnIS2 = 10B)

3. n = 0 to 7

(3) TPnCCS1 = 1, TPnCCS0 = 0 settings

When TPnCE = 1 is set, the counter counts from 0000H to FFFFH and free-running count-up operation continues until TPnCE = 0 is set. The TPnCCR0 register is used as a compare register. An interrupt signal is output upon a match between the value of the 16-bit counter and the setting value transferred to the CCR0 buffer register from the TPnCCR0 register as an interval function. Even if TPnOE1 = 1 is set to realize the capture function, the TPnCCR1 register cannot control TOPn1.

Figure 9-31: Basic Operation Timing in Free-Running Mode (TPnCCS1 = 1, TPnCCS0 = 0)



Remarks: 1. D00, D01: Setting values of TPnCCR0 register (0000H to FFFFH)
D10, D11, D12, D13, D14, D15: Values captured to TPnCCR1 register (0000H to FFFFH)

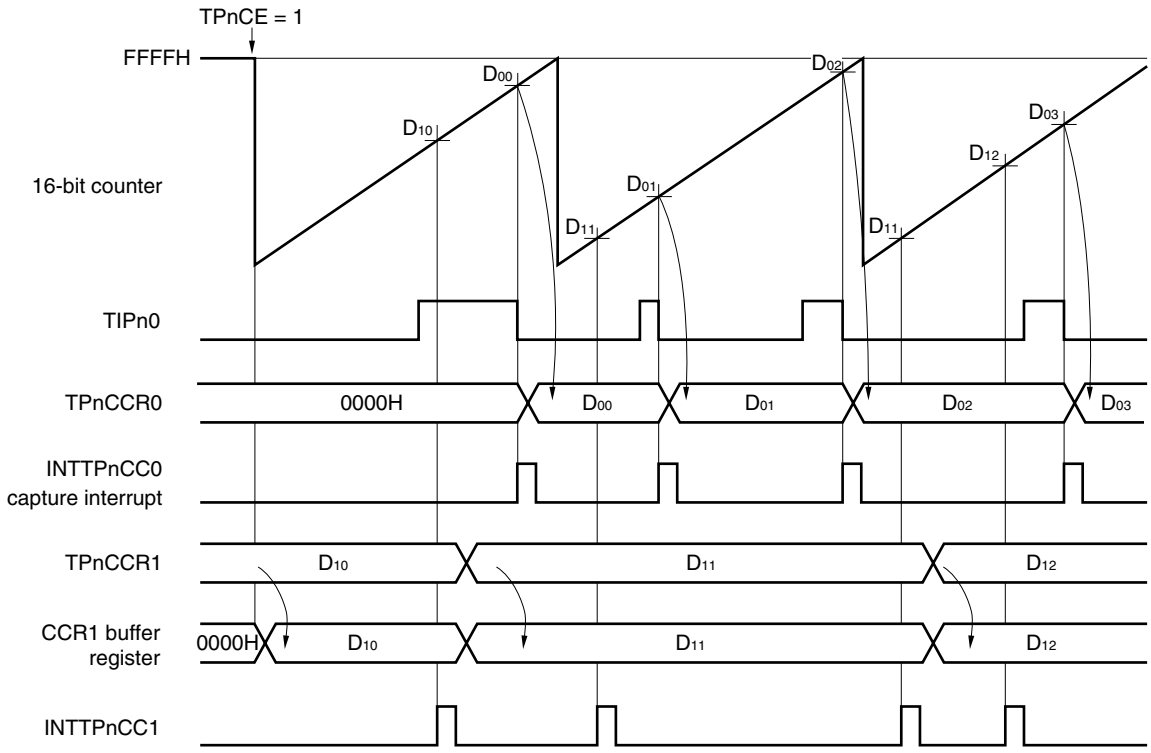
2. TIPn1: Set to detection of both rising and falling edges (TPnIS3, TPnIS2 = 11B)

3. n = 0 to 7

(4) TPnCCS1 = 0, TPnCCS0 = 1 settings

When TPnCE is set to 1, the 16-bit counter counts from 0000H to FFFFH and free-running count-up operation continues until TPnCE = 0 is set. The TPnCCR1 register is used as a compare register. An interrupt signal is output upon a match between the value of the 16-bit counter and the setting value of the TPnCCR1 register as an interval function. When TPnOE1 = 1 is set, TOPn1 performs toggle output upon mach between the value of the 16-bit counter and the setting value of the TPnCCR1 register.

Figure 9-32: Basic Operation Timing in Free-Running Mode (TPnCCS1 = 0, TPnCCS0 = 1)



Remarks: 1. D_{00}, D_{01}, D_{02}, D_{03}: Values captured to TPnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}: Setting value of TPnCCR1 register (0000H to FFFFH)

2. TIPn0: Set to falling edge detection (TPnIS1, TPnIS0 = 10B)

3. n = 0 to 7

(5) Overflow flag

When the counter overflows from FFFFH to 0000H in the free-running mode, the overflow flag (TPnOVF) is set to 1 and an overflow interrupt (INTTPnOV) is output.
Be sure to confirm that the overflow flag (TPnOVF) is set to “1” when the overflow interrupt (INTTPnOV) has occurred.
The overflow flag is cleared by writing 0 from the CPU.

9.5.8 Pulse width measurement mode (TPnMD2 to TPnMD0 = 110B)

In the pulse width measurement mode, free-running count is performed. The value of the 16-bit counter is saved to capture register 0 (TPnCCR0), or capture register 1 (TPnCCR1) respectively, and the 16-bit counter is cleared upon edge detection of the TIPn0 pin, or TIPn1 respectively. The external input pulse width can be measured as a result.
However, when measuring a large pulse width that exceeds 16-bit counter overflow, perform judgment with the overflow flag. Since measurement of pulses for which overflow occurs twice or more is not possible, adjust the operating frequency of the 16-bit counter.

Depending on the selected capture input sources and specified edge detection three different measurement methods can be applied.

<1> Pulse period measurement
<2> Alternating pulse width and pulse space measurement:
This requires a fast interrupt handling, in order to measure pulse width and pulse space correctly.
<3> Simultaneous pulse width and pulse space measurement:
Both capture inputs are required to measure pulse width and pulse space simultaneously.

The measurements methods are explained in the following sub-chapters.

Cautions: 1. In the pulse width measurement mode, select the internal clock (TPnEEE of TPnCTL1 register = 0).

2. Pulse width measurement cannot be performed by timer P8 (TMP8).

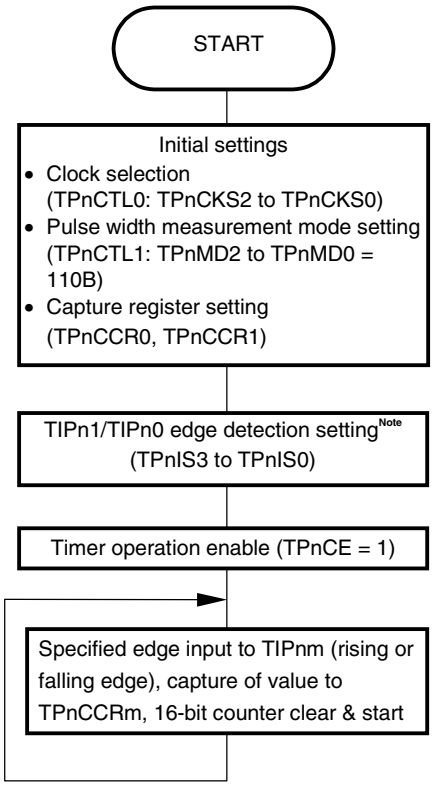
Remark: n = 0 to 7

(1) Pulse period measurement

The pulse period of a signal can be measured in the pulse width measurement mode, when the edge detection of one of the inputs TIPn0 and TIPn1 is set either to “rising edge” or “falling edge”. The detection of the other input should be set to “no edge detection”.

By detection of the specified edge the resulting value is captured in the corresponding capture register (TPnCCR0 or TPnCCR1), and the timer is cleared and restarts counting.

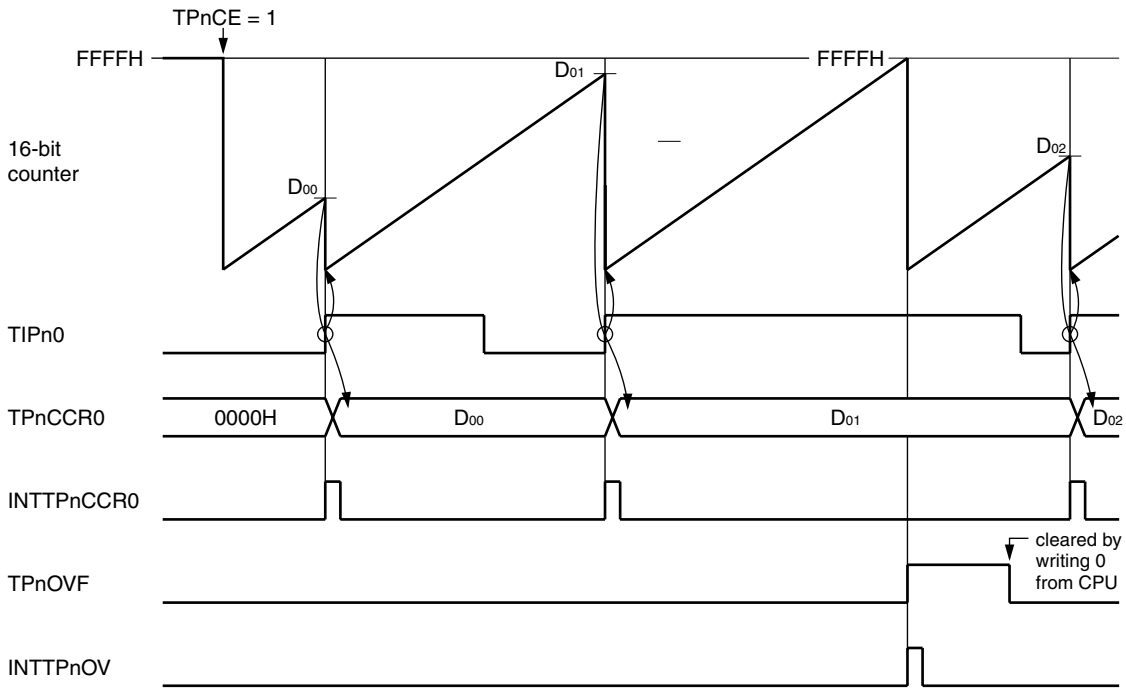
Figure 9-33: Flowchart of Pulse Period Measurement



Note: External pulse input is possible for both TIPn0 and TIPn1, but only one should be selected for the pulse period measurement.
Specify either “rising edge” or “falling edge” for edge detection. Specify the edge of the external input pulse that is not used as “no edge detection”.

Remark: n = 0 to 7
m = 0, 1

Figure 9-34: Basic Operation Timing of Pulse Period Measurement



Remarks: **1.** D_{00}, D_{01}, D_{02}: Values captured to TPnCCR0 register (0000H to FFFFH)

2. TIPn0: Set to detection of rising edge (TPnIS1, TPnIS0 = 01B)

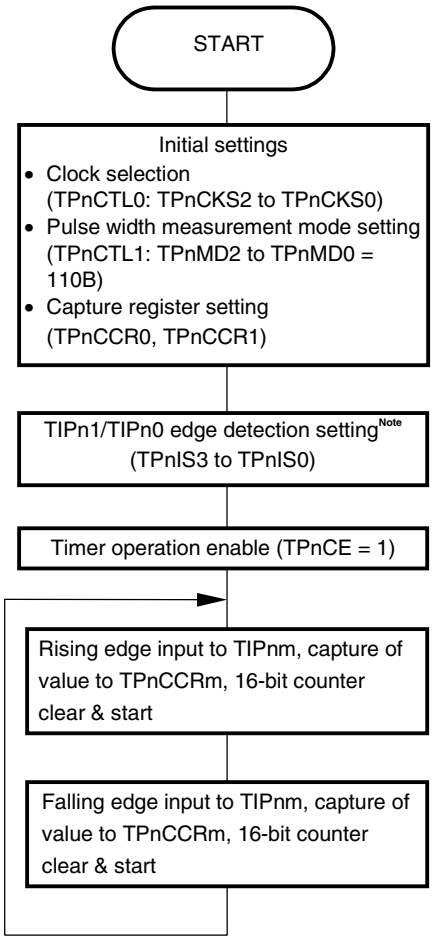
3. TIPn1: Set to no edge detection (TPnIS3, TPnIS2 = 00B)

4. n = 0 to 7

(2) Alternating pulse width and pulse space measurement

The pulse period of a signal can be measured in the pulse width measurement mode alternating in one capture register, when the edge detection of one of the inputs TIPn0 and TIPn1 is set to “both rising and falling edges”. The detection of the other input should be set to “no edge detection”. By detection of a falling or rising edge the resulting value is captured in the corresponding capture register (TPnCCR0 or TPnCCR1), and the timer is cleared and restarts counting.

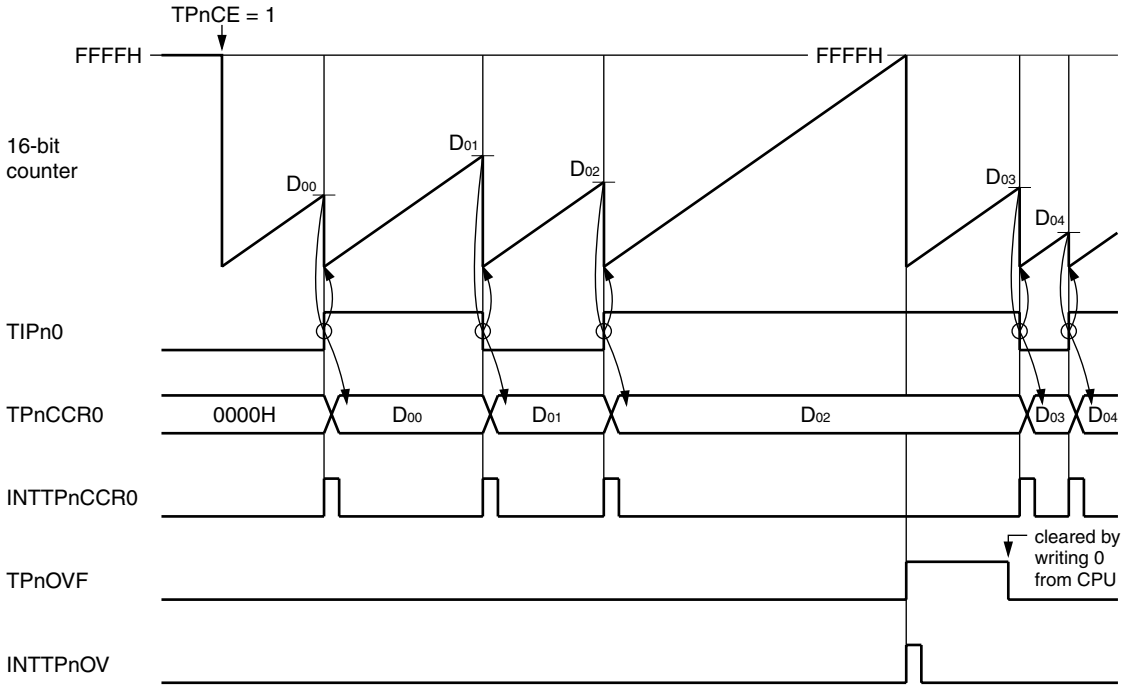
Figure 9-35: Flowchart of Alternating Pulse Width and Pulse Space Measurement



Note: External pulse input is possible for both TIPn0 and TIPn1, but only one should be selected for the alternating pulse width and pulse space measurement.
Specify “both rising and the falling edges” for edge detection. Specify the edge of the external input pulse that is not used as “no edge detection”.

Remark: n = 0 to 7
m = 0, 1

Figure 9-36: Basic Operation Timing of Alternating Pulse Width and Pulse Space Measurement



Remarks: **1.** D_{00}, D_{01}, D_{02}, D_{03}, D_{04}: Values captured to TPnCCR0 register (0000H to FFFFH)

2. TIPn0: Set to detection of both rising and falling edges (TPnIS1, TPnIS0 = 11B)

3. TIPn1: Set to no edge detection (TPnIS3, TPnIS2 = 00B)

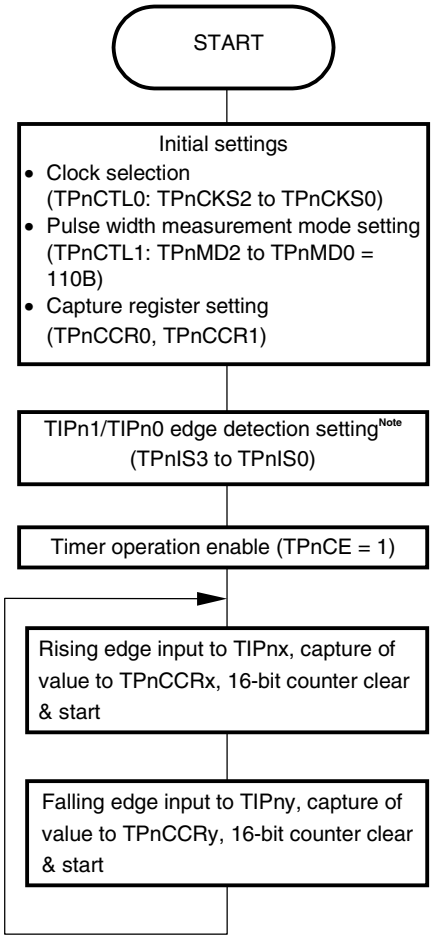
4. n = 0 to 7

(3) Simultaneous pulse width and pulse space measurement

Pulse width and pulse space can be measure simultaneously in the pulse width measurement mode, when the signal is input to both inputs TIPn0 and TIPn1, where both inputs detect opposite edges. Alternatively the signal can be input to TIPn0 only, when the capture source input selection for capture register 1 is used (ref. to **9.4 (7) TMP input control register 0 (TPIC0)** and **9.4 (8) TMP input control register 1 (TPIC1)**).

By detection of the specified edge the resulting values of pulse width or pulse space are captured in the corresponding capture registers (TPnCCR0, TPnCCR1), and the timer is cleared and restarts counting.

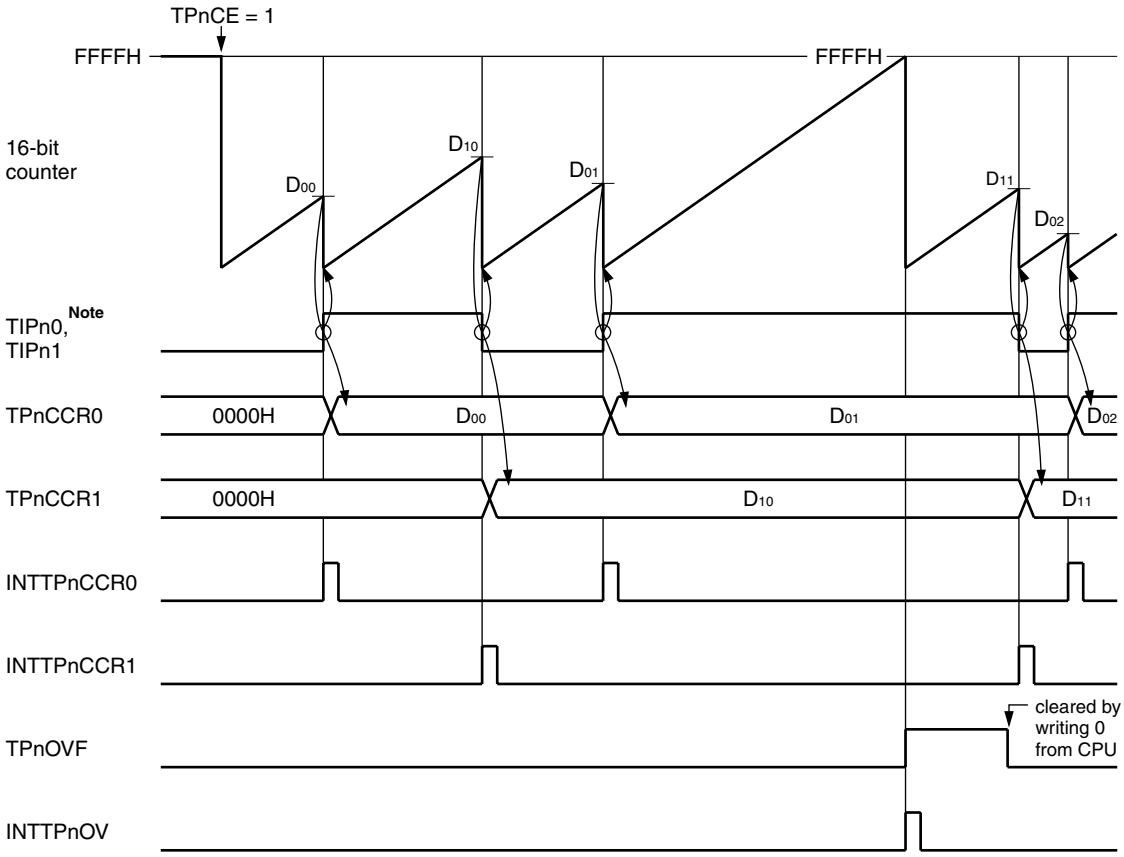
Figure 9-37: Flowchart of Simultaneous Pulse Width and Pulse Space Measurement



Note: External pulse input must be input to both TIPn0 and TIPn1, or to TIPn0 only, if the internal connection between both inputs is selected.
Specify “rising edge” for edge detection of first input, and “falling edge” for the second input, or vice versa.

Remark: n = 0 to 7
x = 0, 1
y = 0 when x = 1; y = 1 when x = 0

Figure 9-38: Basic Operation Timing of Simultaneous Pulse Width and Pulse Space Measurement



Note: The signal to measure has to be assigned to both inputs, TIPn0 and TIPn1. This can be done either by external pin connection, or internally when selecting TIPn1 input on TIPn0 pin. In case of internal connection the signal has to be input on TIPn0 pin.

Remarks: **1.** D_{00}, D_{01}, D_{02}: Values captured to TPnCCR0 register (0000H to FFFFH)

2. D_{10}, D_{11}: Values captured to TPnCCR1 register (0000H to FFFFH)

3. TIPn0: Set detection to rising edge (TPnIS1, TPnIS0 = 01B)

4. TIPn1: Set detection to falling edge (TPnIS3, TPnIS2 = 10B)

5. n = 0 to 7

9.5.9 Counter synchronous operation function

Timer P supports a function to start several timers P simultaneously. For this purpose two timer groups are defined, TMP0 to TMP3, as well as TMP4 to TMP7. For each timer group the counting of one to three slave counters (TMP1 to TMP3, or TMP5 to TMP7) can be synchronized with the corresponding master counter (TMP0 or TMP4). The synchronous operation function is enabled for each incorporated timer by the TPnSYE bit in the TPnCTL1 register (ref. to **9.4 (2) TMPn control register 1 (TPnCTL1)**).

When enabling the synchronous operation function, observe the following procedure:
<1> Clear the synchronous mode selection bit TPmSYE of the master counter TMPm to 0.
<2> Disable the count operation of the master counter TMPm (TPmCE = 0).
<3> Enable the synchronous operation for each of the incorporated slave counters TMPs (TPsSYE = 1).
<4> Enable the operation of the master counter TMPm (TPmCE = 1).

Master and incorporated slave counters of that group start and clock synchronously. When the master counter is cleared, the slave counters are cleared synchronously too.

Cautions: 1. In synchronous operation mode, the master counter can be used only in PWM mode (TPmMD2 to TPmMD0 = 100B), external trigger pulse output mode (TPmMD2 to TPmMD0 = 010B), one-shot pulse output mode (TPmMD2 to TPmMD0 = 011B), and free-running mode (TPmMD2 to TPmMD0 = 101B).

2. In synchronous operation mode, the slave counters can be used in free-running mode only (TPsMD2 to TPsMD0 = 101B).

Remark: n = 0 to 7,
m = 0, 4
s = 1 to 3, 5 to 7

Chapter 10 16-bit Inverter Timer/Counter R

10.1 Features

Timer R is a 16-bit timer/counter that provides various motor control functions.

- Count clock resolution: 31.25 ns min. (when using 32 MHz count clock)
- General-purpose timer and operation mode supporting various motor control methods
- Compare registers with reload buffers
- 10-bit dead time counter
  - Dead time value independently settable through normal phase →inverted phase →normal phase
- A/D conversion trigger signal generation
  - Generation of A/D conversion trigger with 2 compare registers, TRnCCR4 and TRnCCR5
  - Dedicated output pin (TORn7) set with the TRnADTRG0 signal and reset with the TRnADTRG1 signal
- Interrupt thinning out function
  - Thinning out rates of 1/1 to 1/32
- Forced output stop function: ESO
  - High-impedance output of pins TORn0 to TORn7 possible during ESOn input
- Compare value setting
  - Reload (batch rewrite)/anytime rewrite mode selectable **Note**
- Reload mode
  - Reload enabled by writing to TRnCCR1 register last, multiple registers simultaneity maintained
  - Peak/valley/peak and valley reload, transfer possible at reload timing **Note**
  - Provision of reload request flag TRnRSF
  - DMA transferable register address placement
- High-accuracy T-PWM mode
  - 0 to 100% duty PWM output possible, including dead time reduction
  - Increased output resolution without software load, because presence/absence of added pulse to PWM output on up-count side can be controlled with LSB of compare register
- 8 selectable count clocks: ϕ/2, ϕ/4, ϕ/8, ϕ/16, ϕ/ 32, ϕ/64, ϕ/256, ϕ/1024
- Active level of output pins TORn0 to TORn7 settable for each pin
- Fail-safe function (error interrupt output possible)
  - Simultaneous active output detection function in normal phase/inverted phase

Note: High-accuracy T-PWM mode

10.2 Configuration

Timer R is configured of the following hardware.

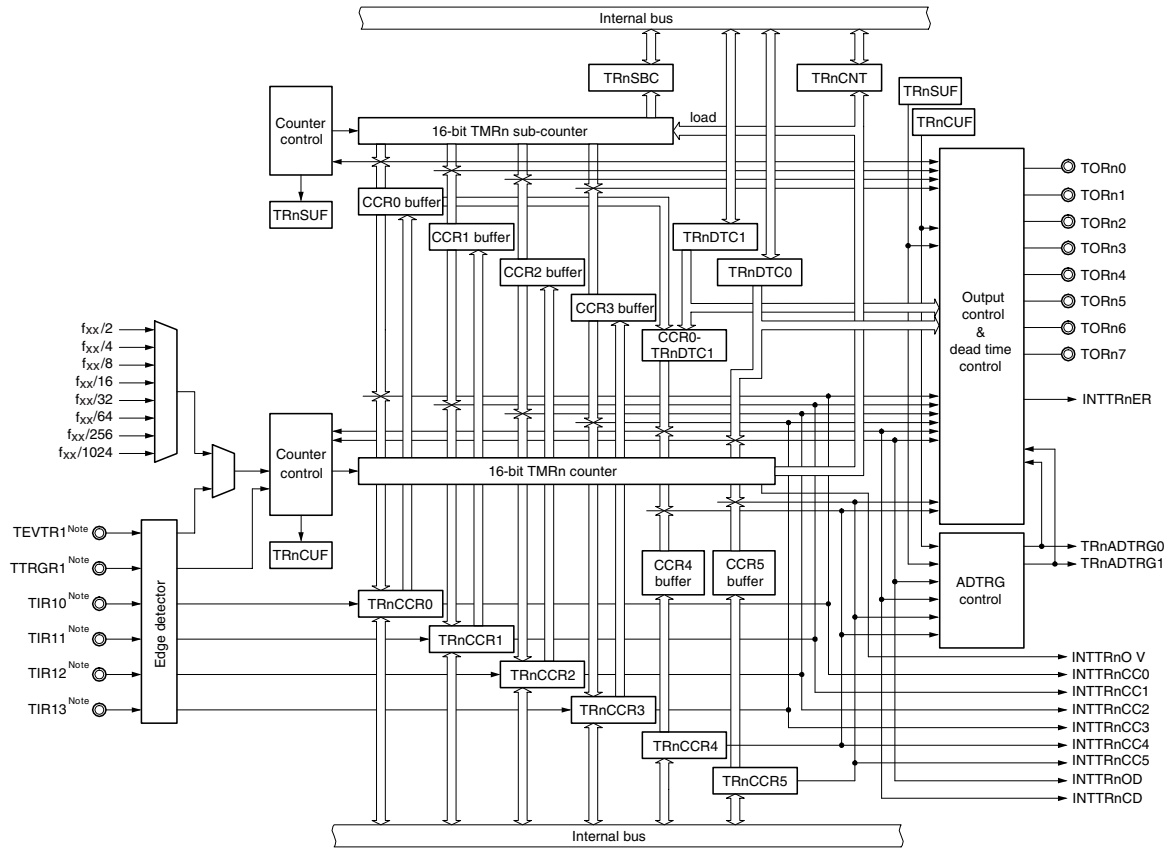
Table 10-1: Timer R Configuration

Item	Configuration
Counters	16-bit counter × 1 16-bit sub-counter × 1 10-bit dead time counter × 3
Registers	Timer Rn counter read register (TRnCNT) Timer Rn sub-counter read register (TRnSBC) Timer Rn dead time setting registers 0, 1 (TRnDTC0, TRnDTC1) Timer Rn capture/compare registers 0 to 3 (TRnCCR0-TRnCCR3) Timer Rn compare registers 4, 5 (TRnCCR4, TRnCCR5) TRnCCR0 to TRnCCR5 buffer registers TRnDTC0, TRnDTC1 buffer registers
Timer input pins	3 (TIR10 to TIR13, TTRGR1, TEVTR1, ESOn)**Note**
Timer output pins	8 (TORn0 to TORn7)**Note**
Timer input signal	-
Timer output signal	TRnADTRG0, TRnADTRG1
Control registers	Timer Rn control registers 0, 1 (TRnCTL0, TRnCTL1) Timer Rn I/O control registers 0 to 4 (TRnIOC0 to TRnIOC4) Timer Rn option registers 0 to 3, 6, 7 (TRnOPT0 to TRnOPT3, TRnOPT6, TRnOPT7)
Interrupt requests	Compare match interrupts (INTTRnCC0 to INTTRnCC5) Peak interrupt (INTTRnCD) Valley interrupt (INTTRnOD) Overflow interrupt (INTTRnOV) Error interrupt (INTTRnER)

Note: Alternate-function pins

Remark: n = 0, 1

Figure 10-1: Timer Rn Block Diagram



Note: Timer inputs are only available in TMR1 (n = 1). The TIR10 to TIR13 capture inputs are shared with TOR11 to TOR14. External trigger input TTRGR1 is shared with TIR10 output, and external event input TEVTR1 is shared with TIR17 output.

Remarks: 1. n = 0, 1

2. f_{XX}: Internal system clock

(1) TMRn capture/compare register 0 (TRnCCR0)

The TRnCCR0 register is a 16-bit register provided with a capture function and a compare function. In the case of the free-running mode only, bit TRnCCS0 of the TRnOPT0 register is used to select use of the register as a capture register or as a compare register.
In the pulse width measurement mode, this register can be used as a capture-only register. (The register cannot be used as a compare register.)
In modes other than the free-running mode and the pulse width measurement mode, the register is used as a compare-only register.
This register can be read and written in 16-bit units.
$\overline{\text{RESET}}$ input clears this register to 0000H.

Remarks: 1. In the high-accuracy T-PWM mode, writing to bit 0 of the TRnCCR0 register is ignored. Moreover, bit 0 is read as 0.

2. n = 0, 1

Figure 10-2: TMRn Capture/Compare Register 0 (TRnCCR0)

After reset: 0000H R/W Address: TR0CCR0 FFFFF598H, TR1CCR0 FFFFF5D8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRnCCR0																

(a) Use as compare register

When TRnCE = 1, the TRnCCR0 register write access method is as follows.

Timer Rn Operation Mode	TRnCCR0 Register Write Access Mode
PWM mode, external trigger pulse output mode, triangular wave PWM mode, PWM mode with dead time	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switchable

Remarks: 1. For details about the compare register rewrite operation, refer to **10.4.2 Compare register rewrite operation**.

2. n = 0, 1

Caution: To set the carrier frequency in the high-accuracy T-PWM mode, set the TRnCCR0 register as follows.
Number of count clocks of carrier frequency + TRnDTC0 register value + TRnDTC1 register value.
For details about the carrier wave and dead time settings, refer to 10.10.9 (4) Counter operation in high-accuracy T-PWM mode.

(b) Use as capture register

The counter value is saved to the TR1CCR0 register upon detection of the edge of the capture trigger (TIR10) input.

Remark: The capture function is provided only for TMR1.

(2) TMRn capture/compare register 1 (TRnCCR1)

The TRnCCR1 register is a 16-bit register that functions both as a capture register and a compare register.

When a compare register is rewritten in the reload mode, the reload request flag (TRnRSF) becomes 1 when write access is performed to the TRnCCR1 register, and all the registers are rewritten at the same time at the next reload timing.

In the free-running mode only, the TRnCCS1 bit of the TRnOPT0 register is used to select whether to use the TRnCCR1 register as a capture register or as a compare register.

In the pulse width measurement mode, the TRnCCR1 register can be used as a dedicated capture register. (The register cannot be used as a compare register.)

In modes other than the free-running mode and the pulse width measurement mode, all TRnCCR1 registers function as dedicated compare registers.

This register can be read and written in 16-bit units.

$\overline{\text{RESET}}$ input clears this register to 0000H.

Remarks: 1. In the high-accuracy T-PWM mode, when bit 0 is set to 1, the additional pulse control function is engaged. (For details about the additional pulse control function, refer to **10.10.9 (6) Additional pulse control in high-accuracy T-PWM mode**.)

2. n = 0, 1

Figure 10-3: TMRn Capture/Compare Register 1 (TRnCCR1)

After reset: 0000H R/W Address: TR0CCR1 FFFFF59EH, TR1CCR1 FFFFF5DEH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRnCCR1																

(a) Use as compare register

When TRnCE = 1, the TRnCCR1 register write access method is as follows.

Timer Rn Operation Mode	TRnCCR1 Register Write Access Mode
PWM mode, external trigger pulse output mode, triangular wave PWM mode, PWM mode with dead time	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switchable

Remarks: 1. For details about the compare register rewrite operation, refer to **10.4.2 Compare register rewrite operation**.

2. n = 0, 1

(b) Use as capture register

The counter value is saved to the TR1CCR1 register upon detection of the edge of the capture trigger (TIR11) input.

Remark: The capture function is provided only for TMR1.

(3) TMRn capture/compare register 2 (TRnCCR2)

The TRnCCR2 register is a 16-bit register that functions both as a capture register and compare register.

In the free-running mode only, bit TRnCCS2 of the TRnOPT0 register is used to select whether to use the TRnCCR2 register as a capture register or a compare register.

In the pulse width measurement mode, the TRnCCR2 register can be used as a dedicated capture register. (The register cannot be used as a compare register.)

In modes other than the free-running mode and the pulse width measurement mode, all TRnCCR2 registers function as dedicated compare registers.

This register can be read and written in 16-bit units.

$\overline{\text{RESET}}$ input clears this register to 0000H.

Remarks: 1. In the high-accuracy T-PWM mode, when bit 0 is set to “1”, the additional pulse control function is engaged. (For details about the additional pulse control function, refer to **10.10.9 (6) Additional pulse control in high-accuracy T-PWM mode**.)

2. n = 0, 1

Figure 10-4: TMRn Capture/Compare Register 2 (TRnCCR2)

After reset: 0000H R/W Address: TR0CCR2 FFFFF59CH, TR1CCR2 FFFFF5DCH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRnCCR2																

(a) Use as compare register

When TRnCE = 1, the TRnCCR2 register write access method is as follows.

Timer Rn Operation Mode	TRnCCR2 Register Write Access Mode
PWM mode, external trigger pulse output mode, triangular wave PWM mode, PWM mode with dead time	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload /anytime rewrite switchable

Remarks: 1. For details about the compare register rewrite operation, refer to **10.4.2 Compare register rewrite operation**.

2. n = 0, 1

(b) Use as capture register

The counter value is saved to the TRnCCR2 register upon detection of the edge of the capture trigger (TIR12) input.

Remark: The capture function is provided only for TMR1.

(4) TMRn capture/compare register 3 (TRnCCR3)

the TRnCCR3 register is a 16-bit register that functions both as a capture register and a compare register.
In the free-running mode only, bit TRnCCS3 of the TRnOPT0 register is used to select whether to use the TRnCCR3 register as a capture register or a compare register.
In the pulse width measurement mode, the TRnCCR3 register can be used as a dedicated capture register. (The register cannot be used as a compare register.)
In modes other than the free-running mode and the pulse width measurement mode, all TRnCCR3 registers function as dedicated compare registers.
This register can be read and written in 16-bit units.
$\overline{\text{RESET}}$ input clears this register to 0000H.

Remarks: 1. In the high-accuracy T-PWM mode, when bit 0 is set to “1”, the additional pulse control function is engaged. (For details about the additional pulse control function, refer to **10.10.9 (6) Additional pulse control in high-accuracy T-PWM mode**.)

2. n = 0, 1

Figure 10-5: TMRn Capture/Compare Register 3 (TRnCCR3)

After reset: 0000H R/W Address: TR0CCR3 FFFFF59AH, TR1CCR3 FFFFF5DAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRnCCR3																

(a) Use as compare register

When TRnCE = 1, the TRnCCR3 register write access method is as follows.

Timer Rn Operation Mode	TRnCCR3 Register Write Access Mode
PWM mode, external trigger pulse output mode, triangular wave PWM mode, PWM mode with dead time	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switchable

Remarks: 1. For details about the compare register rewrite operation, refer to **10.4.2 Compare register rewrite operation**.

2. n = 0, 1

(b) Use as capture register

The counter value is saved to the TR1CCR3 register upon detection of the edge of the capture trigger (TIR13) input.

Remark: The capture function is provided only for TMR1.

(5) TMRn compare register 4 (TRnCCR4)

The TRnCCR4 register is a 16-bit register that functions as a compare function.
In the high-accuracy T-PWM mode and the PWM mode with dead time, the interrupt for matches between the counter and the TRnCCR4 register can be selected as the timing for A/D conversion trigger input.
This register can be read and written in 16-bit units.
$\overline{\text{RESET}}$ input clears this register to 0000H.

Remarks: 1. In the high-accuracy T-PWM mode, bit 0 of the TRnCCR4 register is ignored.

2. n = 0, 1

Figure 10-6: TMRn Compare Register 4 (TRnCCR4)

After reset: 0000H R/W Address: TR0CCR4 FFFFF592H, TR1CCR4 FFFFF5D2H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRnCCR4																

When TRnCE = 1, the TRnCCR4 register write access method is as follows.

Timer Rn Operation Mode	TRnCCR4 Register Write Access Mode
PWM mode, external trigger pulse output mode, triangular wave PWM mode, PWM mode with dead time	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switchable

Remarks: 1. For details about the compare register rewrite operation, refer to **10.4.2 Compare register rewrite operation**.

2. n = 0, 1

(6) TMRn compare register 5 (TRnCCR5)

The TRnCCR5 register is a 16-bit compare register.
In the high-accuracy T-PWM mode and the PWM mode with dead time, the interrupt for matches between the counter and the TRnCCR5 register can be selected as the timing for A/D conversion trigger input.
This register can be read and written in 16-bit units.
$\overline{\text{RESET}}$ input clears this register to 0000H.

Remarks: 1. In the high-accuracy T-PWM mode, bit 0 of the TRnCCR5 register is ignored

2. n = 0, 1

Figure 10-7: TMRn Compare Register 5 (TRnCCR5)

After reset: 0000H R/W Address: TR0CCR3 FFFFF590H, TR1CCR3 FFFFF5D0H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRnCCR5																

When TRnCE = 1, the TRnCCR5 register write access method is as follows.

Timer Rn Operation Mode	TRnCCR5 Register Write Access Mode
PWM mode, external trigger pulse output mode, triangular wave PWM mode, PWM mode with dead time	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switchable

Remarks: 1. For details about the compare register rewrite operation, refer to **10.4.2 Compare register rewrite operation**.

2. n = 0, 1

(7) TMRn counter read register (TRnCNT)

The TRnCNT register is a timer read register that can read the values of the 16-bit counter.
This register can only be read in 16-bit units.
$\overline{\text{RESET}}$ input or setting TRnCE = 0 clears this register to 0000H.
During the interval from when CE = 1 until count up, the value of the TRnCNT register is FFFFH.

Figure 10-8: TMRn Counter Read Register (TRnCNT)

After reset: 0000H R Address: TR0CNT FFFFF5A4H, TR1CNT FFFFF5E4H

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

Note: In the high-accuracy T-PWM mode, bit 0 is read as “0”.

Remark: n = 0, 1

(8) TMRn sub-counter read register (TRnSBC)

The TRnSBC register can read the value of the 16-bit counter.
This register can only be read in 16-bit units.
$\overline{\text{RESET}}$ input or setting TRnCE = 0 clears this register to 0000H.

Remarks: 1. In the high-accuracy T-PWM mode, this register can be used only in the PWM mode with dead time.

2. n = 0, 1

Figure 10-9: TMRn Sub-Counter Read Register (TRnSBC)

After reset: 0000H R Address: TR0SBC FFFFF5A6H, TR1SBC FFFFF5E6H

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

Note: In the high-accuracy T-PWM mode, bit 0 is read as 0.

(9) TMRn dead time setting register 0 (TRnDTC0)

The TRnDTC0 register is a 10-bit register that specifies the dead time value.
This register can be read and written in 16-bit units.
$\overline{\text{RESET}}$ input clears this register to 0000H.
The dead time counter operates in the high-accuracy T-PWM mode and the PWM mode with dead time. In all other modes, be sure to set the TRnDTC0 register to 0000H.

Cautions: 1. When TRnCE = 1, do not rewrite TRnDTC0 with a different value.

2. When the TRnDTC0 register is set to 0000H, dead time is not inserted.

3. Bits 0 and 10 to 15 are fixed to 0.

Remark: n = 0, 1

Figure 10-10: TMRn Dead Time Setting Register 0 (TRnDTC0)

After reset: 0000H R/W Address: TR0DTC0 FFFFF5A0H, TR1DTC0 FFFFF5E0H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRnDTC0	0	0	0	0	0	0										0

(10) TMRn dead time setting register 1 (TRnDTC1)

The TRnDTC1 register is a 10-bit register that specifies the dead time value.
This register can be read and written in 16-bit units.
Reset input clears this register to 0000H.
The dead time counter operates in the high-accuracy T-PWM mode and the PWM mode with dead time. In all other modes, be sure to set the TRnDTC1 register to 0000H.

Cautions: 1. When TRnCE = 1, do not rewrite TRnDTC1 with a different value.

2. When the TRnDTC1 register is set to 0000H, dead time is not inserted.

3. Bits 0 and 10 to 15 are fixed to 0.

Remark: n = 0, 1

Figure 10-11: TMRn Dead Time Setting Register 1 (TRnDTC1)

After reset: 0000H R/W Address: TR0DTC1 FFFFF5A2H, TR1DTC1 FFFFF5E2H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRnDTC1	0	0	0	0	0	0										0

10.3 Control Registers

(1) TMRn control register 0 (TRnCTL0)

The TRnCTL0 register is an 8-bit register that controls the operation of timer Rn.
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input changes the value of this register to initial setting 00H.

Caution: When TRnCE = 1, do not rewrite bits other than bit TRnCE of the TRnCTL0 register.

Figure 10-12: TMRn Control Register 0 (TRnCTL0) (1/2)

After reset: 00H R/W Address: TR0CTL0 FFFFF580H, TR1CTL0 FFFFF5C0H

	7	6	5	4	3	2	1	0
TRnCTL0 (n = 0, 1)	TRnCE	0	0	0	0	TRnCKS2	TRnCKS1	TRnCKS0

TRnCE	Timer Rn Operation Control
0	Internal operating clock operation disabled (Reset timer Rn asynchronously)
1	Internal operating clock operation enabled

When bit TRnCE is set to “0”, the internal operation clock of timer Rn stops (fixed to low level), and timer Rn is set asynchronously.
When bit TRnCE is set to “1”, the internal operation of timer Rn is enabled from when bit TRnCE was set to “1” and count-up is performed. The time until count-up is as listed in Table 10-2, “TMRn Count Clock and Count Delay,” on page 311.

Remark: By setting TRnCE = 0 following functions of timer Rn are reset.

- Internal registers and internal latch circuits other than registers that can be written to/from the CPU
- TRnOVF flag and flags in TRnOPT6 register
- Counter, sub-counter, dead time counter, counter read register, sub-counter read register
- TRnCCR0 to TRnCCR5 buffer registers, TRnDTC0 buffer register, and TRnDTC1 buffer register
- Timer output (inactive level output)

Remark: n = 0, 1

Figure 10-12: TMRn Control Register 0 (TRnCTL0) (2/2)

TRnCKS2	TRnCKS1	TRnCKS0	Internal Count Clock Selection of Timer Rn
0	0	0	$f_{XX}/2$
0	0	1	$f_{XX}/4$
0	1	0	$f_{XX}/8$
0	1	1	$f_{XX}/16$
1	0	0	$f_{XX}/32$
1	0	1	$f_{XX}/64$
1	1	0	$f_{XX}/256$
1	1	1	$f_{XX}/1024$
Caution:	**Set bits TRnCKS2 to TRnCKS0 when TRnCE = 0. When bit TRnCE is set from 0 to 1, bits TRnCKS2 to TRnCKS0 can be simultaneously set.**		
Remark:	f_{XX}: System clock		

Remark: n = 0, 1

Table 10-2: TMRn Count Clock and Count Delay

Count Clocks	TTnCKS2	TTnCKS1	TTnCKS0	Count Delay	
				Minimum	Maximum
$f_{XX}/2$	0	0	0	3 base clocks	4 base clocks
$f_{XX}/4$	0	0	1		
$f_{XX}/8$	0	1	0		
$f_{XX}/16$	0	1	1	4 base clocks	5 base clocks + 1 count clock
$f_{XX}/32$	1	0	0		
$f_{XX}/64$	1	0	1		
$f_{XX}/256$	1	1	0		
$f_{XX}/1024$	1	1	1		

Remarks: 1. f_{XX}: System clock

2. f_{TMRn}: Base clock of timer Rn ($f_{TMRn} = f_{XX}/2$)

3. n = 0, 1

(2) TMRn control register 1 (TRnCTL1)

The TRnCTL1 register is an 8-bit register that controls the operation of timer Rn.
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input changes the value of this register to initial setting 00H.

Cautions: 1. In the one-shot pulse mode and external trigger pulse output mode, write access using "1", the same value as that of bit TRnEST, functions as one trigger.

2. Set bits TRnEEE and TRnMD2 to TRnMD0 when TRnCE = 0. (The same value as when TRnCE = 1 can be written). Do not perform rewrite when TRnCE = 1.

Figure 10-13: TMRn Control Register 1 (TRnCTL1) (1/2)

After reset: 00H R/W Address: TR0CTL1 FFFFF581H, TR1CTL1 FFFFF5C1H

	7	6	5	4	3	2	1	0
TRnCTL1 (n = 0, 1)	0	TRnEST	TRnEEE	0	TRnMD3	TRnMD2	TRnMD1	TRnMD0

TRnEST	Software Trigger Control
0	No operation
1	Enables software trigger control • In one-shot pulse mode: One-shot pulse software trigger • In external trigger pulse output mode: Pulse output software trigger
• The TRnEST bit functions as a software trigger in the one-shot pulse mode and the external trigger pulse output mode, if it is set to 1 when TRnCE = 1. Always write TRnEST = 1 when TRnCE = 1. • The read value of the TRnEST bit is always 0.	

TRnEEE	Count Clock Specification
0	Use the internal clock (selected with bits TRnCKS2 to TRnCKS0 of the TRnCTL0 register)
1	Use external clock input (TEVTR1 pin input edge)**Note**
• When TR1EEE = 1 (external clock input TEVTR1), the valid edge is specified by bits TR1EES1 and TR1EES0 of the TRnIOC2 register. **Note:** External clock input pin is not available for TMR0 (n = 0).	

Remark: n = 0, 1

Figure 10-13: TMRn Control Register 1 (TRnCTL1) (2/2)

TRnMD3	TRnMD2	TRnMD1	TRnMD0	Timer Mode Selection
0	0	0	0	Interval timer mode
0	0	0	1	External event count mode **Note 1**
0	0	1	0	External trigger pulse output mode **Note 2**
0	0	1	1	One-shot pulse mode
0	1	0	0	PWM mode
0	1	0	1	Free-running mode
0	1	1	0	Pulse width measurement mode **Note 1**
0	1	1	1	Triangular wave PWM mode
1	0	0	0	High accuracy T-PWM mode
1	0	0	1	PWM mode with dead time
Other than above				Setting prohibited

Notes: 1. Setting prohibited for TMR0.

2. For TMR0 an output pulse can be triggered only by software trigger (TR0EST = 1).

Remark: n = 0, 1

(3) TMRn I/O control register 0 (TRnIOC0)

The TRnIOC0 register is an 8-bit register that controls the timer output (pins TORn0 to TORn3).
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Caution: If the dead time cannot be secured or if spikes (noise) may occur on the output pin, set the TRnIOC0 register when TRnCE = 0. When TRnCE = 1, the TRnIOC0 register can be write accessed using the same value.

Figure 10-14: TMRn I/O Control Register 0 (TRnIOC0)

After reset: 00H R/W Address: TR0IOC0 FFFFF582H, TR1IOC0 FFFFF5C2H

	7	6	5	4	3	2	1	0
TRnIOC0 (n = 0, 1)	TRnOL3	TRnOE3	TRnOL2	TRnOE2	TRnOL1	TRnOE1	TRnOL0	TRnOE0

TRnOLm	Timer Output Level Setting of TORnm Pin
0	Active level = High level
1	Active level = Low level

TRnOEm	Timer Output Control (TORnm pin)
0	Disable timer output (inactive level is output)
1	Enable timer output

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

(4) TMR1 I/O control register 1 (TR1IOC1)

The TR1IOC1 register is an 8-bit register that controls the valid edge of external signal inputs (pins TIR10 to TIR13).

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Cautions: 1. Set the TR1IOC1 register when TR1CE = 0. When TR1CE = 1, write access to the TR1IOC1 register can be performed using the same value.

2. The TR1IOC1 register is valid only in the free-running mode and the pulse width measurement mode. In all other modes, capture operation is not performed.

Figure 10-15: TMR1 I/O Control Register 1 (TR1IOC1)

After reset: 00H R/W Address: FFFFF5C3H

	7	6	5	4	3	2	1	0
TR1IOC1	TR1IS7	TR1IS6	TR1IS5	TR1IS4	TR1IS3	TR1IS2	TR1IS1	TR1IS0

TR1IS7	TR1IS6	Capture Input (TIR13) Valid Edge Setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection

TR1IS5	TR1IS4	Capture Input (TIR12) Valid Edge Setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection

TR1IS3	TR1IS2	Capture Input (TIR11) Valid Edge Setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection

TR1IS1	TR1IS0	Capture Input (TIR10) Valid Edge Setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection

(5) TMR1 I/O control register 2 (TR1IOC2)

The TR1IOC2 register is an 8-bit register that controls the valid edge of external event count input (pin TEVTR1) and external trigger input (pin TTRGR1).
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Caution: Set the TR1IOC2 register when TR1CE = 0. When TR1CE = 1, write access to the TR1IOC2 register can be performed using the same value.

Figure 10-16: TMR1 I/O Control Register 2 (TR1IOC2)

After reset: 00H R/W Address: FFFFF5C4H

	7	6	5	4	3	2	1	0
TR1IOC2	0	0	0	0	TR1EES1	TR1EES0	TR1ETS1	TR1ETS0

TR1EES1	TR1EES0	External Event Counter Input (TEVTR1) Valid Edge Setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection
Remark:		Bits TR1EES1 and TR1EES0 are valid only when TR1CTL1 register bit TR1EEE = 1, or when the external event count mode (TR1CTL1 register bits TR1MD3 to TR1MD0 = 0001B) is set.

TR1ETS1	TR1ETS0	External Trigger Input (TTRGR1) Valid Edge Setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection
Remark:		Bits TR1ETS1 and TR1ETS0 are valid only when the external trigger pulse output mode or one-shot pulse mode (TR1CTL1 register bits TR1MD3 to TR1MD0 = 0010B or 0011B) is set.

(6) TMRn I/O control register 3 (TRnIOC3)

The TRnIOC3 register is an 8-bit register that controls timer output (pins TORn4 to TORn7).
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Caution: If the dead time cannot be secured or if spikes (noise) may occur on the output pin, set the TRnIOC3 register when TRnCE = 0. When TRnCE = 1, the TRnIOC0 register can be write accessed using the same value.

Figure 10-17: TMRn I/O Control Register 3 (TRnIOC3)

After reset: 00H R/W Address: TR0IOC3 FFFFF585H, TR1IOC3 FFFFF5C5H

TRnIOC3 (n = 0, 1)

7	6	5	4	3	2	1	0
TRnOL7	TRnOE7	TRnOL6	TRnOE6	TRnOL5	TRnOE5	TRnOL4	TRnOE4

TRnOLm	Timer Output Level Setting of TORnm Pin
0	Active level = High level
1	Active level = Low level

TRnOEm	Timer Output Control (TORnm pin)
0	Disable timer output (inactive level is output)
1	Enable timer output

Remark: n = 0, 1
m = 4 to 7

(7) TMRn I/O control register 4 (TRnIOC4)

The TRnIOC4 register is an 8-bit register that controls timer output error detection.
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Caution: Set the TRnIOC4 register when TRnCE = 0. When TRnCE = 1, write access to the TRnIOC4 register can be performed using the same value.

Figure 10-18: TMRn I/O Control Register 4 (TRnIOC4)

After reset: 00H R/W Address: TR0IOC3 FFFFF586H, TR1IOC3 FFFFF5C6H

	7	6	5	4	3	2	1	0
TRnIOC4 (n = 0, 1)	0	TRnTBA2	TRnTBA1	TRnTBA0	0	0	0	TRnEOC

TRnTBA2	Timer Outputs (TORn5/TORn6) True Bar Active Detection Control
0	No detection of simultaneous active state of pins TORn5 and TORn6
1	Detection of simultaneous active state of pins TORn5 and TORn6
Remark:	If simultaneous active state is detected when TRnTBA2 = 1, the TRnTBF flag is set (1), and an error interrupt (INTTRnER) is output.

TRnTBA1	Timer Outputs (TORn3/TORn4) True Bar Active Detection Control
0	No detection of simultaneous active state of pins TORn3 and TORn4
1	Detection of simultaneous active state of pins TORn3 and TORn4
Remark:	If simultaneous active state is detected when TRnTBA1 = 1, the TRnTBF flag is set (1), and an error interrupt (INTTRnER) is output.

TRnTBA0	Timer Outputs (TORn1/TORn2) True Bar Active Detection Control
0	No detection of simultaneous active state of pins TORn1 and TORn2
1	Detection of simultaneous active state of pins TORn1 and TORn2
Remark:	If simultaneous active state is detected when TRnTBA0 = 1, the TRnTBF flag is set (1), and an error interrupt (INTTRnER) is output.

TRnEOC	Error Interrupt Output Control
0	Disable output of error interrupt (INTTRnER)
1	Enable output of error interrupt (INTTRnER)
Remark:	For details about error interrupt control, refer to **10.9 Error Interrupts**.

Remark: n = 0, 1

(8) TMRn option register 0 (TRnOPT0)

The TRnOPT0 register is an 8-bit register that sets the capture/compare operation and detects overflow.

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Caution: When TR1CE = 1, do not rewrite bits TR1CCS3 to TR1CCS0.

Figure 10-19: TMRn Option Register 0 (TRnOPT0) (1/2)

After reset: 00H R/W Address: FFFFF587H

	7	6	5	4	3	2	1	0
TR0OPT0	0	0	0	0	0	TR0CMS	TR0CUF	TR0OVF

After reset: 00H R/W Address: FFFFF5C7H

	7	6	5	4	3	2	1	0
TR1OPT0	TR1CCS3	TR1CCS2	TR1CCS1	TR1CCS0	0	TR1CMS	TR1CUF	TR1OVF

TR1CCS3	TR1CCR3 register capture/compare selection
0	Select compare register
1	Select capture register
Remark:	Bit TR1CCS3 is only valid in the free-running mode. In all other modes, this bit is invalid.

TR1CCS2	TR1CCR2 register capture/compare selection
0	Select compare register
1	Select capture register
Remark:	Bit TR1CCS2 is only valid in the free-running mode. In all other modes, this bit is invalid.

TR1CCS1	TR1CCR1 register capture/compare selection
0	Select compare register
1	Select capture register
Remark:	Bit TR1CCS1 is only valid in the free-running mode. In all other modes, this bit is invalid.

TR1CCS0	TRnCCR0 register capture/compare selection
0	Select compare register
1	Select capture register
Remark:	Bit TR1CCS0 is only valid in the free-running mode. In all other modes, this bit is invalid.

Remark: n = 0, 1

Figure 10-19: TMRn Option Register 0 (TRnOPT0) (2/2)

TRnCMS	Compare Register Transfer Timing Mode Selection
0	Reload mode (batch rewrite): When the TRnCCR1 register is written to, all the registers are updated at the next reload timing (reload). Even if registers other than the TRnCCR1 register are written, reload is not executed.
1	Anytime rewrite mode: Each register is updated independently, and when write access is performed to a compare register, the register is updated to the value used during anytime write access. Several clocks are required until the value is transferred to the register following write. (Refer to **10.4.2 (1) Anytime rewrite**.)
Remark:	The TRnCMS bit is valid only in the high-accuracy T-PWM mode, In all other modes it is invalid and has to be cleared (TRnCMS = 0).

TRnCUF	Timer R Counter Up/Down Detection Flag
0	The timer counter is in up count state.
1	The timer counter is in down count state.
Remark:	The TRnCUF bit is valid only in the high-accuracy T-PWM mode and triangular wave PWM mode. In all other modes, it is invalid (TRnCUF = 0).

TRnOVF	Timer R Overflow Detection Flag
0	No overflow occurrence (after bit was cleared)
1	Overflow occurrence
Remarks:	**1.** The TRnOVF bit is set (1) when the 16-bit counter value overflows from FFFFH to 0000H. **2.** The TRnOVF bit is cleared (0) when either 0 is written to it, or TRnCE = 0 is set. **3.** When TRnOVF bit is set (1), an overflow interrupt (INTTRnOV) is simultaneously output.
Cautions:	**1. Overflow can only occur in the free-running mode and the T-PWM mode. If, in the high-accuracy T-PWM mode, the set conditions for the TRnDTC0 and TRnDTC1 registers are incorrect, the TRnOVF bit may be set (1).** **2. When TRnOVF = 1, even if the TRnOVF bit and the TRnOPT0 register are read, the TRnOVF bit is not cleared.** **3. The TRnOVF bit can be read and written, but even if "1" is written to TRnOVF bit from the CPU, this is ignored.**

Remark: n = 0, 1

(9) TMRn option register 1 (TRnOPT1)

The TRnOPT1 register is an 8-bit register used to enable/disable peak/valley interrupts and set interrupt thinning out.
This register can be read and written in 16-bit or 8-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Cautions: 1. The TRnOPT1 register write method is as follows.

- **In high-accuracy T-PWM mode: Anytime write, or reload write**
- **In mode other than high-accuracy T-PWM mode: Reload write**

2. Do not set TRnICE = 0 and TRnIOE = 0.
Since reload does not occur when TRnICE, TRnIOE = 00, the TRnOPT1 register, which is a reload write register, stops being updated.

Figure 10-20: TMRn Option Register 1 (TRnOPT1) (1/2)

After reset: 00H R/W Address: TR0OPT1 FFFFF58EH
TR1OPT1 FFFFF5CEH

	7	6	5	4	3	2	1	0
TRnOPT1 (n = 0, 1)	TRnICE	TRnIOE	TRnRDE	TRnID4	TRnID3	TRnID2	TRnID1	TRnID0

TRnICE	Peak Interrupts (INTTRnCD) Control
0	Disable peak interrupt (INTTRnCD) output in the counter's peak timing Interrupt thinning out is not performed. Reload operation is disabled in the counter's peak timing.
1	Enable peak interrupt (INTTRnCD) in the counter's peak timing Interrupt thinning out is performed. Reload operation is enabled in the counter's peak timing.
Remark:	Bit TRnICE is valid only in the PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time.

TRnIOE	Valley Interrupt (INTTRnOD) Control
0	Disable valley interrupt (INTTRnOD) output in the counter's valley timing Reload operation is disabled in the counter's valley timing.
1	Enable valley interrupt (INTTRnOD) output in the counter's valley timing Reload operation is enabled in the counter's valley timing.
Remark:	Bit TRnIOE is valid only in the high-accuracy T-PWM mode and triangular wave PWM mode.

Remark: n = 0, 1

Figure 10-20: TMRn Option Register 1 (TRnOPT1) (2/2)

TRnRDE	Reload Timing Thinning Out control
0	Don't perform reload thinning out Reload timing occurs at each peak/valley.
1	Perform reload thinning out Reload timing occurs at the same interval as interrupt thinning out.
Remark:	Bit TRnRDE is valid only in the PWM mode, high-accuracy T-PWM mode, triangular wave PWM output mode, and PWM mode with dead time.

TRnID4	TRnID3	TRnID2	TRnID1	TRnID0	Interrupt Thinning Out Rate
0	0	0	0	0	No thinning out
0	0	0	0	1	1/2
0	0	0	1	0	1/3
0	0	0	1	1	1/4
. . .	. . .	. . .	. . .	. . .	. . .
1	1	1	0	1	1/30
1	1	1	1	0	1/31
1	1	1	1	1	1/32
Caution:	**If, when TRnCE = 1, the TRnOPT1 register is write accessed (including same value to bits TRnID4 to TRnID0), the interrupt thinning out counter is cleared.**				
Remark:	Bits TRnID0 to TRnID4 are valid only in the PWM mode, high-accuracy T-PWM mode, triangular wave PWM mode, and PWM mode with dead time.				

Remark: n = 0, 1

(10) TMRn option register 2 (TRnOPT2)

The TRnOPT2 register is an 8-bit register that controls A/D conversion trigger output (TRnADTRG0 signal).

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Caution: The settings of the TRnCCR5 and TRnCCR4 registers have an influence on the PWM output of pins TORn5 and TORn4 at the same time as the TRnADTRG0 signal output. Therefore, if setting bits TRnAT05 to TRnAT02, it is recommended to set the TRnOPT3 register as follows.

- **In the triangular wave PWM mode, when setting TRnAT05 = 1, set TRnOE5 = 0.**
- **In the PWM mode and triangular wave PWM mode, when setting TRnAT04 = 1, set TRnOE5 = 0.**
- **In the triangular wave PWM mode, when setting TRnAT03 = 1, set TRnOE4 = 0**
- **In the PWM mode and the triangular wave PWM mode, when setting TRnAT02 = 1, set TRnOE4 = 0.**

Figure 10-21: TMRn Option Register 2 (TRnOPT2) (1/2)

After reset: 00H R/W Address: TR0OPT2 FFFFF588H
TR1OPT2 FFFFF5C8H

	7	6	5	4	3	2	1	0
TRnOPT2 (n = 0, 1)	0	0	TRnAT05	TRnAT04	TRnAT03	TRnAT02	TRnAT01	TRnAT00

TRnAT05	TRnAT04	A/D Converter Trigger Signal (TRnADTRG0) Generation with Occurrence of Compare Match Interrupt (INTTRnCCR5)
0	0	No trigger signal is generated when INTTRnCCR5 occurs.
0	1	Trigger signal is generated, when INTTRnCCR5 occurs and TMRn is counting up.
1	0	Trigger signal is generated, when INTTRnCCR5 occurs and TMRn is counting down.
1	1	Trigger signal is generated, when INTTRnCCR5 occurs in any state (TMRn is counting up or down)

Cautions: 1. Bit TRnAT05 can be set to 1 only in the triangular wave PWM mode and high-accuracy T-PWM mode. In all other modes, be sure to set this bit to 0.

2. Bit TRnAT04 can be set to 1 only in the PWM mode, triangular wave PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. In all other modes, be sure to set this bit to 0.

Remark: n = 0, 1

Figure 10-21: TMRn Option Register 2 (TRnOPT2) (2/2)

TRnAT03	TRnAT02	A/D Converter Trigger Signal (TRnADTRG0) Generation with Occurrence of Compare Match Interrupt (INTTRnCCR4)
0	0	No trigger signal is generated when INTTRnCCR4 occurs.
0	1	Trigger signal is generated, when INTTRnCCR4 occurs and TMRn is counting up.
1	0	Trigger signal is generated, when INTTRnCCR4 occurs and TMRn is counting down.
1	1	Trigger signal is generated, when INTTRnCCR4 occurs in any state (TMRn is counting up or down)

Cautions: 1. Bit TRnAT03 can be set to 1 only in the triangular wave PWM mode and high-accuracy T-PWM mode. In all other modes, be sure to set this bit to 0.

2. Bit TRnAT02 can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, triangular wave PWM mode, and PWM mode with dead time. In all other modes, be sure to set this bit to 0.

TRnAT01	A/D Converter Trigger Signal (TRnADTRG0) Generation with Occurrence of Peak Interrupt (INTTRnCD)
0	No trigger signal is generated when peak interrupt (INTTRnCD) occurs.
1	Trigger signal is generated when peak interrupt (INTTRnCD) occurs after thinning out.

Caution: Bit TRnAT01 can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. In all other modes, be sure to set this bit to 0.

Remark: When bit TRnAT01 is set (1) the trigger signal coincides with the peak interrupt (INTTRnCD) controlled by the TRnOPT1 register (including thinning out).

TRnAT00	A/D Converter Trigger Signal (TRnADTRG0) Generation with Occurrence of Valley Interrupt (INTTRnOD)
0	No trigger signal is generated when valley interrupt (INTTRnOD) occurs.
1	Trigger signal is generated when valley interrupt (INTTRnOD) occurs after thinning out.

Caution: Bit TRnAT00 can be set to 1 only in the high-accuracy T-PWM mode and triangular wave PWM mode. In all other modes, be sure to set this bit to 0.

Remark: When bit TRnAT00 is set (1) the trigger signal coincides with the valley interrupt (INTTRnOD) controlled by the TRnOPT1 register (including thinning out).

Remark: n = 0, 1

(11) TMRn option register 3 (TRnOPT3)

The TRnOPT3 register is an 8-bit register that controls A/D conversion trigger output (signal TRnADTRG1).

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

$\overline{\text{RESET}}$ input clears this register to 00H.

Caution: The settings of the TRnCCR5 and TRnCCR4 registers have an influence on the PWM outputs of pins TORn5, TORn4 at the same time as the TRnADTRG0 signal output. Therefore, if setting bits TRnAT15 to TRnAT12, it is recommended to set the TRnOPT3 register as follows.

- **In the triangular wave PWM mode, when setting TRnAT15 = 1, set TRnOE5 = 0.**
- **In the PWM mode and triangular wave PWM mode, when setting TRnAT14 = 1, set TRnOE5 = 0.**
- **In the triangular wave PWM mode, when setting TRnAT13 = 1, set TRnOE4 = 0**
- **In the PWM mode and the triangular wave PWM mode, when setting TRnAT12 = 1, set TRnOE4 = 0.**

Figure 10-22: TMRn Option Register 3 (TRnOPT3) (1/2)

After reset: 00H R/W Address: TR0OPT3 FFFFF589H, TR1OPT3 FFFFF5C9H

	7	6	5	4	3	2	1	0
TRnOPT3 (n = 0, 1)	0	0	TRnAT15	TRnAT14	TRnAT13	TRnAT12	TRnAT11	TRnAT10

TRnAT15	TRnAT14	A/D Converter Trigger Signal (TRnADTRG1) Generation with Occurrence of Compare Match Interrupt (INTTRnCCR5)
0	0	No trigger signal is generated when INTTRnCCR5 occurs.
0	1	Trigger signal is generated, when INTTRnCCR5 occurs and TMRn is counting up.
1	0	Trigger signal is generated, when INTTRnCCR5 occurs and TMRn is counting down.
1	1	Trigger signal is generated, when INTTRnCCR5 occurs in any state (TMRn is counting up or down)

Cautions: 1. Bit TRnAT15 can be set to 1 only in the triangular wave PWM mode and high-accuracy T-PWM mode. In all other modes, be sure to set this bit to 0.

2. Bit TRnAT14 can be set to 1 only in the PWM mode, triangular wave PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. In all other modes, be sure to set this bit to 0.

Remark: n = 0, 1

Figure 10-22: TMRn Option Register 3 (TRnOPT3) Format (2/2)

TRnAT13	TRnAT12	A/D Converter Trigger Signal (TRnADTRG1) Generation with Occurrence of Compare Match Interrupt (INTTRnCCR4)
0	0	No trigger signal is generated when INTTRnCCR4 occurs.
0	1	Trigger signal is generated, when INTTRnCCR4 occurs and TMRn is counting up.
1	0	Trigger signal is generated, when INTTRnCCR4 occurs and TMRn is counting down.
1	1	Trigger signal is generated, when INTTRnCCR4 occurs in any state (TMRn is counting up or down)

Cautions: 1. Bit TRnAT13 can be set to 1 only in the triangular wave PWM mode and high-accuracy T-PWM mode. In all other modes, be sure to set this bit to 0.

2. Bit TRnAT12 can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, triangular wave PWM mode, and PWM mode with dead time. In all other modes, be sure to set this bit to 0.

TRnAT11	A/D Converter Trigger Signal (TRnADTRG1) Generation with Occurrence of Peak Interrupt (INTTRnCD)
0	No trigger signal is generated when peak interrupt (INTTRnCD) occurs.
1	Trigger signal is generated when peak interrupt (INTTRnCD) occurs after thinning out.

Caution: Bit TRnAT11 can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. In all other modes, be sure to set this bit to 0.

Remark: When bit TRnAT11 is set (1) the trigger signal coincides with the peak interrupt (INTTRnCD) controlled by the TRnOPT1 register (including thinning out).

TRnAT10	A/D Converter Trigger Signal (TRnADTRG1) Generation with Occurrence of Valley Interrupt (INTTRnOD)
0	No trigger signal is generated when valley interrupt (INTTRnOD) occurs.
1	Trigger signal is generated when valley interrupt (INTTRnOD) occurs after thinning out.

Caution: Bit TRnAT10 can be set to 1 only in the high-accuracy T-PWM mode and triangular wave PWM mode. In all other modes, be sure to set this bit to 0.

Remark: When bit TRnAT10 is set (1) the trigger signal coincides with the valley interrupt (INTTRnOD) controlled by the TRnOPT1 register (including thinning out).

Remark: n = 0, 1

(12) TMRn option register 6 (TRnOPT6)

The TRnOPT6 register is an 8-bit register that controls the various flags of timer Rn.
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input or setting TRnCE = 0 clears this register to 00H.

Remark: For the functions of the various flags, refer to **10.6 Flags**.

Figure 10-23: TMRn Option Register 6 (TRnOPT6)

After reset: 00H R/W Address: TR0OPT6 FFFFF58CH
TR1OPT6 FFFFF5CCH

	7	6	5	4	3	2	1	0
TRnOPT6	0	0	0	0	0	TRnTBF	TRnSUF	TRnRSF

(n = 0, 1)

TRnTBF	True Bar Active Detection Flag
0	Normal phase and inverted phase are not simultaneously active.
1	Normal phase and inverted phase are simultaneously active.
This flag detects when the normal phase and inverted phase are simultaneously active, while any of bits TRnTBA2 to TRnTBA0 of the TRnIOC4 register is 1. When bits TRnTBA2 to TRnTBA0 = 000B, simultaneous active is not detected.	
Remarks: 1. The TRnTBF flag is set (1) upon detection that any of the normal phases (TORn1, TORn3, TORn5) and inverted phases (TORn2, TORn4, TORn6) are simultaneously active, and an error interrupt (INTTRnER) is output at such time.	
2. This flag can be cleared by writing “0” to it.	

TRnSUF	Timer R Sub-Counter Up/Down Detection Flag
0	Sub-counter is counting up
1	Sub-counter is counting down
The TRnSUF flag detects sub-counter counting from 0000H until (TRnCCR0 register value - 2) as up count, and counting from TRnCCR0 register value until 0002H as down count.	
Remarks: 1. The TRnSUF flag is a read-only flag.	
2. The TRnSUF flag is valid only in the high-accuracy T-PWM mode.	

TRnRSF	Reload Suspension Flag
0	Write access to TRnCCR0 to TRnCCR5 and TRnOPT1 registers is enabled (no reload request, or completion of reload).
1	Write access to TRnCCR0 to TRnCCR5 and TRnOPT1 registers is disabled (reload request was output).
The TRnRSF flag indicates output of a reload request. It indicates that the data to be transferred next will be held in the TRnCCR0 to TRnCCR5 and TRnOPT1 registers. The TRnRSF flag is set (1) upon write to the TRnCCR1 register, and cleared (0) upon reload completion.	

Remark: n = 0, 1

(13) TMRn option register 7 (TRnOPT7)

The TRnOPT7 register is an 8-bit register that controls time output switching.
This register can be read and written in 8-bit or 1-bit units.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 10-24: TMRn Option Register 7 (TRnOPT7)

After reset: 00H R/W Address: TR0OPT7 FFFFF58DH
TR1OPT7 FFFFF5CDH

	7	6	5	4	3	2	1	0
TRnOPT7 (n = 0, 1)	0	0	0	0	0	0	0	TRnTOS

TRnTOS	Timer Output (TORn0) Switching Control
0	Output counter's (TRnCNT) up/down count flag to TORn0 pin
1	Output sub-counter's (TRnSBC) up/down count flag to TORn0 pin
When TRnTOS = 0, the status of bit TRnCUF of the TRnOPT0 register is output to pin TORn0. When TRnTOS = 1, the status of bit TRnSUF of the TRnOPT6 register is output to the TORn0 pin. **Remark:** The TRnTOS bit is valid only in the high-accuracy T-PWM mode.	

Remark: n = 0, 1

10.4 Basic Operation

10.4.1 Basic counter operation

This section describes the basic operation of the 16-bit counter. For details, refer to the description of the operation of each mode.

(1) Count start operation

The 16-bit counter of timer R starts counting from initial value FFFFH in all the modes except the high-accuracy T-PWM mode.
The counter counts up FFFFH, 0000H, 0001H, 0002H, 0003H, …
For the count operation in the high-accuracy T-PWM mode refer to section **10.10.9 (4) Counter operation in high-accuracy T-PWM mode**.

(2) Clear operation

The 16-bit counter is cleared to 0000H upon a match between the 16-bit counter and the compare register. Counting immediately following the start of count operation and counting from FFFFH to 0000H in the case of overflow are not detected as clear operations.

(3) Overflow operation

16-bit counter overflow occurs when the value of the 16-bit counter changes from FFFFH to 0000H. When overflow occurs, bit TRnOVF of the TRnOPT0 register is set (to 1), and an interrupt (INTTRnOV) is output. No overflow interrupt (INTTRnOV) is output under the following conditions.

- Immediately after count operation start
- When compare value is matched and cleared at FFFFH

Caution: Be sure to check that the overflow flag (TRnOVF) is set to 1 following output of the overflow interrupt (INTTRnOV).

(4) Counter read operation during count operation

In the case of timer R, the value of the 16-bit counter can be read by the TRnCNT register during count operation.

(5) Interrupt operation

In the case of timer R, the following interrupts are output.

- INTTRnCC0: Functions as TRnCCRn0 buffer register match interrupt.
- INTTRnCC1: Functions as TRnCCRn1 buffer register match interrupt.
- INTTRnCC2: Functions as TRnCCRn2 buffer register match interrupt.
- INTTRnCC3: Functions as TRnCCRn3 buffer register match interrupt.
- INTTRnCC4: Functions as TRnCCRn4 buffer register match interrupt.
- INTTRnCC5: Functions as TRnCCRn5 buffer register match interrupt.
- INTTRnCD: Functions as a peak interrupt at the timing when the counter switches from up count to down count.
- INTTRnOD: Functions as a valley interrupt at the timing when the counter switches from up count to down count.
- INTTRnOV: Functions as an overflow interrupt.
- INTTRnER: Functions as an normal phase/inverted phase simultaneous active detection interrupt.

Remark: n = 0, 1

10.4.2 Compare register rewrite operation

In the PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, external trigger pulse output mode, and triangular wave PWM mode, the reload function is valid. (In all other modes, reload-related settings are invalid.)
The compare/control registers with the reload function are listed below.

- TRnCCR0 to TRnCCR5
- TRnOPT1

Compare registers with the reload function can be rewritten in the following modes.

- **Anytime rewrite mode**
  In this mode, each compare register is updated independently, and when a compare register is written to, the register is updated to the value written during anytime write access.

- **Reload mode (batch rewrite)**
  When the TRnCCR1 register is written to, all the registers are updated at the next reload timing (reload).
  Reload does not occur even if a register other than the TRnCCR1 register is written to.
  A reload request flag (TRnRSF) is provided.

The compare register can be rewritten using DMA transfer. DMA transfer is performed as follows.

Address	Register Name	DMA Transfer Sequence
FFFFF590H	TR0CCR5	
FFFFF592H	TR0CCR4	
FFFFF594H	-	Note
FFFFF596H	-	Note
FFFFF598H	TR0CCR0	
FFFFF59AH	TR0CCR3	
FFFFF59CH	TR0CCR2	
FFFFF59EH	TR0CCR1	

Address	Register Name	DMA Transfer Sequence
FFFFF5D0H	TR1CCR5	
FFFFF5D2H	TR1CCR4	
FFFFF5D4H	-	Note
FFFFF5D6H	-	Note
FFFFF5D8H	TR1CCR0	
FFFFF5DAH	TR1CCR3	
FFFFF5DCH	TR1CCR2	
FFFFF5DEH	TR1CCR1	

Note: Dummy data transfer

For details about the interrupt thinning out function specified by setting the TRnOPT1 register, refer to **10.7 Interrupt Thinning Out Function**.

Mode	Rewrite Timing
Interval mode	Anytime rewrite
External event count mode	Anytime rewrite
External trigger pulse output mode	Reload
One-shot pulse mode	Anytime rewrite
PWM mode	Reload
Free-running mode	Anytime rewrite
Pulse width measurement mode	Reload
Triangular wave PWM mode	Reload **Note 1**
High-accuracy T-PWM mode	Anytime rewrite, Reload **Note 2**
PWM mode with dead time	Reload

Notes: 1. Rewrite is performed upon valley interrupt.

2. Set with TRnOPT0 register bit TRnCMS = 0, TRnOPT1 register bit TRnRDE = 0

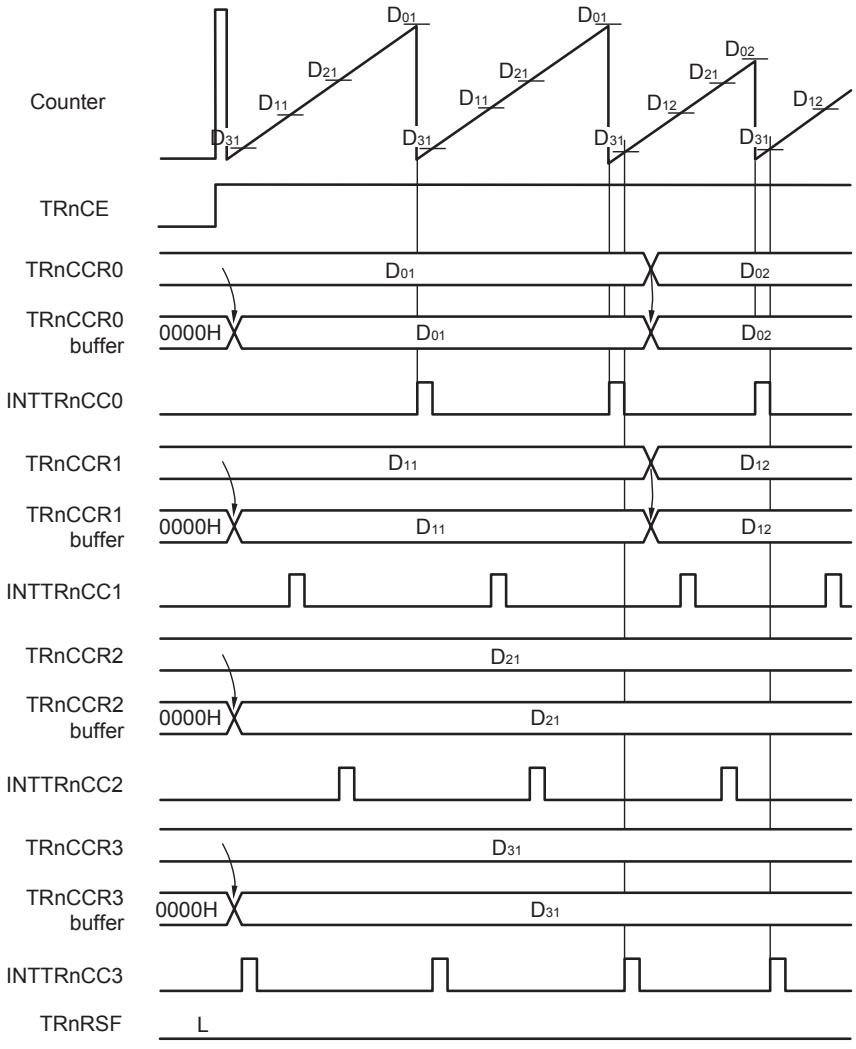
(1) Anytime rewrite

Anytime rewrite is set by setting TRnOPT0 register bit TRnCMS = 1. The TRnOPT1 register bit TRnRDE setting is ignored.

In this mode, the value written to each compare register is immediately transferred to the internal buffer register and compared to the counter value.

Following write to a compare register (TRnCCR0 register, etc.), the value is transferred to the internal buffer register after the lapse of 4 clocks (f_{TMRn}). However, since only the TRnCCR1 register has a 2-stage configuration, the actual transfer timing is after the lapse of 5 clocks (f_{TMRn}).

Figure 10-25: Anytime Rewrite Timing



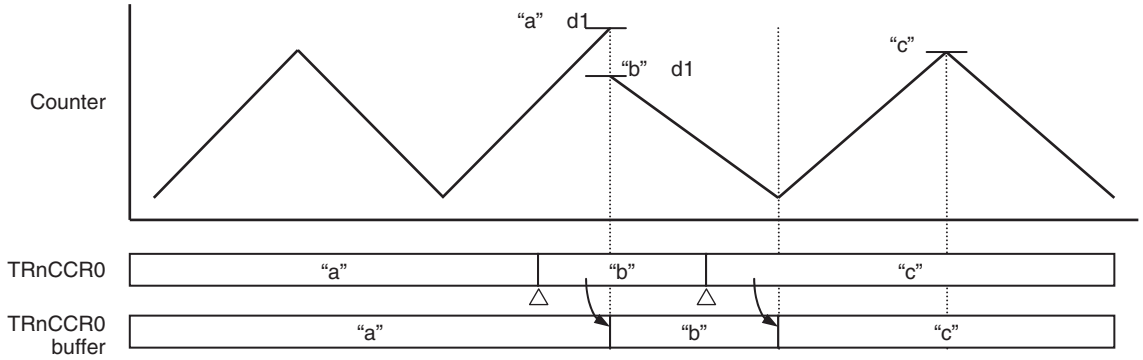
Remarks: 1. D01, D02: TRnCCR0 register setting value (0000H to FFFFH)
D11, D12: TRnCCR1 register setting value (0000H to FFFFH)
D21: TRnCCR2 register setting value (0000H to FFFFH)
D31: TRnCCR3 register setting value (0000H to FFFFH)

2. Timing chart using interval timer mode as an example

3. n = 0, 1

(a) Cautions related to rewriting TRnCCR0 register in high-accuracy T-PWM mode

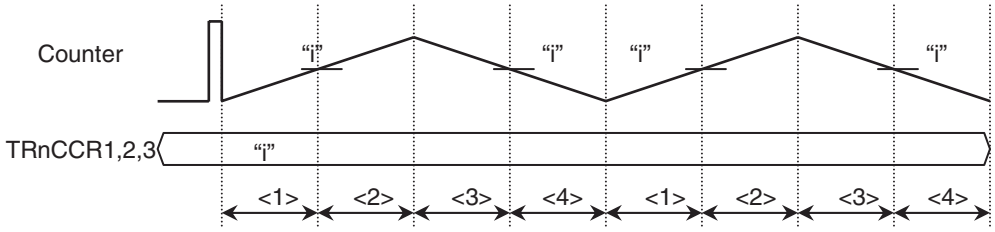
When the TRnCCR0 register is rewritten during operation using the anytime rewrite function, anytime transfer of the value to the TRnCCR0 buffer register is not performed. The timing is shown below.



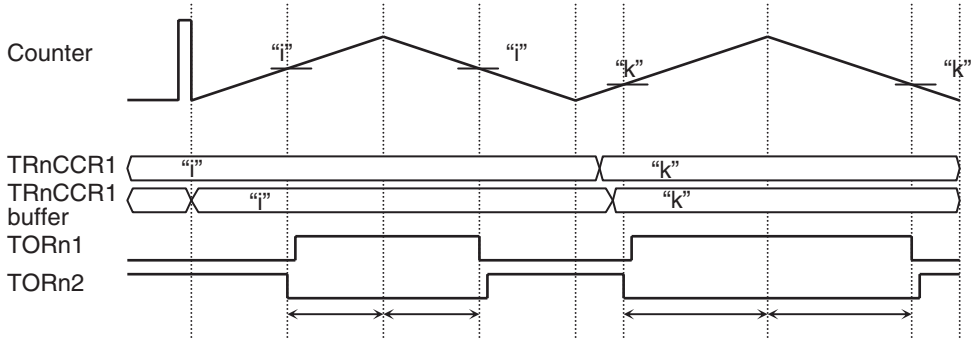
Remark: d1: TRnDTC1 setting value

Following write to the TRnCCR0 register, the value of the TRnCCR0 register is transferred to the TRnCCR0 buffer register at the next peak or at the valley timing. Since TRnCMS = 1 (anytime rewrite), the settings of bits TRnIOE, TRnICE, TRnRDE, and TRnID4 to TRnID0 have no influence.

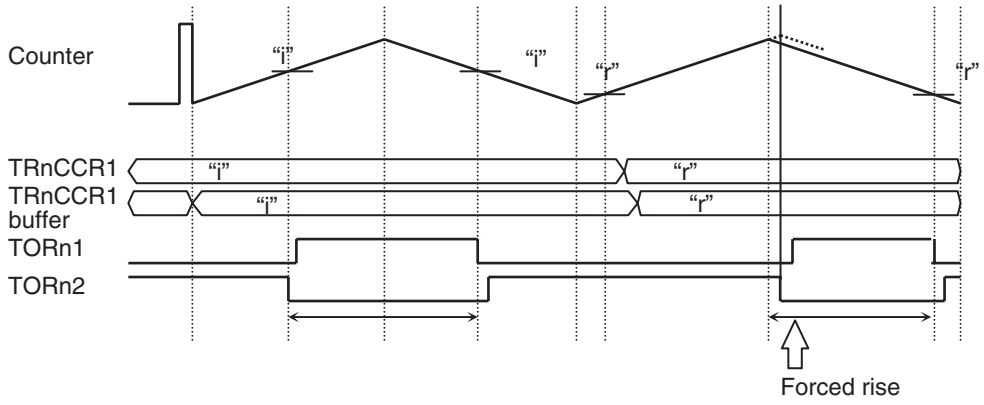
(b) Cautions related to rewriting of TRnCCR1 to TRnCCR3 registers



Rewrite in <1> interval (rewrite before match occurrence)
In the case of rewrite before a match between the TRnCCR1 to TRnCCR3 registers and the counter occurs, a match with the counter occurs following rewrite and the rewrite value is instantly reflected.

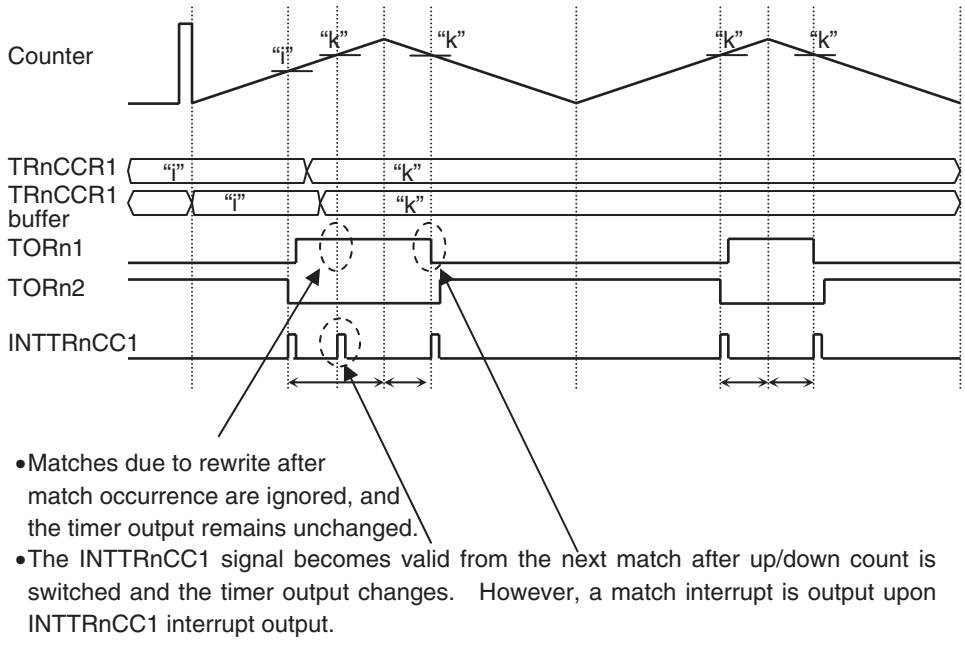


If a value smaller than the counter value is written before match occurrence, no match occurs, so the following output wave results.

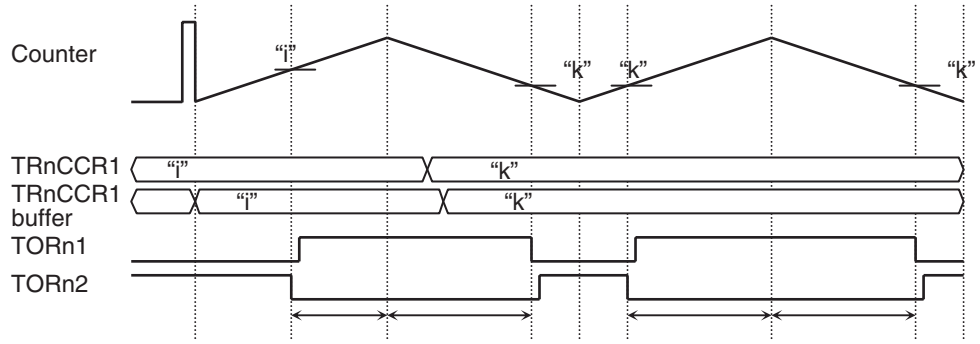


If no match occurs, the timer output remains unchanged.
However, even if a match does not occur the timer output is forcibly changed to normal phase active level at peaks.

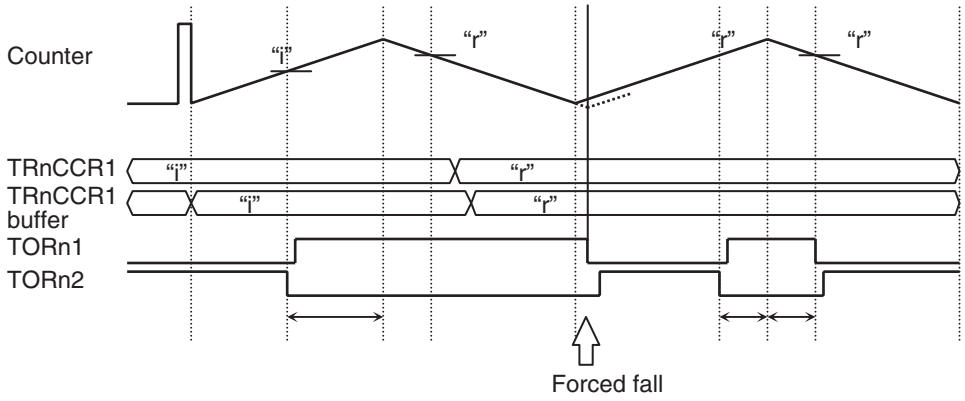
Rewrite in <2> interval (rewrite after match occurrence)
In the case of rewrite after a match between the TRnCCR1 to TRnCCR3 registers and the counter occurs, further match occurrences are ignored, so the rewrite value is not reflected.



Rewrite in <3> interval (rewrite before match occurrence)
In the case of rewrite before a match between the TRnCCR1 to TRnCCR3 registers and the counter occurs, a match with the counter occurs following rewrite and the rewrite value is instantly reflected.

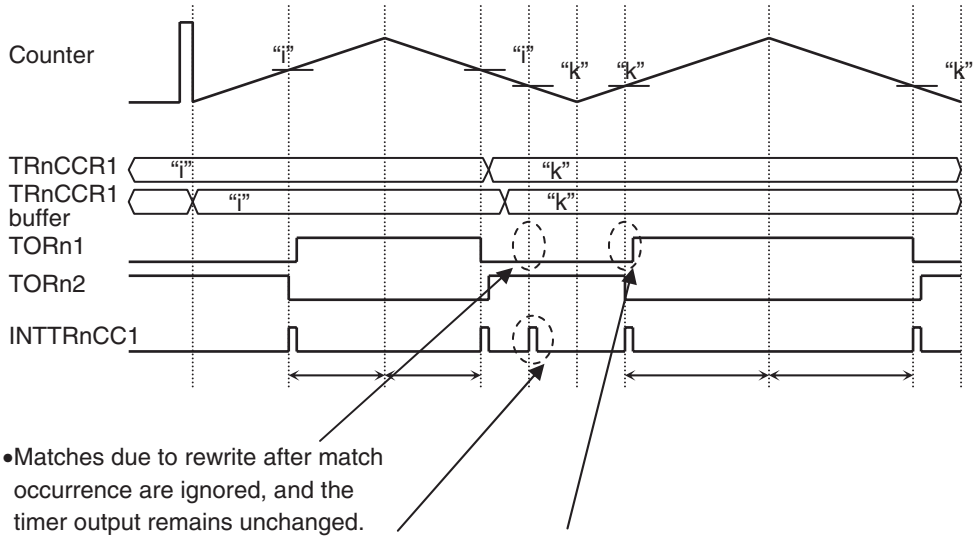


If a value larger than the counter value is written before match occurrence, no match occurs, so the following output wave results.



If no match occurs, the timer output remains unchanged.
However, even if a match occurs, the timer output is forcibly changed to normal phase inactive level at valleys.

Rewrite in <4> interval (rewrite after match occurrence)
In the case of rewrite after a match between the TRnCCR1 to TRnCCR3 registers and the counter occurs, further match occurrences are ignored, so the rewrite value is not reflected.



- Matches due to rewrite after match occurrence are ignored, and the timer output remains unchanged.
- The INTTRnCC1 signal becomes valid from the next match after up/down count is switched and the timer output changes. However, a match interrupt is output upon INTTRnCC1 interrupt output.

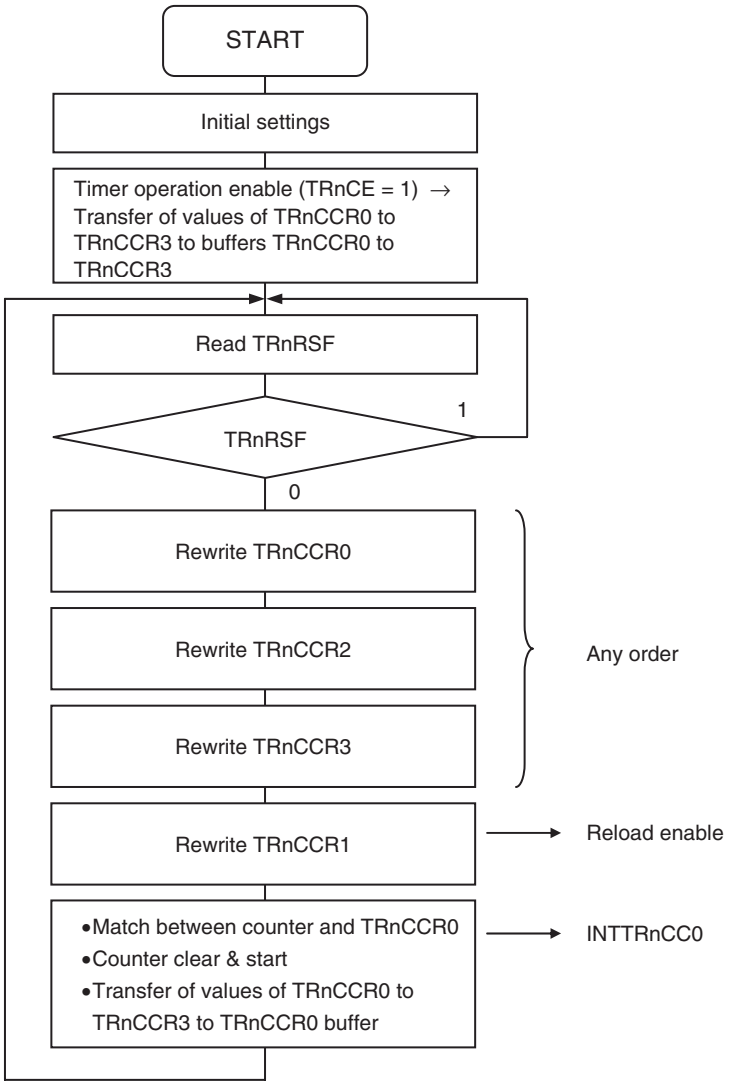
(c) Cautions related to rewriting TRnOPT1

Since the internal interrupt thinning out counter is cleared when the TRnOPT1 register is written to, the interrupt output interval may temporarily become longer.

(2) Batch rewrite (reload mode)

Batch rewrite is set by setting TRnOPT0 register bit TRnCMS = 0, TRnOPT1 register bit TRnRDE = 0, TRnICE = 1 (reload enabled at peaks), and TRnIOE = 1 (reload enabled at valleys). In this mode, the values written to the various compare registers are all transferred at the same time to the respective buffer registers at the reload timing.

Figure 10-26: Basic Operation Flow during Batch Rewrite



Caution: Write access to the TRnCCR1 register includes also the reload enable operation. Therefore, rewrite the TRnCCR1 register after rewriting the other TRnCCR registers.

Remarks: 1. This sample flow chart is for the PWM mode.

2. n = 0, 1

Figure 10-27: Batch Rewrite Timing (1/2)

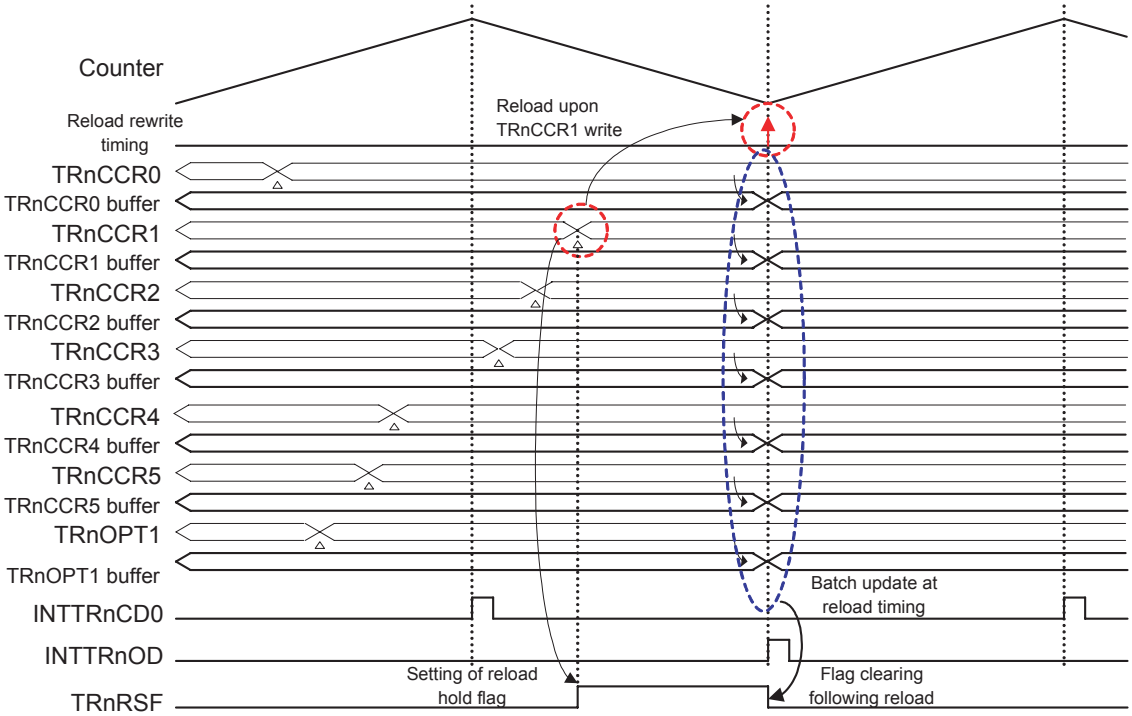
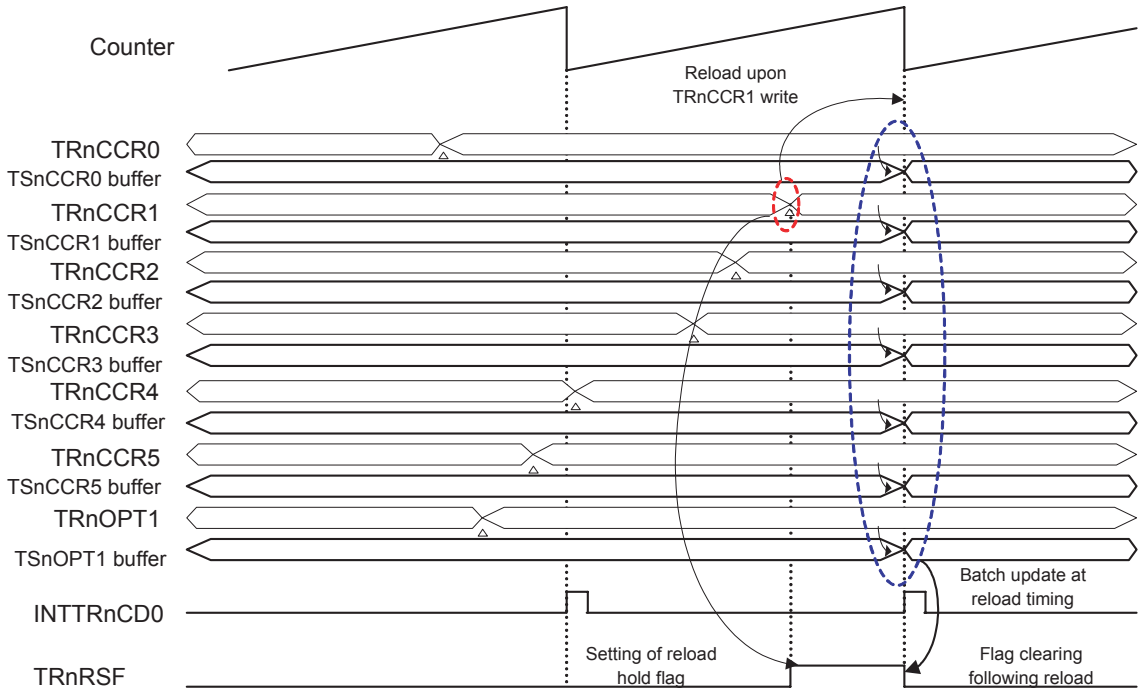
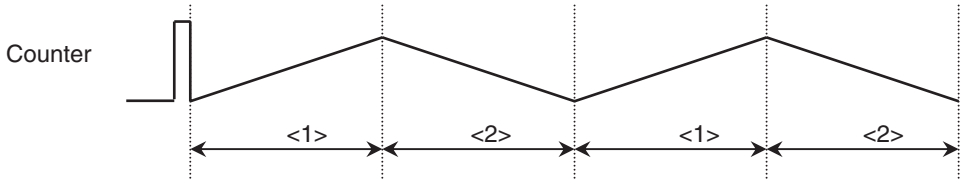


Figure 10-27: Batch Rewrite Timing (2/2)

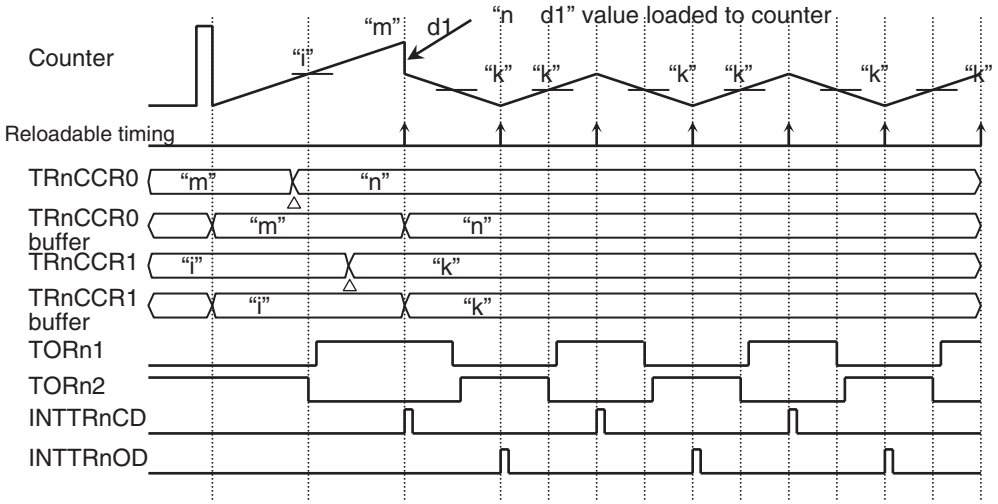


(a) TRnCCR0 register rewrite operation in high-accuracy T-PWM mode

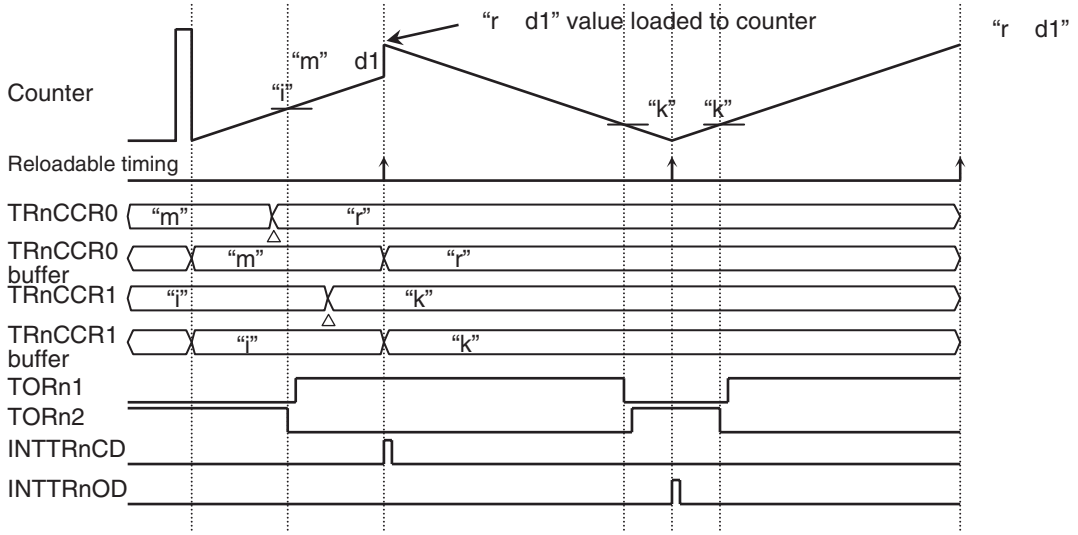
When rewriting the TRnCCR0 register in the batch rewrite mode, the output waveform changes according to whether reload occurs at a peak or at a valley (TRnICE = 1, TRnIOE = 1 settings).



Rewrite in <1> interval (rewrite during up count)
Since the next reload timing becomes the peak point, the cycle on the down count side changes and an asymmetrical triangular waveform is output. Also, since the cycle changes, reset the duty value as necessary.



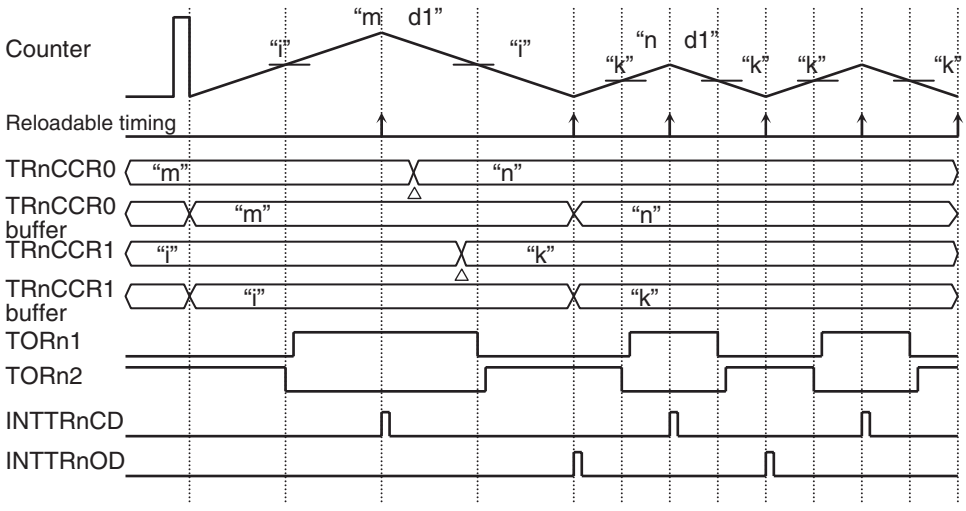
Remark: d1: TRnDTC1 setting value



Remark: d1: TRnDTC1 setting value

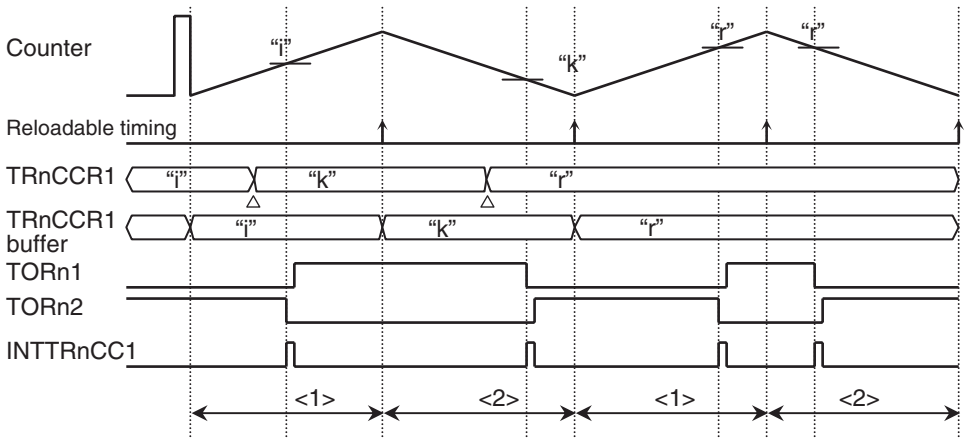
The counter loads the TRnCCR0 value minus "d1" upon occurrence of reload in the high-accuracy T-PWM mode. As a result, the expected waveform can be output even if the cycle value is changed at the peak reload timing.

Rewrite in <2> interval (rewrite during down count)
Since the next reload timing becomes the valley point, the cycle value changes from the next cycle and the asymmetrical triangular waveform output is held. Since the cycle changes, be sure to set again the duty value as required.



Remark: d1: TRnDTC1 setting value

(b) TRnCCR1 to TRnCCR3 register rewrite operation in high-accuracy T-PWM mode



Remark: When TRnDTC0 = 0, TRnDTC1 = 0

Rewrite in <1> interval (rewrite during up count)
Since reload is performed at the peak interrupt timing, an asymmetric triangular waveform is output.

Rewrite in <2> interval (rewrite during down count)
Since reload is performed at the valley interrupt timing, an asymmetric triangular waveform is output.

10.4.3 List of outputs in each mode

(1) Timer outputs in each mode

The timer outputs (pins TORn0 to TORn7) in each mode are listed below.

Table 10-3: List of Timer Outputs in Each Mode (1/2)

Operation Mode	TORn0	TORn1	TORn2	TORn3
Interval mode	Toggle output upon TRnCCR0 compare match	Toggle output upon TRnCCR1 compare match	Toggle output upon TRnCCR2 compare match	Toggle output upon TRnCCR3 compare match
External event count mode	Toggle output upon TRnCCR0 compare match	Toggle output upon TRnCCR1 compare match	Toggle output upon TRnCCR2 compare match	Toggle output upon TRnCCR3 compare match
External trigger pulse output mode	Toggle output upon CCR0 compare match or external trigger input	External trigger pulse waveform output	External trigger pulse waveform output	External trigger pulse waveform output
One-shot pulse mode	Active at count start. Inactive upon TRnCCR0 match.	Active upon TRnCCR1 match. Inactive upon TRnCCR0 match.	Active upon TRnCCR2 match. Inactive upon TRnCCR0 match.	Active upon TRnCCR3 match. Inactive upon TRnCCR0 match.
PWM mode	Toggle output upon TRnCCR0 compare match	PWM output upon TRnCCR1 compare match	PWM output upon TRnCCR2 compare match	PWM output upon TRnCCR3 compare match
Free-running mode	Toggle output upon TRnCCR0 compare match	Toggle output upon TRnCCR1 compare match	Toggle output upon TRnCCR2 compare match	Toggle output upon TRnCCR2 compare match
Pulse width measurement mode	-	-	-	-
Triangular wave PWM mode	Inactive during up count. Active during down count.	PWM output upon TRnCCR1 compare match	PWM output upon TRnCCR2 compare match	PWM output upon TRnCCR3 compare match
High-accuracy T-PWM mode	Inactive during counter or sub-counter up count. Active during down count.	PWM output (with dead time) upon TRnCCR1 compare match	Inverted phase output to TORn1	PWM output (with dead time) upon TRnCCR2 compare match
PWM mode with dead time	Toggle output upon TRnCCR0 compare match	PWM output (with dead time) upon TRnCCR1 compare match	Inverted phase output to TORn1	PWM output (with dead time) upon TRnCCR2 compare match

Table 10-3: List of Timer Outputs in Each Mode (2/2)

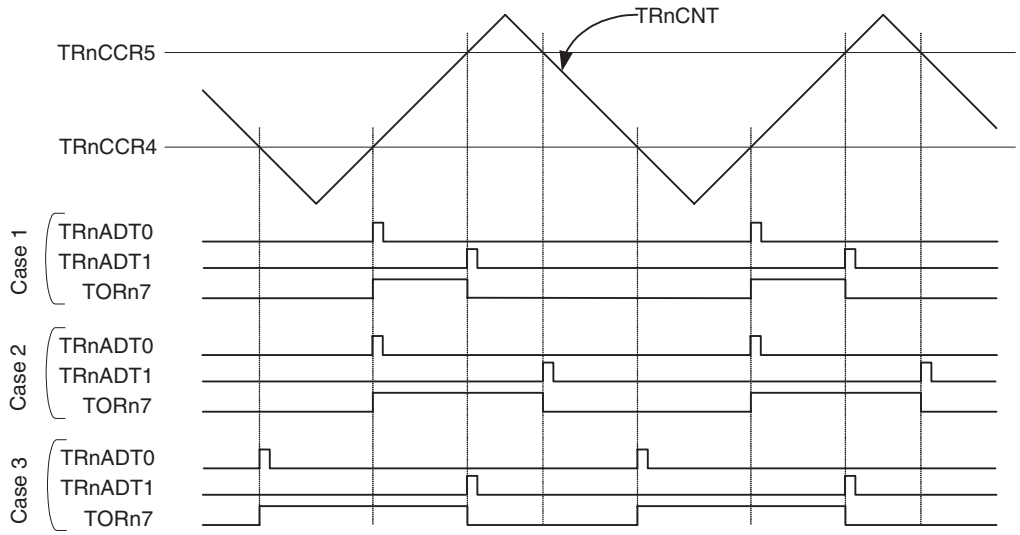
Operation Mode	TORn4	TORn5	TORn6	TORn7
Interval mode	Toggle output upon TRnCCR4 compare match	Toggle output upon TRnCCR5 compare match	-	-
External event count mode	Toggle output upon TRnCCR4 compare match	Toggle output upon TRnCCR5 compare match	-	-
External trigger pulse output mode	External trigger pulse waveform output	External trigger pulse waveform output	-	-
One-shot pulse mode	High upon TRnCCR4 match. Inactive upon TRnCCR0 match.	High upon TRnCCR5 match. Inactive upon TRnCCR0 match.	-	-
PWM mode	PWM output upon TRnCCR4 compare match	PWM output upon TRnCCR5 compare match	-	Pulse output upon A/D conversion trigger **Note**
Free-running mode	Toggle output upon TRnCCR4 compare match	Toggle output upon TRnCCR5 compare match	-	-
Pulse width measurement mode	-	-	-	-
Triangular wave PWM mode	PWM output upon TRnCCR4 compare match	PWM output upon TRnCCR5 compare match	-	Pulse output upon A/D conversion trigger **Note**
High-accuracy T-PWM mode	Inverted phase output to TORn3	PWM output (with dead time) upon TRnCCR3 compare match	Inverted phase output to TORn5	Pulse output upon A/D conversion trigger **Note**
PWM mode with dead time	Inverted phase output to TORn3	PWM output (with dead time) upon TRnCCR3 compare match	Inverted phase output to TORn5	Pulse output upon A/D conversion trigger Note

Note: For details on TORn7, refer to **10.4.3 (a) TORn7 pin output control**.

(a) TORn7 pin output control

The A/D conversion signals can be output to pin TORn7. Pin TORn7 is set (to 1) by the TRnADTRG0 signal trigger, and it is reset (to 0) by the TRnADTRG1 signal trigger. If the TRnADTRG0 trigger occurs while pin TORn7 is set (to 1), its set (1) status is maintained. If the TRnADTRG1 trigger occurs while pin TORn7 is reset (0), the (0) status is maintained. If the TRnADTRG0 and TRnADTRG1 signal triggers occur simultaneously, pin TORn7 is reset (to 0).

Figure 10-28: TORn7 Pin Output Timing 1



Remark: Case 1: When TRnCCR4 < TRnCCR5, TRnOPT2 = 04H, TRnOPT3 = 10H
Case 2: When TRnCCR4 < TRnCCR5, TRnOPT2 = 04H, TRnOPT3 = 20H
Case 3: When TRnCCR4 < TRnCCR5, TRnOPT2 = 08H, TRnOPT3 = 10H

(2) Interrupts in each mode

The interrupts in each mode (INTTRnCC0 to INTTRnCC5, INTTRnOV, INTTRnER) are listed below.

Table 10-4: List of Interrupts in Each Mode (1/2)

Operation Mode	INTTRnCC0	INTTRnCC1	INTTRnCC2	INTTRnCC3
Interval mode	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt	TRnCCR2 compare match interrupt	TRnCCR3 compare match interrupt
External event count mode	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt	TRnCCR2 compare match interrupt	TRnCCR3 compare match interrupt
External trigger pulse output mode	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt	TRnCCR2 compare match interrupt	TRnCCR3 compare match interrupt
One-shot pulse mode	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt	TRnCCR2 compare match interrupt	TRnCCR3 compare match interrupt
PWM mode	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt	TRnCCR2 compare match interrupt	TRnCCR3 compare match interrupt
Free-running mode	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt	TRnCCR2 compare match interrupt	TRnCCR3 compare match interrupt
Pulse width measurement mode	TIR10 capture interrupt	TIR11 capture interrupt	TIR12 capture interrupt	TIR13 capture interrupt
Triangular wave PWM mode	TIR10 capture interrupt	TIR11 capture interrupt	TIR12 capture interrupt	TIR13 capture interrupt
High-accuracy T-PWM mode	TRnCCR0 compare match interrupt **Note 1**	TRnCCR1 compare match interrupt **Note 2**	TRnCCR2 compare match interrupt **Note 2**	TRnCCR3 compare match interrupt **Note 2**
PWM mode with dead time	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt **Note 3**	TRnCCR2 compare match interrupt **Note 3**	TRnCCR3 compare match interrupt **Note 3**

Notes: 1. A compare match interrupt is output when the TRnDTC1 register is set to 000H. INTTRnCD can be used as the peak interrupt.

2. If set in the range of 0000H ≤TRnCCRm < TRnDTC0, (TRnCCR0 - TRnDTC1)< TRnCCRm ≤TRnCCR0 (m = 1 to 5), no compare match interrupt is output.

3. If set to TRnCCR0 < TRnCCRm (m to 1 to 5), no compare match interrupt is output.

Remarks: 1. "-" in the table indicates inactive level output.

2. n = 0, 1

Table 10-4: List of Interrupts in Each Mode (2/2)

Operation Mode	INTTRnCC4	INTTRnCC5	INTTRnOV	INTTRnER
Interval mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	-	-
External event count mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	-	-
External trigger pulse output mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	-	-
One-shot pulse mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	-	-
PWM mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	-	Error interrupt
Free-running mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	Overflow interrupt	-
Pulse width measurement mode	-	-	Overflow interrupt	-
Triangular wave PWM mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	-	Error interrupt
High-accuracy T-PWM mode	TRnCCR4 compare match interrupt **Note 1**	TRnCCR5 compare match interrupt **Note 1**	Overflow interrupt **Note 3**	Error interrupt
PWM mode with dead time	TRnCCR4 compare match interrupt **Note 2**	TRnCCR5 compare match interrupt **Note 2**	-	Error interrupt

Notes: 1. If set in the range of 0000H ≤TRnCCRm < TRnDTC0, (TRnCCR0 - TRnDTC1) < TRnCCRm ≤TRnCCR0 (m = 1 to 5), no compare match interrupt is output.

2. If set to TRnCCR0 < TRnCCRm (m = 1 to 5), no compare match interrupt is output.

3. If a setting error has been made for TRnCCR0, TRnDTC0, TRnDTC1, an overflow interrupt (INTTRnOV) is output.

Remarks: 1. "-" in the table indicates inactive level output.

2. n = 0, 1

(3) A/D conversion triggers, peak interrupts, and valley interrupts in each mode

The A/D conversion triggers, peak interrupts, and valley interrupts in each mode are listed below.

Table 10-5: List of A/D Conversion Triggers, Peak Interrupts and Valley Interrupts in Each Mode

Operation Mode	TRnADTRG0	TRnADTRG1	INTTRnCD	INTTRnOD
Interval mode	-	-	-	-
External event count mode	-	-	-	-
External trigger pulse output mode	-	-	-	-
One-shot pulse mode	-	-	-	-
PWM mode	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Peak interrupt at same timing as INTTRnCC0 interrupt	-
Free-running mode	-	-	-	-
Pulse width measurement mode	-	-	-	-
Triangular wave PWM mode	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	-	Valley interrupt at counter valley (upon switching from down to up count)
High-accuracy T-PWM mode	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Peak interrupt	Valley interrupt at counter valley (upon switching from down to up count)
PWM mode with dead time	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Peak interrupt at same timing as INTTRnCC0 interrupt	-

Remarks: 1. The INTTRnCD interrupt and INTTRnOD interrupt are the occurrence conditions following interrupt thinning out.

2. n = 0, 1

10.5 Match Interrupts

Match interrupts consist of compare match interrupts (INTTRnCC0 to INTTRnCC5), peak interrupts (INTTRnCD), and valley interrupts (INTTRnOD). For details about error interrupts, refer to **10.9 Error Interrupts**.

Compare match interrupts (INTTRnCC0 to INTTRnCC5) are interrupts that occur following a match between the TRnCCR0 to TRnCCR5 registers and the counter, and are output in all modes (no operation mode restrictions).

Peak interrupts (INTTRnCD) are output in the PWM mode, triangular wave PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. If the counter is a triangular wave operation mode (triangular wave PWM mode, high-accuracy PWM mode), a peak interrupt is output when the counter switches from up count to down count. If the counter is in a saw tooth wave operation mode (PWM mode, PWM mode with dead time), a peak interrupt occurs upon a match between the counter and the TRnCCR0 register (same timing as INTTRnCC0 interrupt).

Valley interrupts occur when the counter switches from down count to up count in the triangular wave PWM mode and high-accuracy T-PWM mode.

Figure 10-29: Interrupt Signal Output Example (1/2)

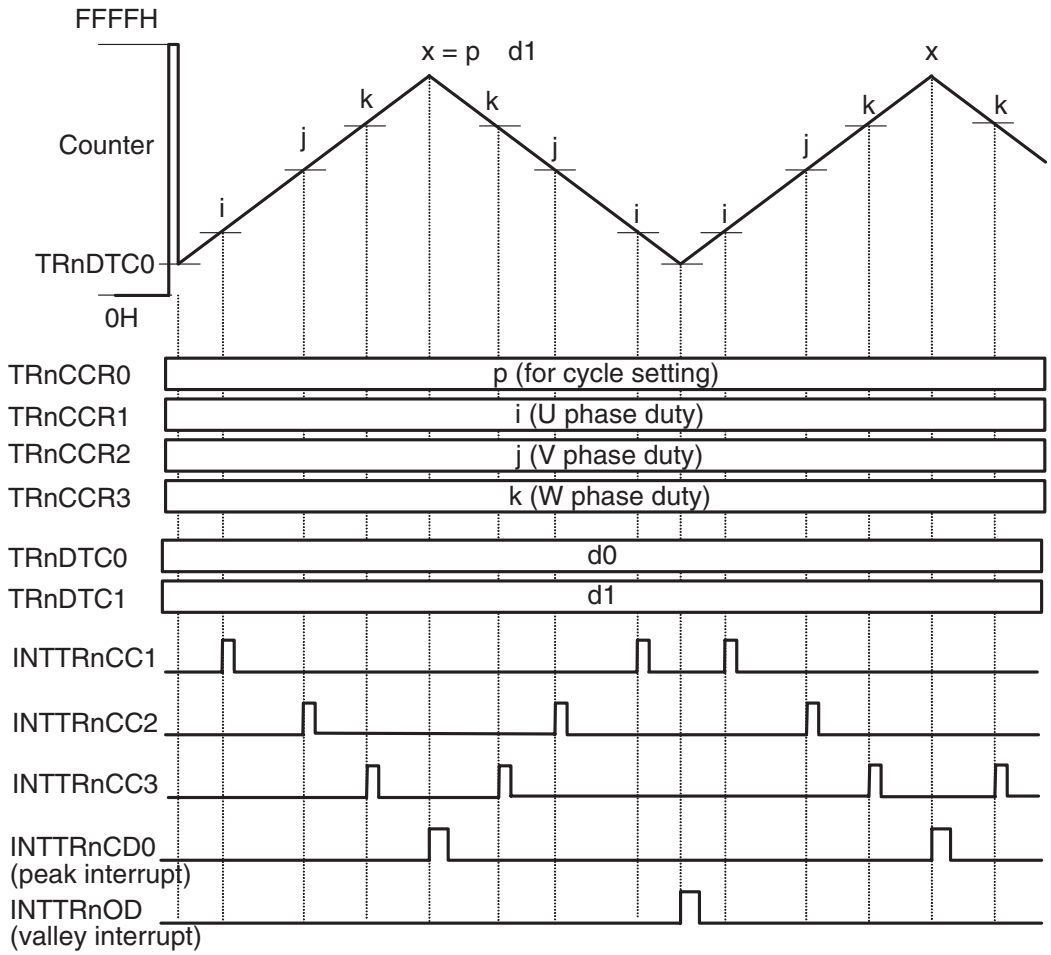
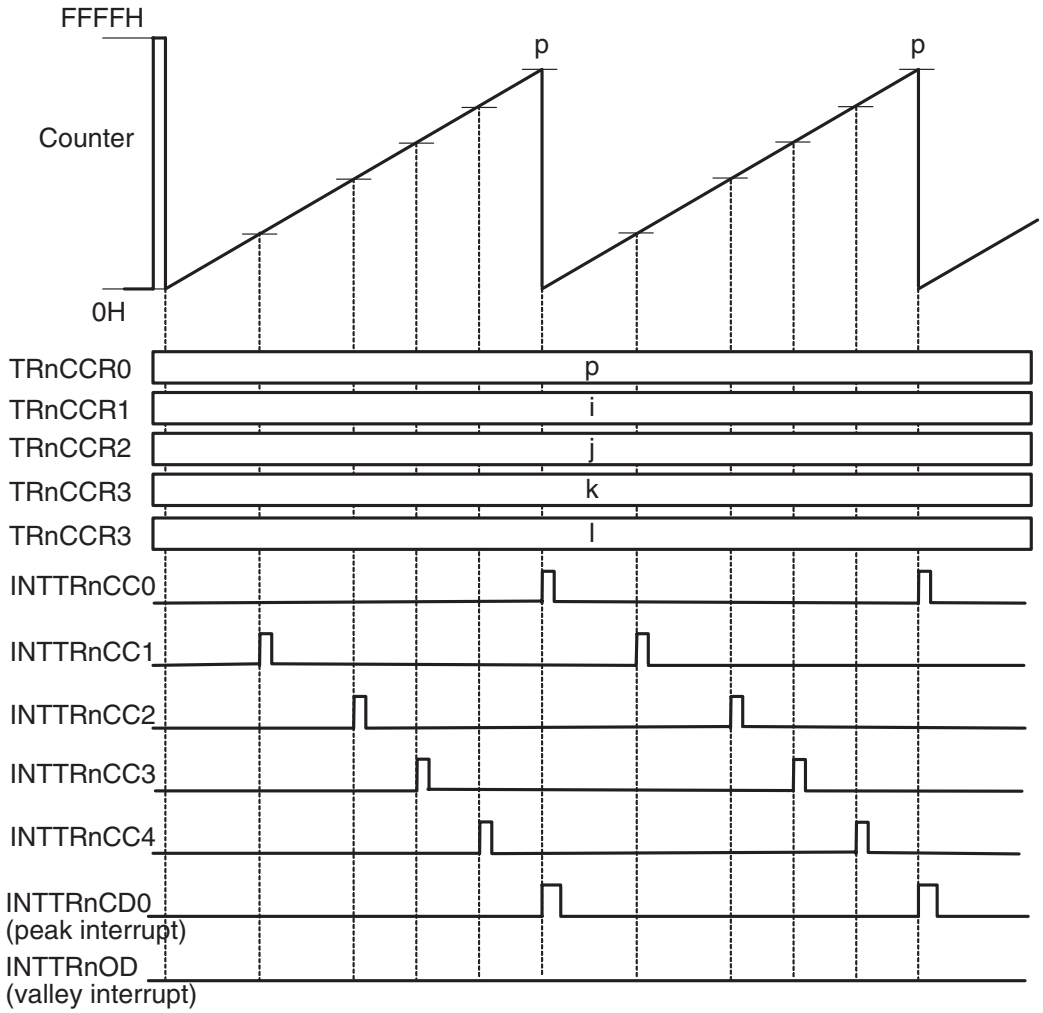


Figure 10-29: Interrupt Signal Output Example (2/2)



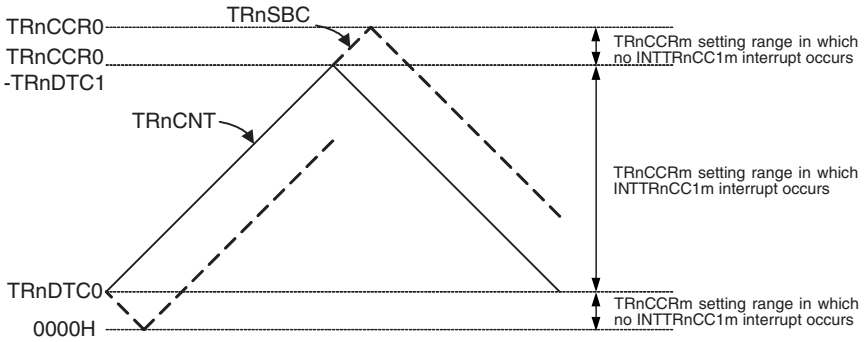
10.5.1 Compare match interrupt related cautions

(1) Cautions in high-accuracy T-PWM mode

Compare match interrupts occur upon a match between the counter and a compare register (TRnCCR0 to TRnCCR5). However, in the high-accuracy T-PWM mode, the compare register can be set exceeding the counter's count operation range. Therefore, under the following conditions, no compare interrupt is output.

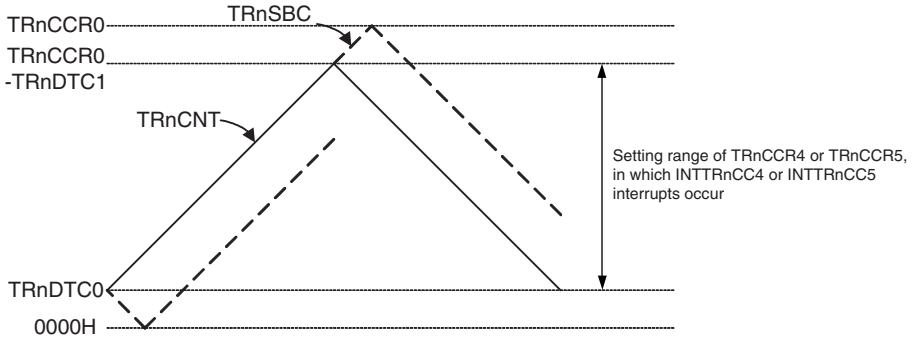
- Restrictions related to compare match interrupt with TRnCCR0 register (INTTRnCC0)
  In the high-accuracy T-PWM mode, when TRnDTC1 ≠ 000H, no compare match interrupt (INTTRnCC0) is output.
  (Use INTTRnOD (valley interrupt) and INTTRnCD (peak interrupt) as the cycle interrupts.)

- Restrictions related to TRnCCR1 to TRnCCR3 register
  In the high-accuracy T-PWM mode, if set in the range of 0000H ≤TRnCCRm < TRnDTC0, (TRnCCR0 - TRnDTC1) < TRnCCRm ≤TRnCCR0, no interrupt occurs upon a match between the compare value and the counter.



Remark: m = 1 to 3

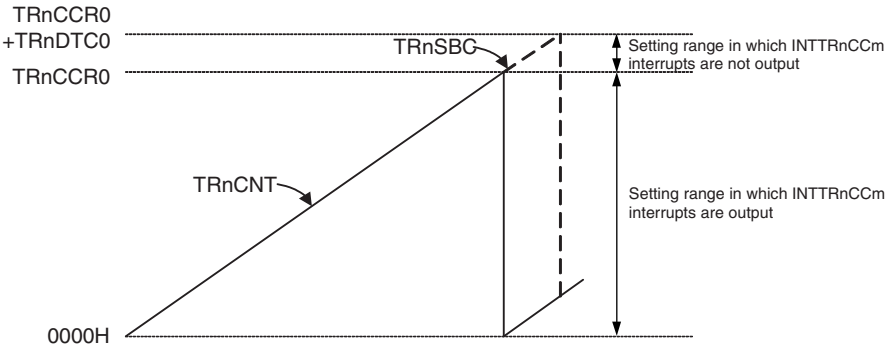
- Restrictions related to TRnCCR4 and TRnCCR5 registers
  In the high-accuracy T-PWM mode, if set in the range of 0000H ≤TRnCCR4, TRnCCR5 < TRnDTC0, (TRnCCR0 - TRnDTC1) < TRnCCR4, TRnCCR5 ≤TRnCCR0, no compare match interrupt is output since no match between the compare value and counter occurs.
  When TRnCCR4 and TRnCCR5 registers are used as trigger causes for A/D triggers, perform setting in the range of TRnDTC0 ≤TRnCCR4, TRnCCR5 ≤(TRnCCR0 - TRnDTC1).



(2) Cautions in PWM mode with dead time

Compare match interrupts are output upon a match between the counter and compare registers (TRnCCR0 to TRnCCR5). However, in the high-accuracy T-PWM mode, the compare register can be set exceeding the counter's count operation range. Therefore, under the following conditions, no compare interrupt is output.

- Restrictions related to TRnCCRm
  In the PWM mode with dead time, if setting is performed in the following range, no match between the compare value and counter occurs, and no compare match interrupt is output:
  When TRnCCR0 < TRnCCRm ≤(TRnCCR0 + TRnDTC0),
  TRnCCR4, TRnCCR5 registers are used as trigger causes for A/D triggers, perform settings with TRnCCR4, TRnCCR5 ≤TRnCCR0.



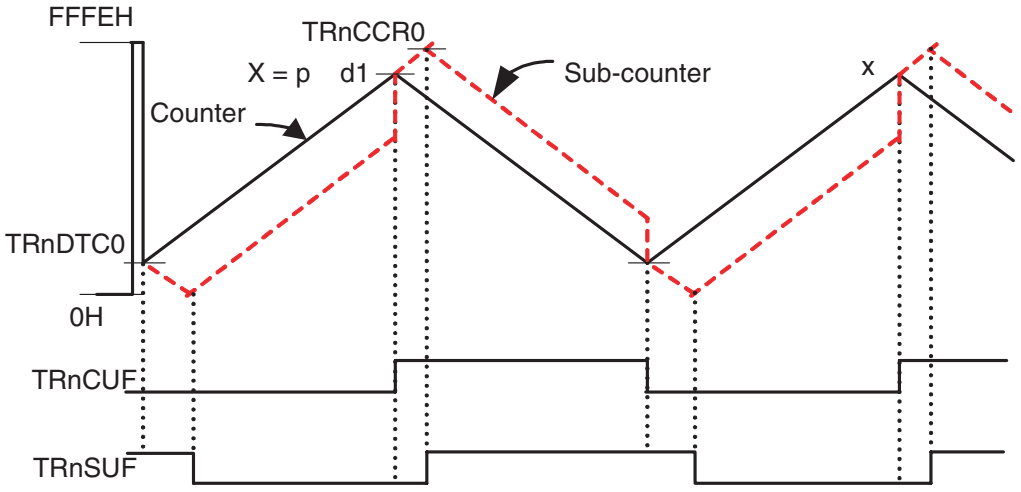
Remark: m = 1 to 5

10.6 Flags

10.6.1 Up count flags

Timer Rn has two counters, a counter and a sub-counter.
TRnCUF is the counter's up/down status flag. It operates in the triangular wave PWM mode and high-accuracy T-PWM mode, and is fixed to 0 in all other modes.
TRnSUF is the sub-counter's up/down status flag. It operates in the high-accuracy T-PWM mode, and is fixed to 0 in all other modes.
For both TRnCUF and TRnSUF, 0 indicates the up count status, and 1 indicates the down count status.

Figure 10-30: Up Count Flags Timings (1/2)



In the triangular wave PWM mode, the values of TRnCUF are as follows.

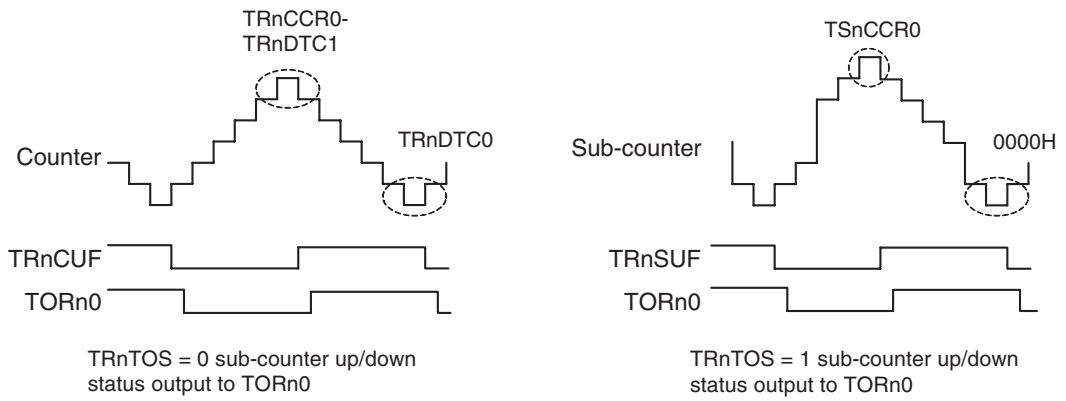
0 ≤counter < TRnCCR0+1 … 0 (up count)
TRnCCR0+1 ≥ counter > 0 … 1 (down count)

In the high-accuracy T-PWM mode, the values of TRnCUF/TRnSUF are as follows.

[TRnCUF]
TRnDTC0 ≤counter < (TRnCCR0 - TRnDTC1) ..0 (up count)
TRnCCR0-TRnDTC1 ≥ counter > TRnDTC0 … 1 (down count)

[TRnSUF]
0 ≤sub-counter < TRnCCR0 ..0 (up count)
TRnCCR0 ≥ sub-counter > 0 ..1 (down count)

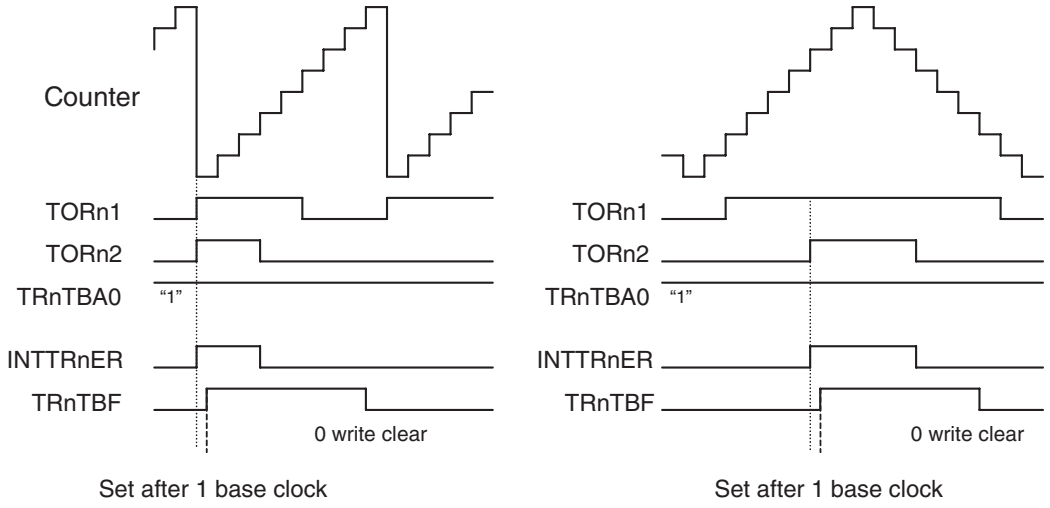
Figure 10-30: Up Count Flags Timings (2/2)



10.6.2 Normal phase/inverted phase simultaneous active detection flag

Timer Rn has a flag (TRnTBF) that detects normal phase/inverted phase simultaneous active states. The TRnTBF flag is valid in the PWM mode, triangular wave PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time.

Figure 10-31: Normal Phase/Inverted Phase Simultaneous Active Detection Flag Timing



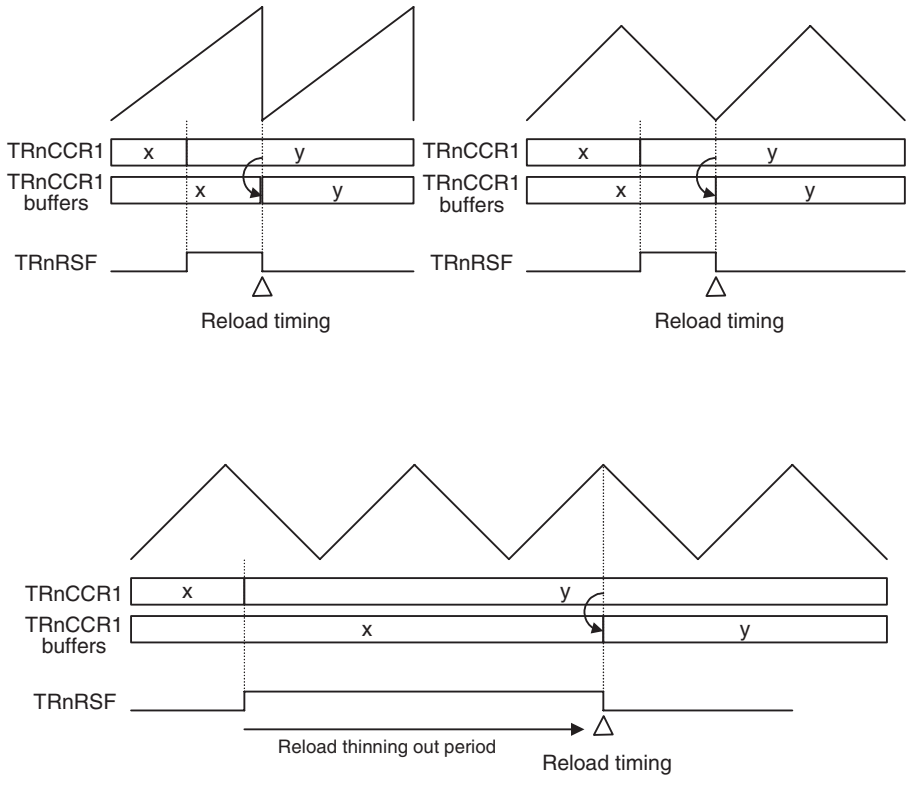
10.6.3 Reload hold flag

In the case of timer Rn, the reload hold flag (TRnRSF) is set to “1” upon occurrence of a reload request (when the TRnCCR1 register is written to). When reload occurs and the values are transferred to all the buffer registers, the reload hold flag is cleared to “0”. The TRnRSF flag is valid in the following operation modes.

- External trigger pulse output mode
- PWM mode
- Triangular wave PWM mode
- High-accuracy T-PWM mode (TRnCMS = 0)
- PWM mode with dead time

Caution: The TRnRSF flag is set to “1” following the lapse of 4 base clocks after TRnCCR1 register write completion.

Figure 10-32: Reload Hold Flag Timings



10.7 Interrupt Thinning Out Function

The operations related to the interrupt thinning out function are indicated below.

- The interrupts subject to thinning out are INTTRnCD (peak interrupt) and INTTRnOD (valley interrupt).
- TRnOPT1 register bit TRnICE is used to enable INTTRnCD interrupt output and to specify thinning out count targets.
- TRnOPT1 register bit TRnIOE is used to enable INTTRnOD interrupt output and to specify thinning out count targets.
- TRnOPT2 register bit TRnRDE is used to specify reload thinning Yes/No.
- If thinning out Yes is specified, reload is executed at the same timing as interrupt output following thinning out.
- If thinning out No is specified, reload is executed at the reload timing after write access to the TRnCCR1 register.
- The reload/anytime rewrite method can be specified with TRnOPT0 register bit TRnCMS.
- When TRnCMS = 0, the register value is updated in synchronization with reload, but when TRnCMS = 1, the register value is updated immediately after write access.

Caution: When write access is performed to the TRnOPT1 register, the internal thinning out counter is cleared when the register value is updated. Therefore, the interrupt interval may temporarily become longer than expected.
To prevent this, it is recommended to set TRnCSM = 0 and TRnRDE = 1, and to change the interrupt thinning out count with the reloaded setting according to interrupt thinning out. Using this method, the interrupt interval is kept the same as the setting value.

10.7.1 Operation of interrupt thinning out function

Figure 10-33: Interrupt Thinning Out Operations (1/2)

(a) when TRnICE = 1, TRnIOE = 1 (peak/valley interrupt output)

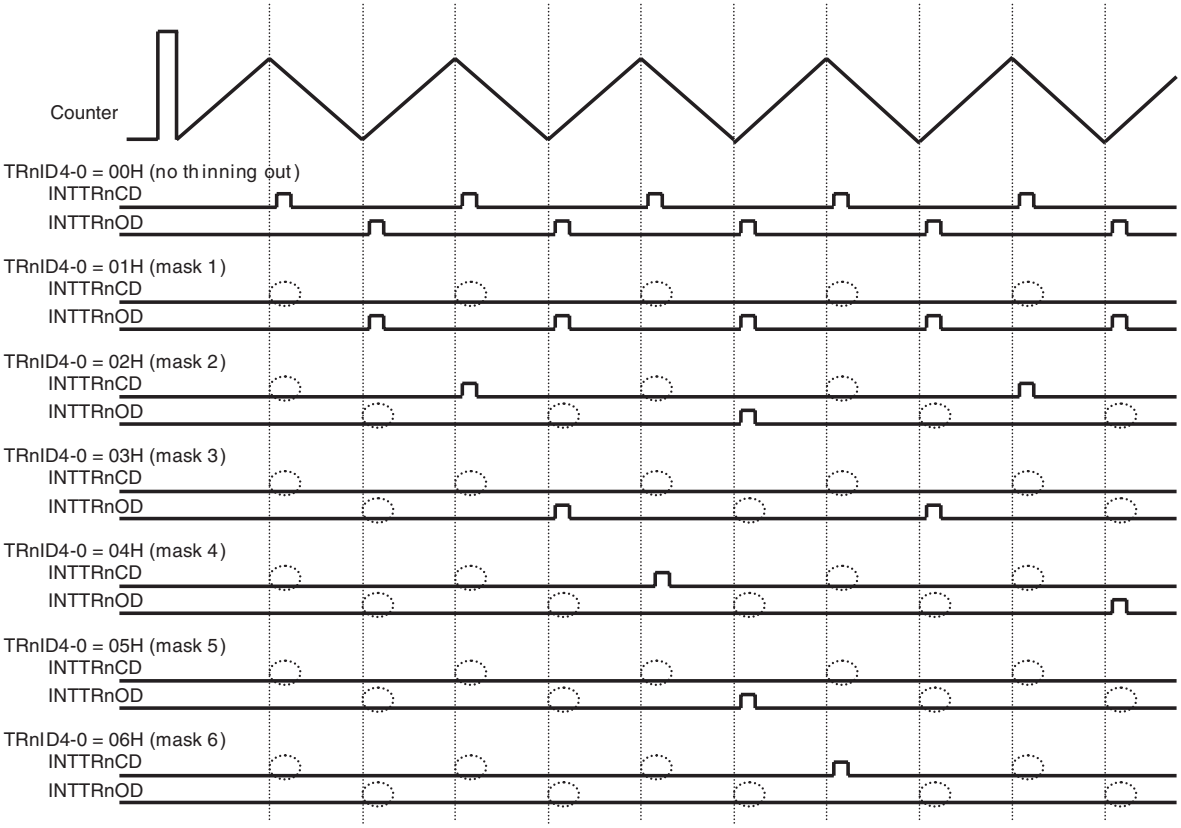
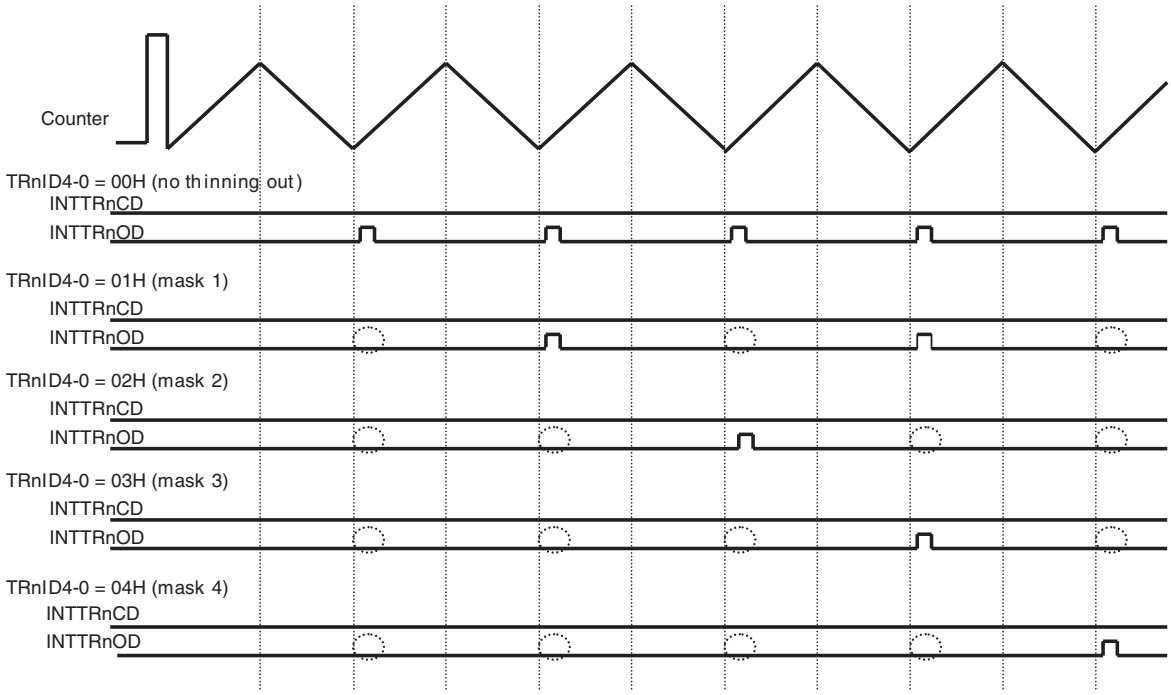
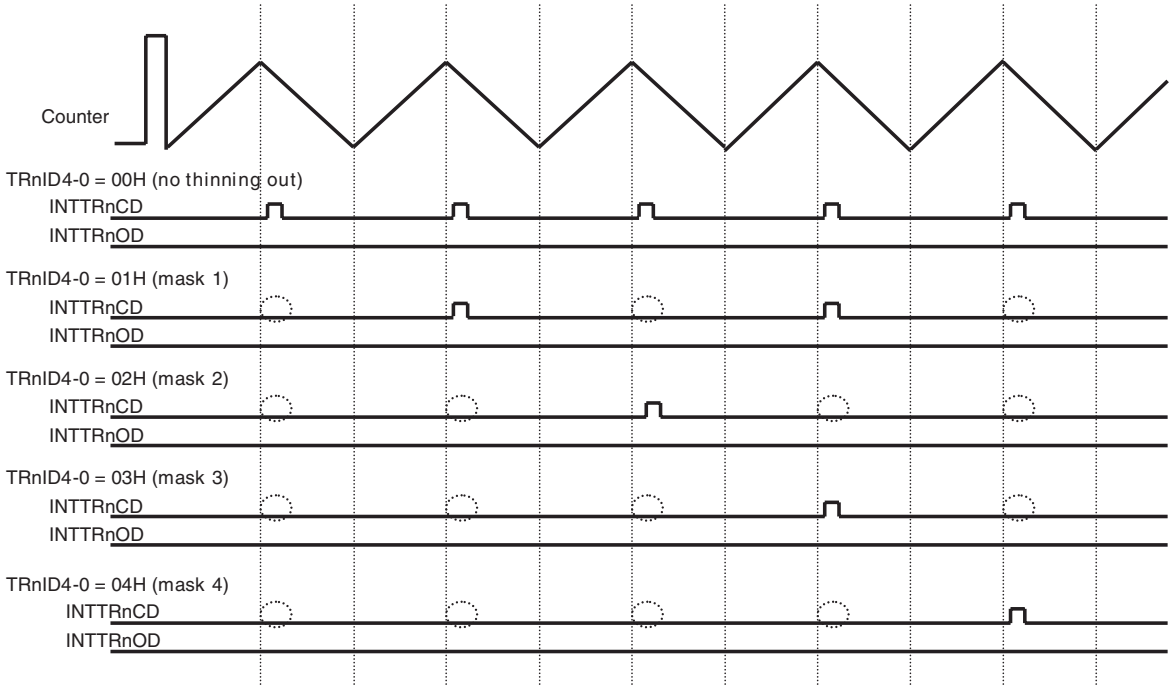


Figure 10-33: Interrupt Thinning Out Operations (2/2)

(b) when TRnICE = 1, TRnIOE = 0 (peak interrupt only output)



(c) when TRnICE = 0, TRnIOE = 1 (valley interrupt only output)



10.7.2 Operation examples when peak interrupts and valley interrupts occur alternately

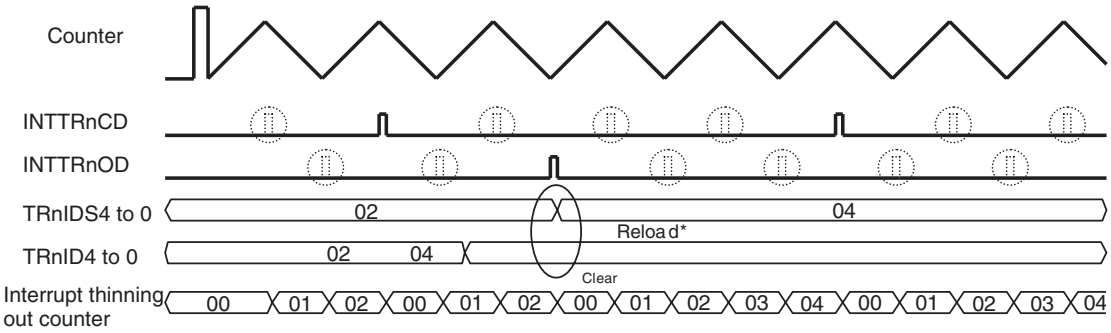
(1) Register settings

Set both TRnOPT1 register bit TRnICE and TRnOPT1 register bit TRnIOE to 1.

(2) Operation example

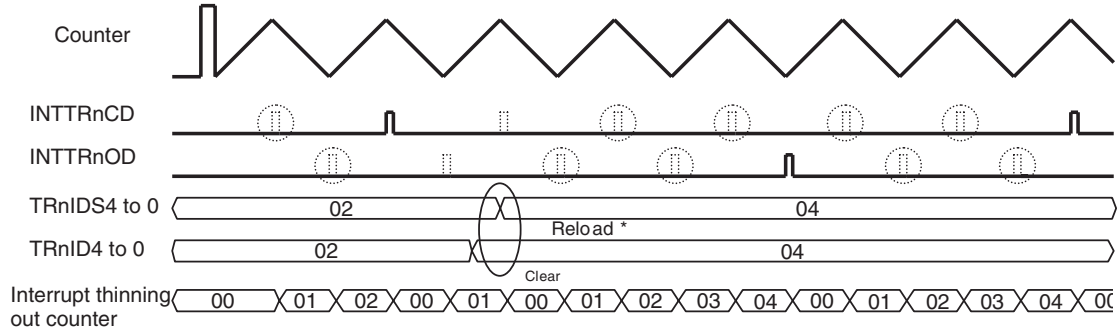
Figure 10-34: Examples when Peak Interrupts and Valley Interrupts Occur Alternately (1/2)

(a) when TRnCMS = 0, TRnRDE = 1 (Reload Thinning Out Control) (Recommended Settings)



* Reload is executed at the thinned out interrupt output timing. All other reload timings are ignored.

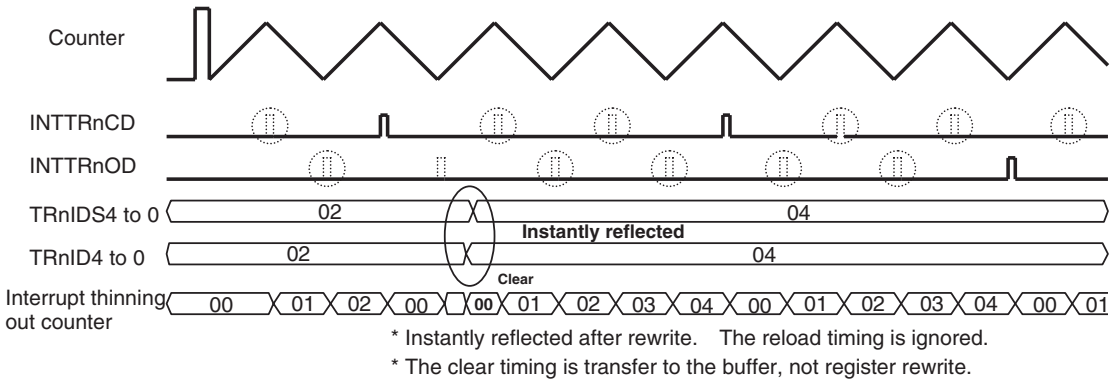
(b) when TRnCMS = 0, TRnRDE = 0 (No Reload Control)



* Reload is executed at the reload timing after rewrite.

Figure 10-34: Examples when Peak Interrupts and Valley Interrupts Occur Alternately (2/2)

(c) when TRnCMS = 1, TRnRDE = x (anytime rewrite)



* Instantly reflected after rewrite. The reload timing is ignored.

* The clear timing is transfer to the buffer, not register rewrite.

10.7.3 Interrupt thinning out function during counter saw tooth wave operation

The operations related to the interrupt thinning out function during counter saw tooth wave operation (PWM mode, PWM mode with dead time) are indicated below.

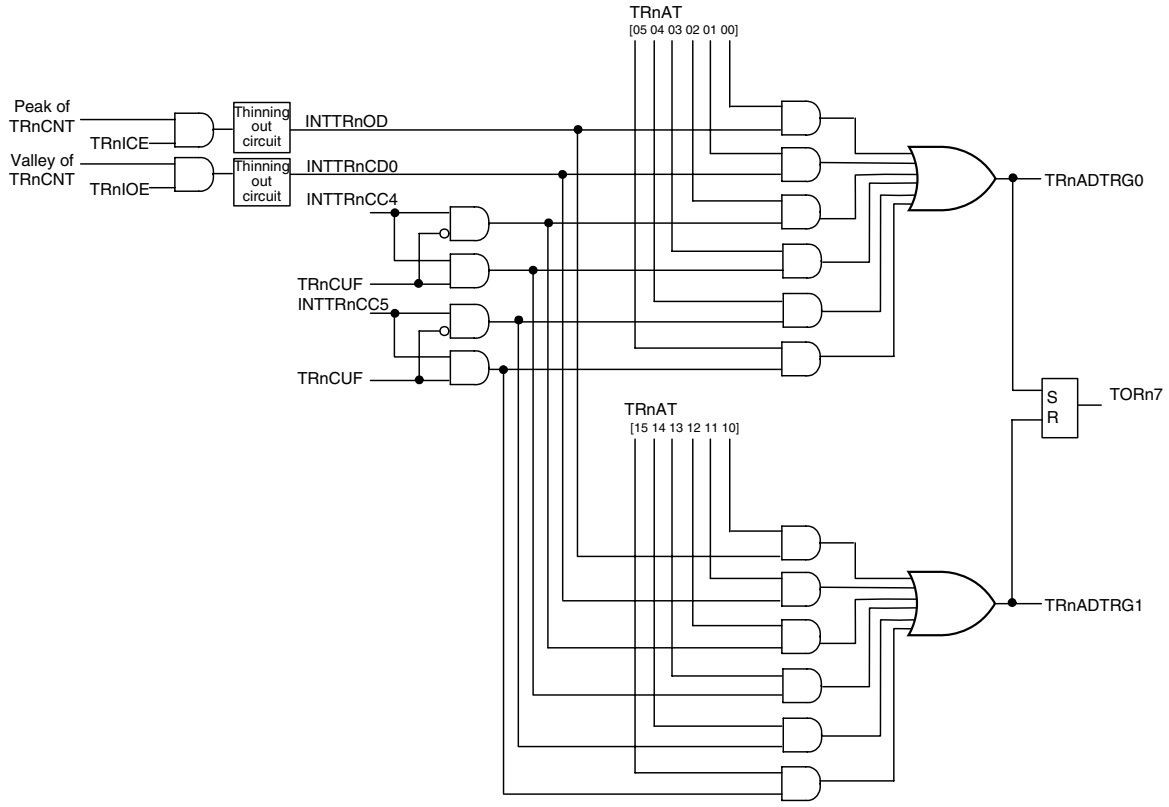
- The interrupt subject to thinning out is INTTRnCD (peak interrupt). The saw tooth wave operation occurs upon a match between the TRnCCR0 register and counter occurs.
- TRnOPT1 register bit TRnICE is used to enable INTTRnCD interrupt output and to specify thinning out count targets.
- The TRnOPT1 register bit TRnIOE setting is invalid. INTTRnOD interrupt output is prohibited.
- TRnOPT1 register bit TRnRDE is used to specify reload thinning out Yes/No.
- If thinning out Yes is specified, reload is executed at the same timing as interrupt output following thinning out.
- If thinning out No is specified, reload is executed at the reload timing after write access to the TRnCCR1 register.

Caution: When write access is performed to the TRnOPT1 register, the internal thinning out counter is cleared when the register value is updated. Therefore, the interrupt interval may temporarily become longer than expected.
To prevent this, it is recommended to set TRnCSM = 0 and TRnRDE = 1, and to change the interrupt thinning out count with the reloaded setting according to interrupt thinning out. Using this method, the interrupt interval is kept the same as the setting value.

10.8 A/D Conversion Trigger Function

This section describes the operation of the A/D conversion triggers output in the PWM mode, triangular wave PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. In these modes, the TRnCCR4 and TRnCCR5 registers are used as match interrupts and for the A/D conversion trigger function, with no influence on timer outputs in terms of the compare operation. For the A/D conversion triggers that can be output in each mode, refer to **10.4.3 (3) A/D conversion triggers, peak interrupts, and valley interrupts in each mode**.

Figure 10-35: A/D Conversion Trigger Output Controller



The above figure shows the A/D conversion trigger controller. As shown in this figure, it is possible to select and perform OR output of compare match interrupts (INTTRnCC5, INTTRnCC4) and peak interrupts (INTTRnCD), valley interrupts (INTTRnOD) interrupt signals, sub-counter peak timing, and sub-counter valley timing.
In the case of timer R, there are two identical A/D conversion trigger controllers, and each one can be controlled independently.

10.8.1 A/D conversion trigger operation

Timer R has a function for generating A/D conversion start triggers (TRnADTRG0, TRnADTRG1 signals), freely selecting 4 trigger sources. The following 4 triggers sources are provided, which can be specified with TRnOPT2 register bits TRnAT05 to TRnAT00 and TRnOPT3 register bits TRnAT15 to TRnAT10. Here, control of the TRnADTRG0 using TRnAT05 to TRnAT00 is described. The same type of control can be achieved for the TRnADTRG1 signal with control bits TRnAT15 to TRnAT10

(1) TRnADTRG0 signal output control

- TRnOPT2 register TRnAT00 = 1: Output of A/D conversion trigger upon valley interrupt (INTTRnOD) output
- TRnOPT2 register TRnAT01 = 1: Output of A/D conversion trigger upon peak interrupt (INTTRnCD) output
- TRnOPT2 register TRnAT02 = 1: A/D conversion trigger outputtable upon compare match interrupt (INTTRnCC4) during counter up count
- TRnOPT2 register TRnAT03 = 1: A/D conversion trigger outputtable upon compare match interrupt (INTTRnCC4) during counter down count
- TRnOPT2 register TRnAT04 = 1: A/D conversion trigger outputtable upon compare match interrupt (INTTRnCC5) during counter up count
- TRnOPT2 register TRnAT05 = 1: A/D conversion trigger outputtable upon compare match interrupt (INTTRnCC5) during counter down count

The A/D conversion start trigger signals selected with bits TRnAT05 to TRnAT00 are all O_{Red} and output to the TRnADTRG0 pin.
The peak and valley interrupts (INTTRnOD, INTTRnCD) selected with bits TRnAT00 and TRnAT01 are the signals after interrupt thinning out. Therefore, they are output at the timing when interrupt thinning out control is received, and when interrupt output enable (bits TRnICE and TRnIOE) is not enabled, neither is any A/D conversion start trigger output.
Moreover, TRnOPT2 register bits TRnAT05 to TRnAT00 can be rewritten during operation.
When the A/D conversion start trigger setting bit is rewritten during operation, this is instantly reflected to the output status of the A/D conversion start trigger.
These control bits do not have a reload function and are write accessed only in the anytime write mode.

Figure 10-36: A/D Conversion Trigger Timings (1/2)

(a) when TRnICE = 1, TRnIOE = 1, TRnID4-TRnID0 = 00H (No Interrupt Thinning Out)

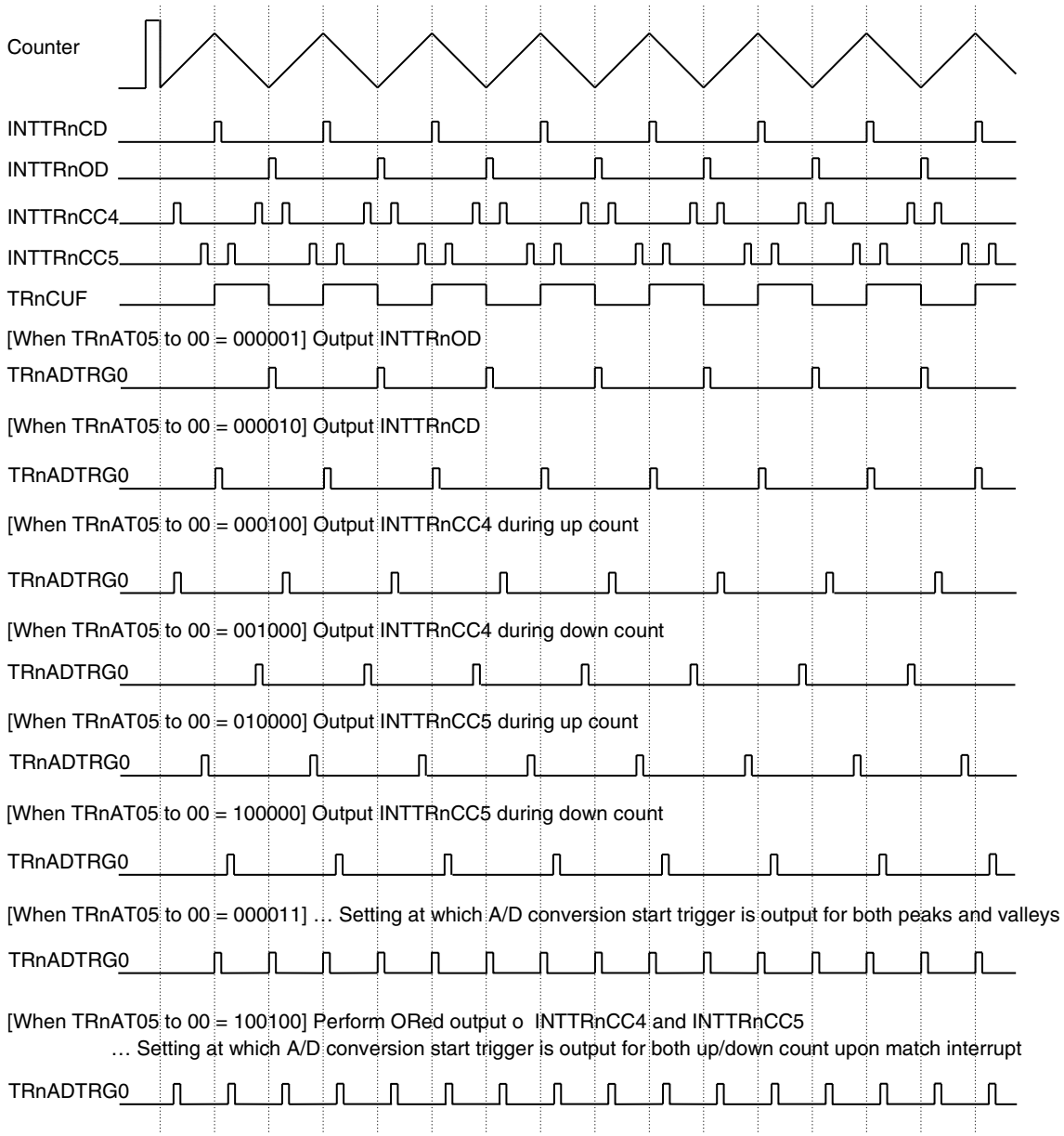
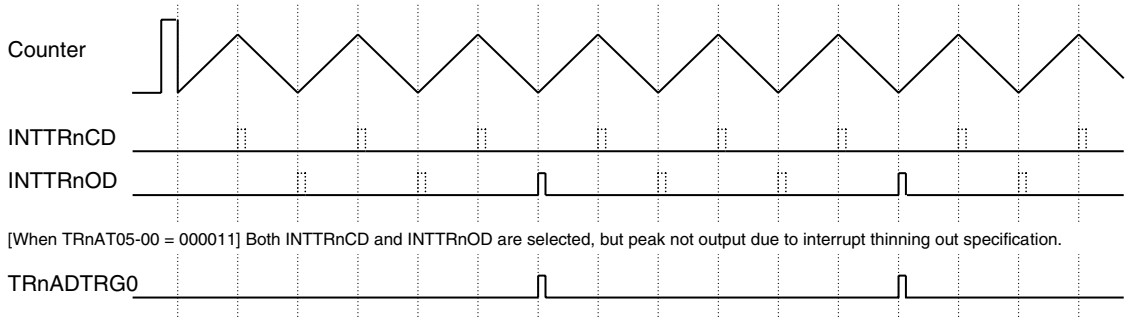
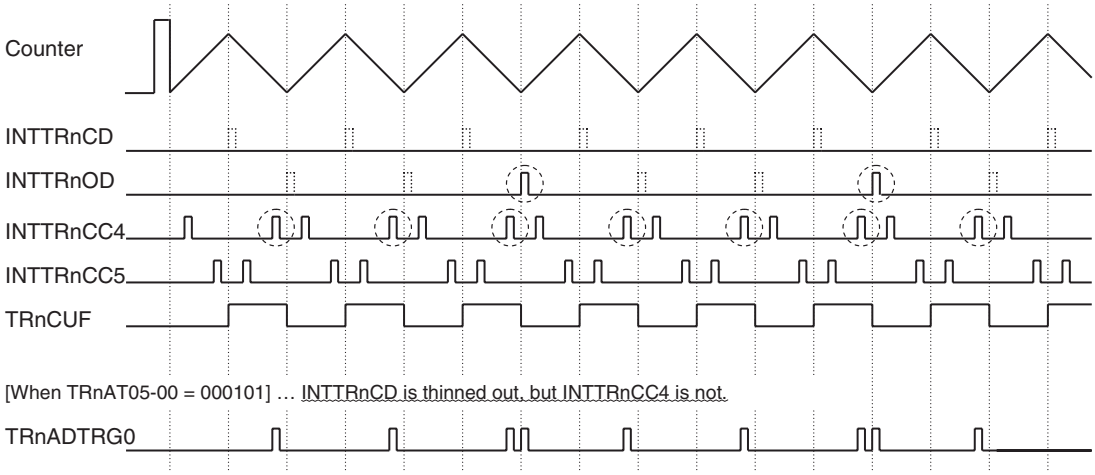


Figure 10-36: A/D Conversion Trigger Timings (2/2)

(b) when TRnICE = 0, TRnIOE = 1, TRnID4 to TRnID0 = 02H (Interrupt Thinning Out)



(c) 0when TRnICE = 0, TRnIOE = 1, TRnID4 to TRnID0 = 02H (Interrupt Thinning Out)



(2) Cautions related to A/D conversion triggers

- In the PWM mode and PWM mode with dead time, no valley interrupt (INTTRnOD) is output. Only peak interrupts (INTTRnCD) are valid.

10.9 Error Interrupts

10.9.1 Error interrupt and error signal output functions

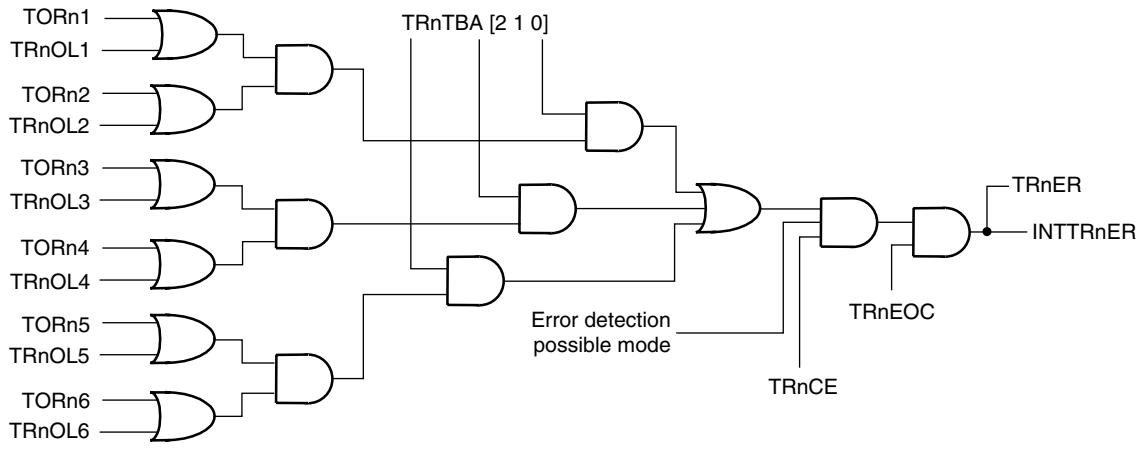
Timer R has an error interrupt (INTTRnER) and an error signal output (TRnER).
As the errors detected with timer R, normal phase/inverted phase simultaneous active (fault of dead time controller) are detected as errors in the high-accuracy T-PWM mode, PWM mode with dead time, and PWM mode. Regarding normal phase/inverted phase simultaneous active errors, following error occurrence, the error occurrence can be confirmed by reading bit TRnTBF of the TRnOPT6 register.
Moreover, detection ON/OFF switching control in each phase (TORn1/TORn2, TORn3/TORn4, TORn5/TORn6) is possible using bits TRnTBA2 to TRnTBA0 of the TRnIOC4 register.

The possibility of normal phase/inverted phase simultaneous active error detection in each mode is indicated below.

Mode	Normal Phase/Inverted Phase Simultaneous Active Detection
Interval mode	×
External event count mode	×
External trigger pulse output mode	×
One-shot pulse mode	×
PWM mode	√
Free-running mode	×
Pulse width measurement mode	×
Triangular wave PWM mode	√
High-accuracy T-PWM mode	√
PWM mode with dead time	√

Remark: √ Error detection possible
×: Error detection not possible

Figure 10-37: Error Interrupt (INTTRnER) and Error Signal (TRnER) Output Controller



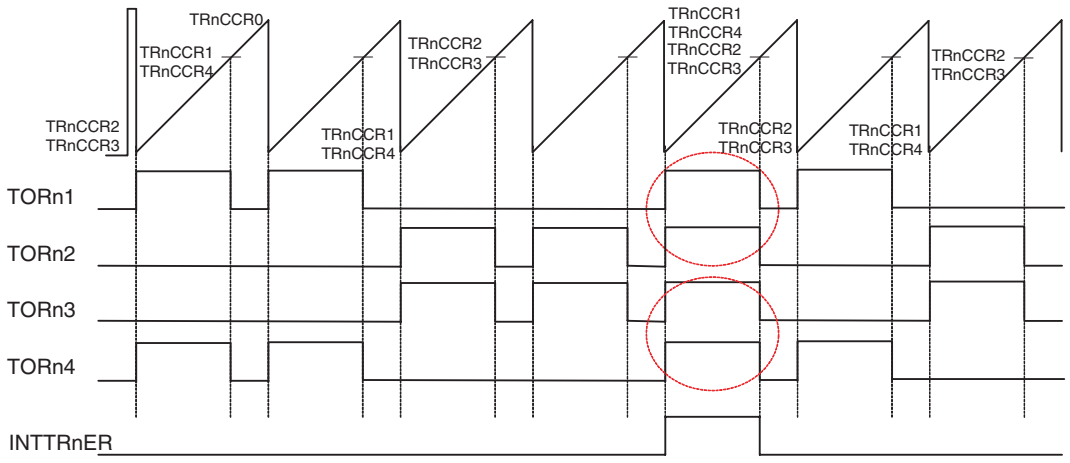
Output of the error signal (TRnER) due to normal phase/inverted phase simultaneous active error is active level during detection of normal phase/inverted phase simultaneous active.

(1) In PWM mode

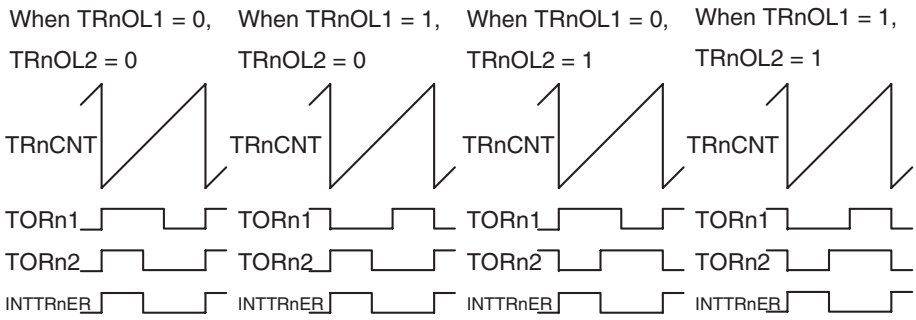
The case of normal phase/inverted phase simultaneous active in the PWM mode is described below.

As shown in the figure below, an error interrupt (INTTRnER) is output when the TRnCCR1 and TRnCCR2 registers are set so that pins TORn1 and TORn2 simultaneously output “H”. Similarly, an error interrupt (INTTRnER) is output when the TRnCCR3 and TRnCCR4 registers are set so that pins TORn3 and TORn4 simultaneously output “H”.

Figure 10-38: Error Interrupt and Error Signal Output Controller in PWM mode



If the output active level is switched by manipulating TRnIOC0 register bits TRnOL1 and TRnOL2, the following results.

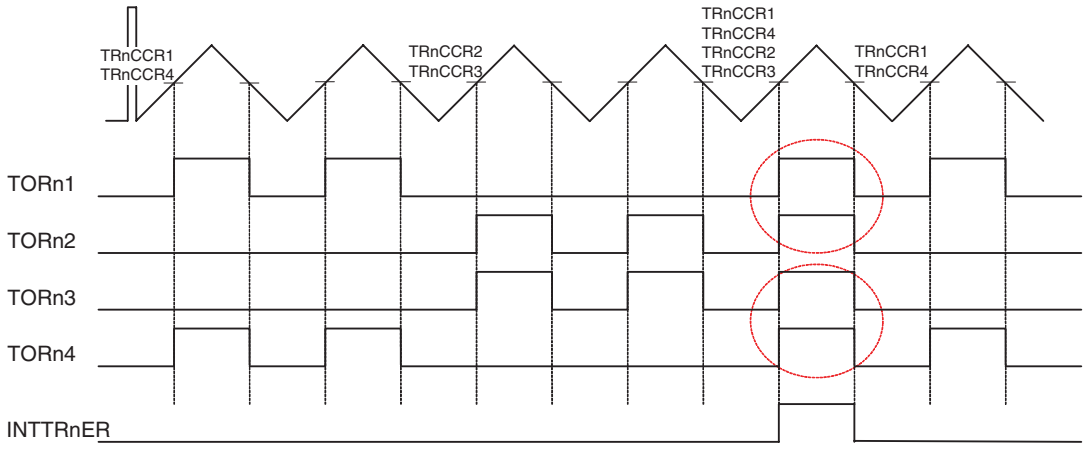


(2) In triangular wave PWM mode

The case of normal phase/inverted phase simultaneous active in the triangular wave PWM mode is described below.

As shown in the figure below, an error output (INTTRnER) is output when the TRnCCR0 and TRnCCR1 registers are set so that pins TORn1 and TORn2 simultaneously output “H”. Similarly, an error interrupt (INTTRnER) is output when the TRnCCR3 and TRnCCR4 registers are set so that pins TORn3 and TORn4 simultaneously output “H”.

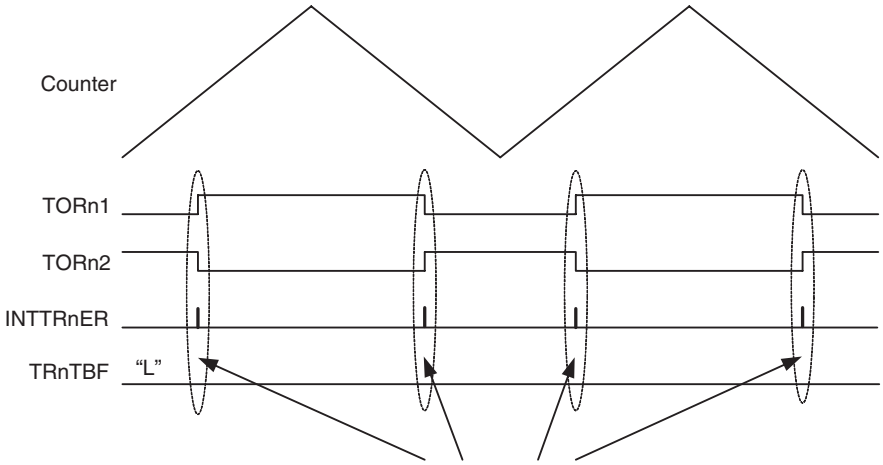
Figure 10-39: Error Interrupt and Error Signal Output Controller in triangular wave PWM mode



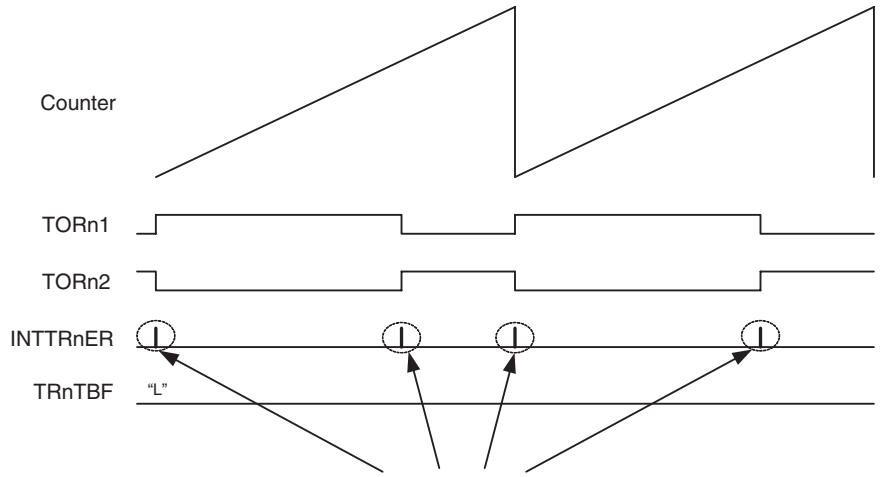
(3) In high-accuracy T-PWM mode/PWM mode with dead time

In the high-accuracy T-PWM mode and PWM mode with dead time, no error occurs except when the dead time setting is “0”. If an error occurs, this is likely due to an internal circuit fault.

Figure 10-40: Error Interrupt and Error Signal Output Controller in High-Accuracy T-PWM Mode / PWM Mode with Dead Time



A glitch may occur during normal phase/inverted phase switching.
The detection flag (TRnTBF) is not set.



A glitch may occur during normal phase/inverted phase switching.
The detection flag (TRnTBF) is not set.

10.10 Operation in Each Mode

10.10.1 Interval timer mode

(1) Outline of interval timer mode

In the interval timer mode, a compare match interrupt (INTTRnCC0) occurs and the counter is cleared upon a match between the setting value of the TRnCCR0 register and the counter value. The occurrence interval for this counter and TRnCCR0 register match interrupt becomes the interval time.

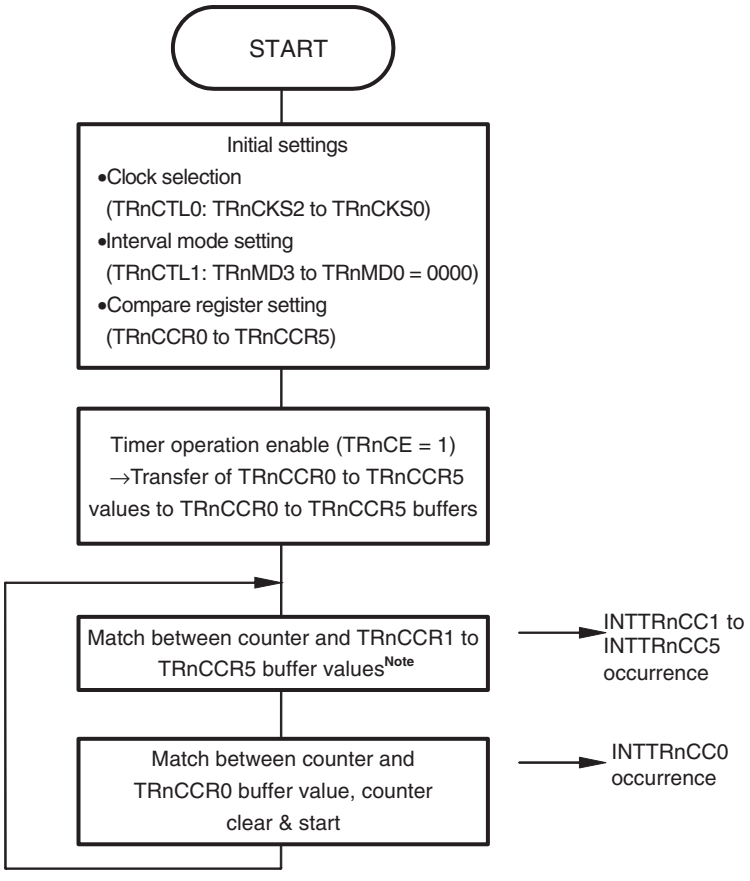
In the interval timer mode, the counter is cleared only upon a match between the counter and the value of the TRnCCR0 register. Counter clearing using the TRnCCR1 to TRnCCR5 registers is not performed.

However, the setting values of the TRnCCR1 to TRnCCR5 registers are compared to the counter values transferred to the TRnCCR1 to TRnCCR5 buffer registers and compare match interrupts (INTTRnCC1 to INTTRnCCR5) are output.

The TRnCCR0 to TRnCCR5 registers can be rewritten using the anytime write method, regardless of the value of bit TRnCE.

Pins TORn0 to TORn7 are toggle output controlled when bits TRnOE0 to TRnOE7 are set to 1.

Figure 10-41: Basic Operation Flow in Interval Timer Mode



Note: In the case of a match between the counter and TRnCCR1 to TRnCCR5 registers, the counter is not cleared.

(2) Interval timer mode operation list

(a) Compare registers

Register	Rewrite Method	Rewrite during Operation	Function
TRnCCR0	Reload	Possible	Compare value
TRnCCR1 to TRnCCR3	Reload	Possible	Compare value
TRnCCR4, TRnCCR5	Reload	Possible	Compare value

(b) Input pins

Pin	Function
TIR1m	- (m = 0 to 3)
TTRGR1	-
TEVTR1	-

(c) Output pins

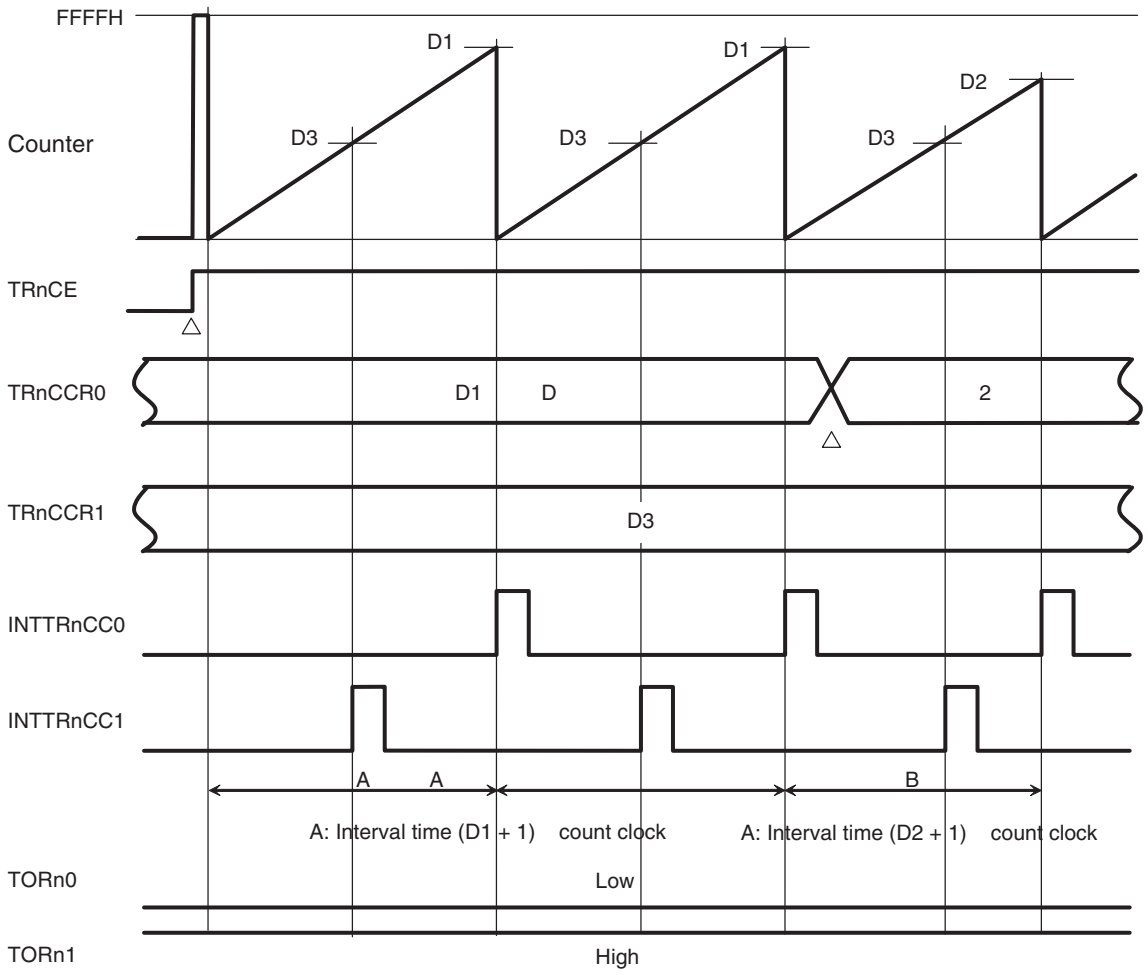
Pin	Function
TORnm	Toggle output upon TRnCCRm register compare match (m = 0 to 5)
TORn6, TORn7	-

(d) Interrupts

Interrupt	Function
INTTRnCCm	TRnCCRm register compare match (m = 0 to 5)
INTTRnOV	-
INTTRnER	-

Figure 10-42: Basic Timing in Interval Timer Mode (1/2)

(a) When D1>D2>D3, only value of TRnCCR0 register is rewritten, TORn0 and TORn1 are not output (TRnOE0, 1 = 0, TRnOL0 = 0, TRnOL1 = 1)



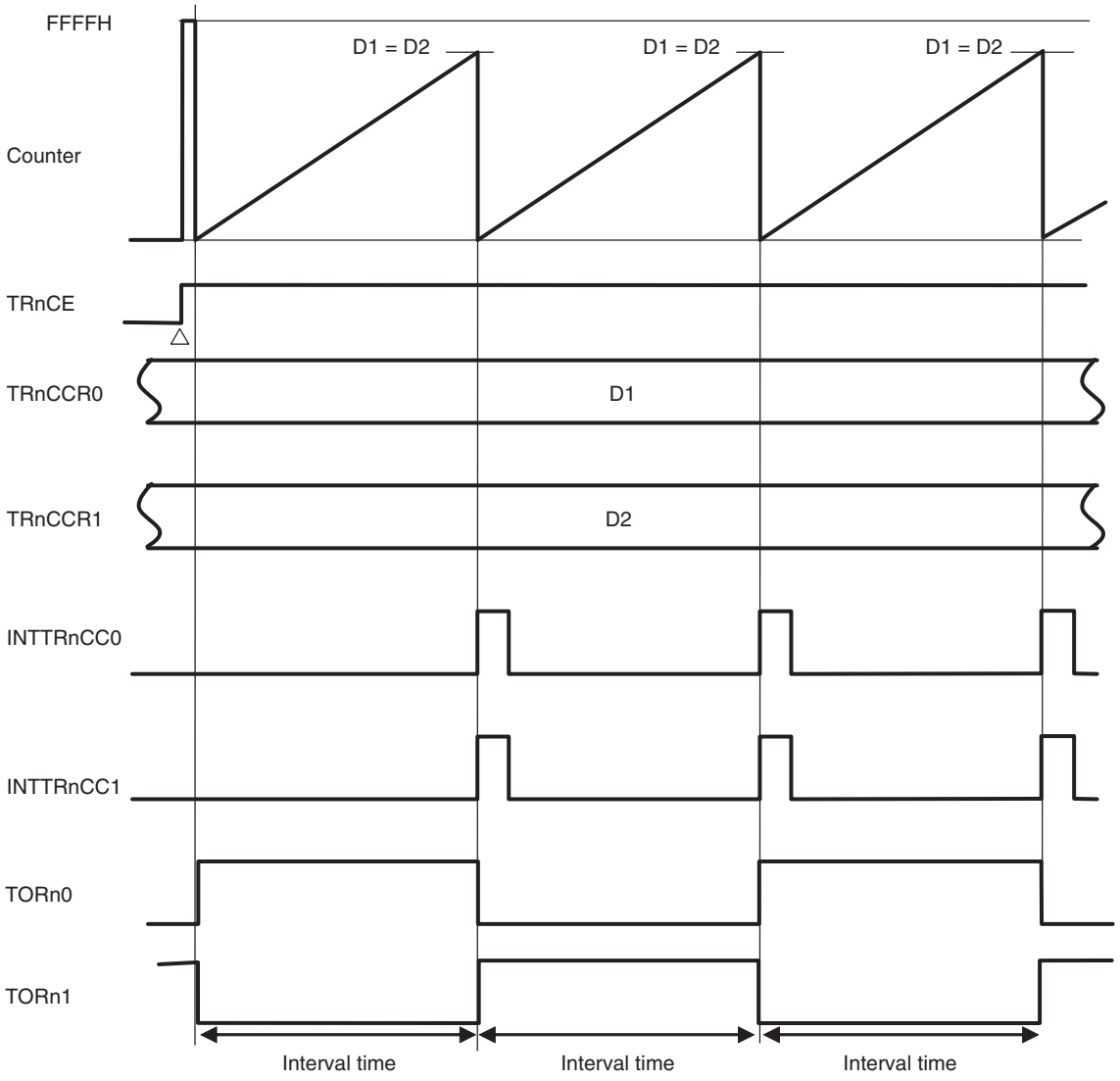
Remarks: 1. D1, D2: Setting values of TRnCCR0 register (0000H to FFFFH)
D3: Setting values of TRnCCR1 register (0000H to FFFFH)

2. Interval time = (Dm + 1) × (count clock cycle)

3. m = 1 to 3, n = 0, 1

Figure 10-42: Basic Timing in Interval Timer Mode (2/2)

(b) When D1 = D2, values of TRnCCR0 and TRnCCR1 registers not rewritten, TORn1 output performed (TRnOE0, 1 = 1, TRnOL0 = 0, TRnOL1 = 1)



Remarks: 1. D1: Setting value of TRnCCR0 register (0000H to FFFFH)
D2: Setting value of TRnCCR1 register (0000H to FFFFH)

2. Interval time = (Dm + 1) × (count clock cycle)

3. TORn0, TORn1 toggle time = (Dm + 1) × (count clock cycle)

4. m = 1, 2, n = 0, 1

10.10.2 External event count mode

(1) Outline of external event count mode

In the external event count mode, count up starts upon external event input (TEVTRn pin). (The external event input (TEVTRn) is used as the count clock, regardless of bit TRnEEE of the TRnCTL1 register.)

In the external event count mode, the counter is cleared only upon a match between the counter and the value of the TRnCCR0 register. Counter clearing using the TRnCCR1 to TRnCCR5 registers is not performed.

However, the values of the TRnCCR1 to TRnCCR5 registers are transferred to the TRnCCR1 to TRnCCR5 buffer registers, compared to the counter value, and compare match interrupts (INTTRnCC1to INTRnCCR5) are output.

The TRnCCR0 to TRnCCR5 registers can be rewritten with the anytime write method, regardless of the value of bit TRnCE.

Pins TORn0 to TORn7 are toggle output controlled when bits TRnOE0 to TRnOE7 are set to 1.

When using only one compare register channel, it is recommended to set the TRnCCR1 to TRnCCR5 registers to FFFFH.

[External event count operation flow]

<1> TRnCTL1 register bits TRnMD3 to TRnMD0 = 0001B (mode setting)
Edge detection set with TRnIOC2 register bits TRnEES1 and TRnEES0 (TRnEES1, TRnEES0 = setting other than 00B)
<2> TRnCTL0 register bit TRnCE = 1 (count enable)
<3> TEVTRn pin input edge detection (count-up start)

Cautions: 1. In the case of the external event count mode, when the setting value of the TRnCCR0 register is set to m, the number of TEVTRn pin input edge detection times is m+1.

2. Do not set the value of the TRnCCR0 register to 0000H in external event count mode.

Remark: n = 0, 1

(2) External event count mode operation list

(a) Compare registers

Register	Rewrite Method	Rewrite during Operation	Function
TRnCCR0	Anytime rewrite	Possible	Compare value
TRnCCR1 to TRnCCR3	Anytime rewrite	Possible	Compare value
TRnCCR4, TRnCCR5	Anytime rewrite	Possible	Compare value

(b) Input pins

Pin	Function
TIR1m	- (m = 0 to 3)
TTRGR1	-
TEVTR1	-

(c) Output pins

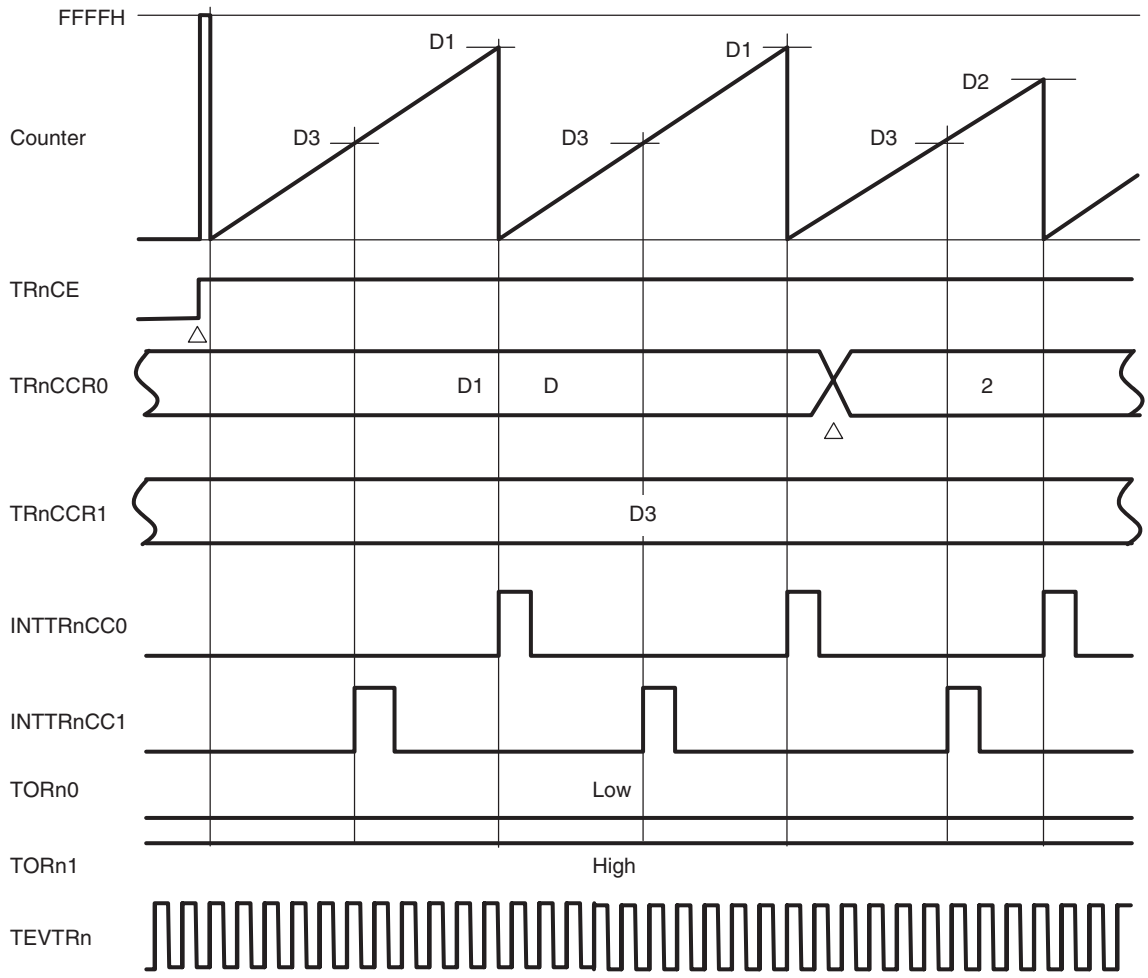
Pin	Function
TORnm	Toggle output upon TRnCCRm register compare match (m = 0 to 5)
TORn6, TORn7	-

(d) Interrupts

Interrupt	Function
INTTRnCCm	TRnCCRm register compare match (m = 0 to 5)
INTTRnOV	-
INTTRnER	-

Figure 10-43: Basic Operation Timing in External Event Count Mode (1/4)

(a) When D1>D2>D3, only value of TRnCCR0 register is rewritten, TORn0 and TORn1 are not output. The signal input from TEVTRn and internally synchronized is counted as the count clock (TRnOE0, 1 = 0, TRnOL0 = 0, TRnOL1 = 1)



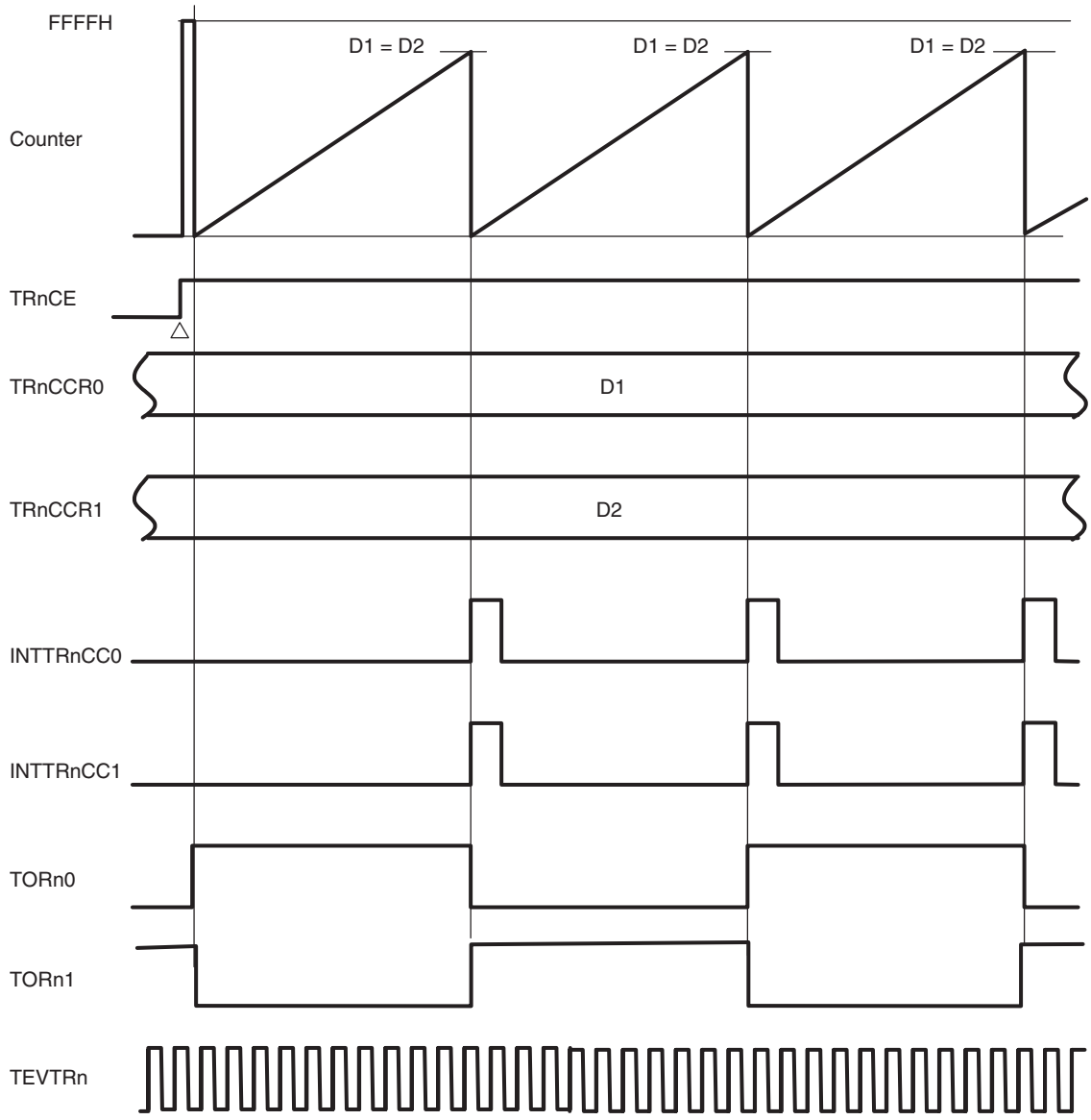
Remarks: 1. D1, D2: Setting values of TRnCCR0 register (0000H to FFFFH)
D3: Setting value of TRnCCR1 register (0000H to FFFFH)

2. Number of event counts = (Dm + 1) (m = 1, 2)

3. n = 0, 1

Figure 10-43: Basic Operation Timing in External Event Count Mode (2/4)

(b) When D1 = D2, TRnCCR0 and TRnCCR1 register values are not rewritten, TORn0 and TORn1 are output (TRnOE0, 1 = 1, TRnOL0 = 0, TRnOL1 = 1)



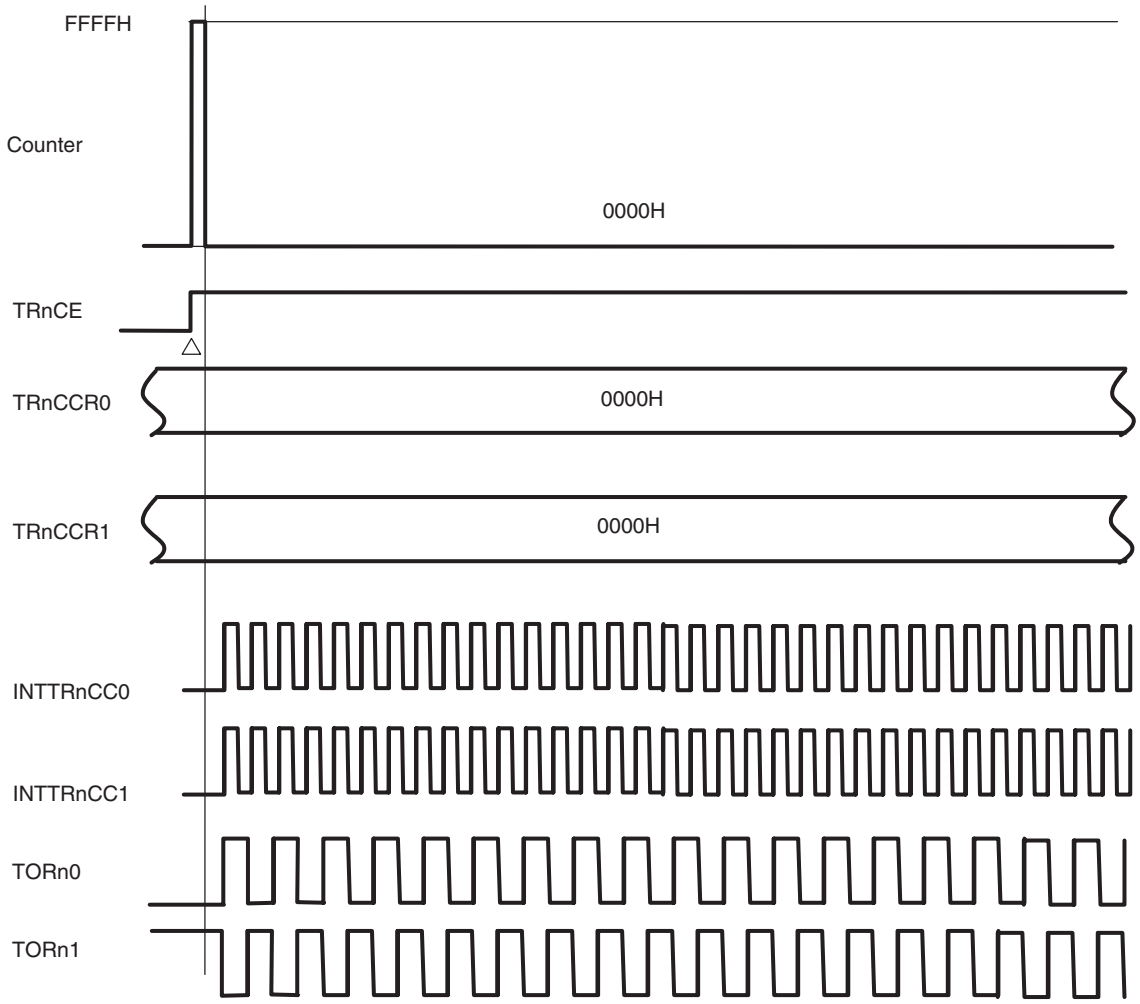
Remarks: 1. D1: Setting value of TRnCCR0 register (0000H to FFFFH)
D2: Setting value of TRnCCR1 register (0000H to FFFFH)

2. Number of event counts = (Dm + 1) (m = 1, 2)

3. n = 0, 1

Figure 10-43: Basic Operation Timing in External Event Count Mode (3/4)

(c) When D1 = D2, TRnCCR0 and TRnCCR1 register values are not rewritten, TORn0 and TORn1 are output (TRnOE0, 1 = 1, TRnOL0 = 0, TRnOL1 = 1)



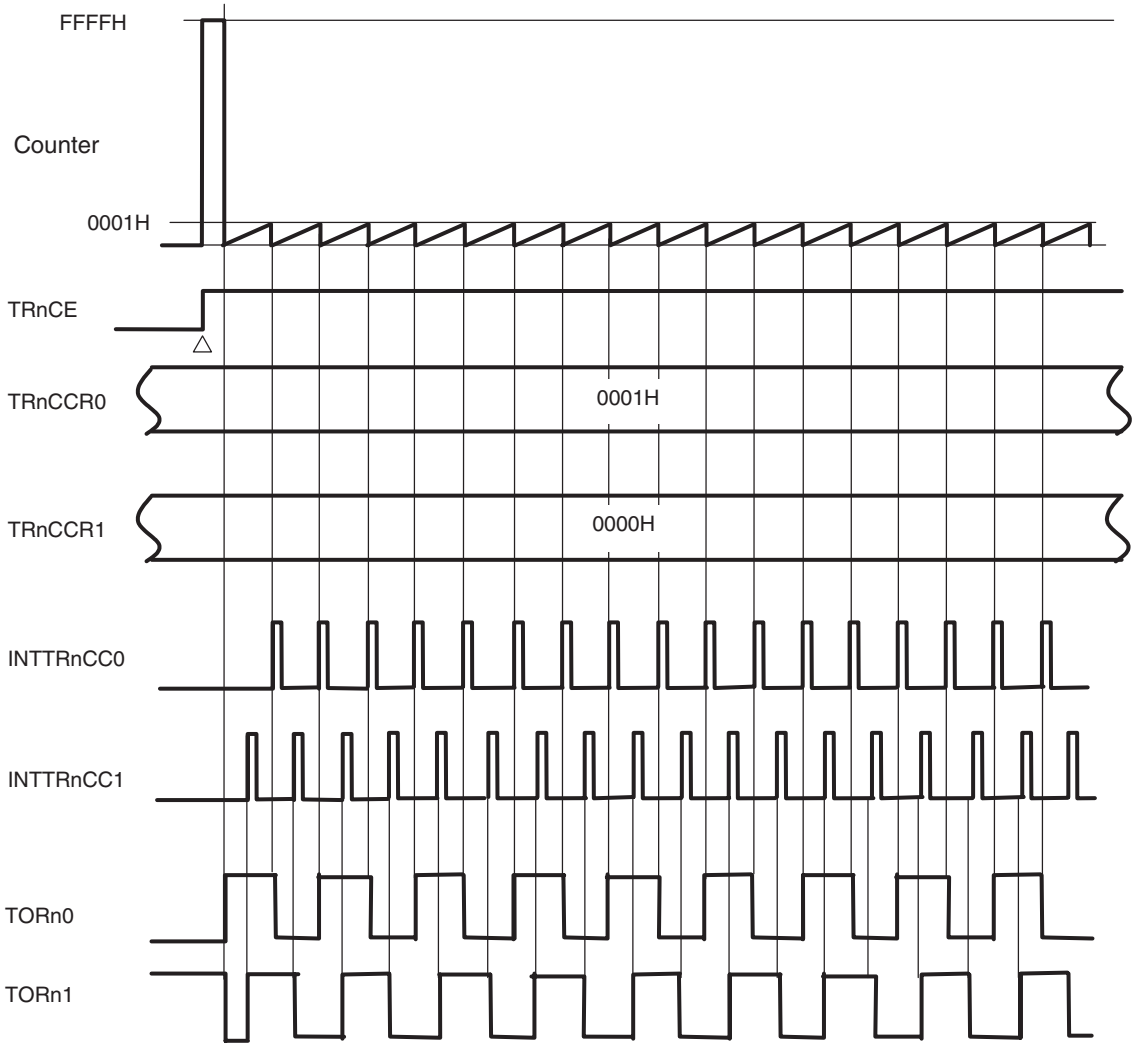
Remarks: 1. D1: Setting value of TRnCCR0 register (0000H)
D2: Setting value of TRnCCR1 register (0000H)

2. Number of event counts = (Dm + 1) (m = 1, 2)

3. n = 0, 1

Figure 10-43: Basic Operation Timing in External Event Count Mode (4/4)

(d) When D1 = D2, TRnCCR0, TRnCCR1 register values are not rewritten, TORn0 and TORn1 are output (TRnOE0, 1 = 1, TRnOL0 = 0, TRnOL1 = 1)



Remarks: 1. D1: Setting value of TRnCCR0 register (0001H)
D2: Setting value of TRnCCR1 register (0000H)

2. Number of event counts = (Dm + 1) (m = 1, 2)

3. n = 0, 1

10.10.3 External trigger pulse output mode

(1) Outline of external trigger pulse output mode

When, in the external trigger pulse mode, the duty is set to the TRnCCR1 to TRnCCR5 registers, the cycle is set to the TRnCCR0 register, and TRnCE = 1 is set, external trigger input (TTRGRn pin) wait results, with the counter remaining stopped at FFFFH. Upon detection of the valid edge of external trigger input (TTRGRn pin), or when the TRnEST bit of the TRnCTL1 register is set, count up starts. An external trigger pulse is output from pins TORn1 to TORn5, and toggle output is performed from pin TORn0 upon a match with the TRnCCR0 register. Moreover, during the count operation, upon a match between the counter and the TRnCCR0 register, a compare match interrupt (INTTRnCC0) is output, and upon a match between the counter and TRnCCR1 to TRnCCR5 registers, compare match interrupts (INTTRnCC1 to INTTRnCC5) are output.

The TRnCCR0 to TRnCCR5 registers can be rewritten during count operation. Compare register reload is performed at the timing when the counter value and the TRnCCR0 register match.

However, when write access to the TRnCCR1 register is performed, the next reload timing becomes valid, so that even if wishing to rewrite only the value of the TRnCCR0 register, write the same value to the TRnCCR1 register. In this case, reload is not performed even if only the TRnCCR0 register is rewritten.

If, during operation in the external trigger pulse output mode, the external trigger (TTRGRn pin) edge is detected several times, or if the TRnEST bit of the TRnCTL1 register is set (to 1), the counter is cleared and count up is resumed. Moreover, if at this time, the TORn1 to TORn5 pins are in the low level status, the TORn1 to TORn5 pin outputs become high level when an external trigger is input. If the TORn1 pin is in the high level status, it remains high level even if external trigger input occurs.

In the external trigger pulse output mode, the TRnCCR0 to TRnCCR3 registers have their function fixed as compare registers, so the capture function cannot be used.

Remark: n = 0, 1

Caution: In the external trigger pulse mode, set bit TRnEEE of the TRnCTL1 register to 0.

(2) External trigger pulse output mode operation list

(a) Compare registers

Register	Rewrite Method	Rewrite during Operation	Function
TRnCCR0	Reload	Possible	Cycle
TRnCCR1 to TRnCCR3	Reload	Possible	Duty
TRnCCR4, TRnCCR5	Reload	Possible	Duty

(b) Input pins

Pin	Function
TIR1m	- (m = 0 to 3)
TTRGR1	Counter clear & start through external trigger input
TEVTR1	Timer count through external event count input

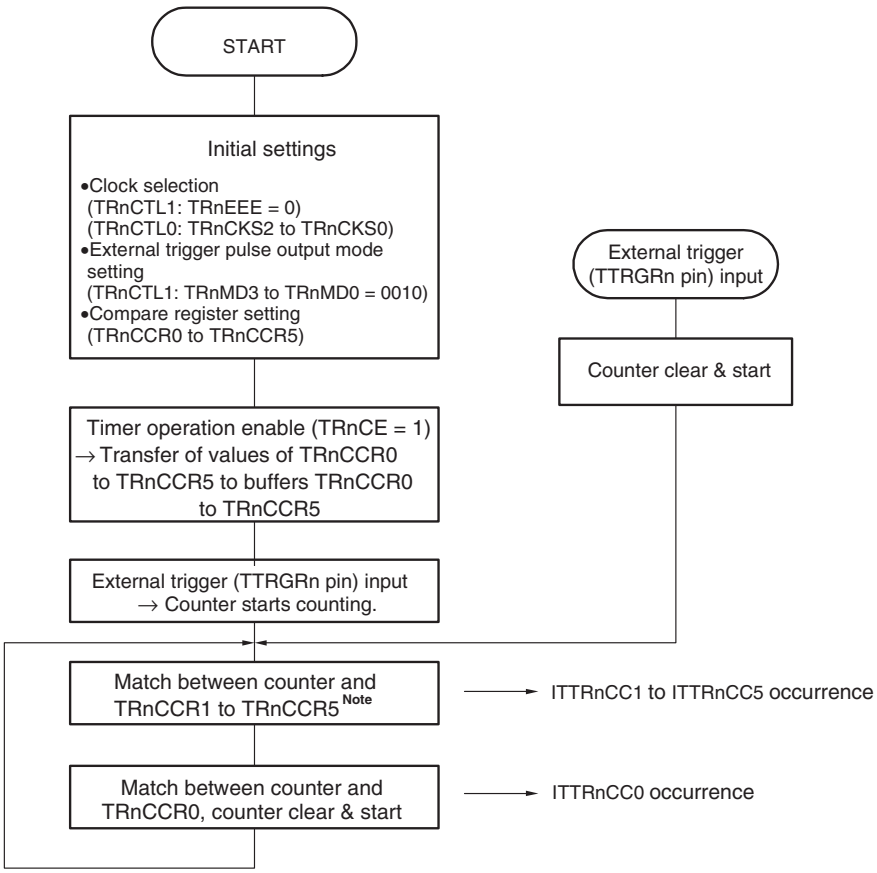
(c) Output pins

Pin	Function
TORn0	Toggle output upon TRnCCR0 register compare match or external trigger input
TORnm	External trigger pulse waveform output (m = 1 to 5)
TORn6, TORn7	-

(d) Interrupts

Interrupt	Function
INTTRnCCm	TRnCCRm register compare match (m = 0 to 5)
INTTRnOV	-
INTTRnER	-

Figure 10-44: Basic Operation Flow in External Trigger Pulse Output Mode

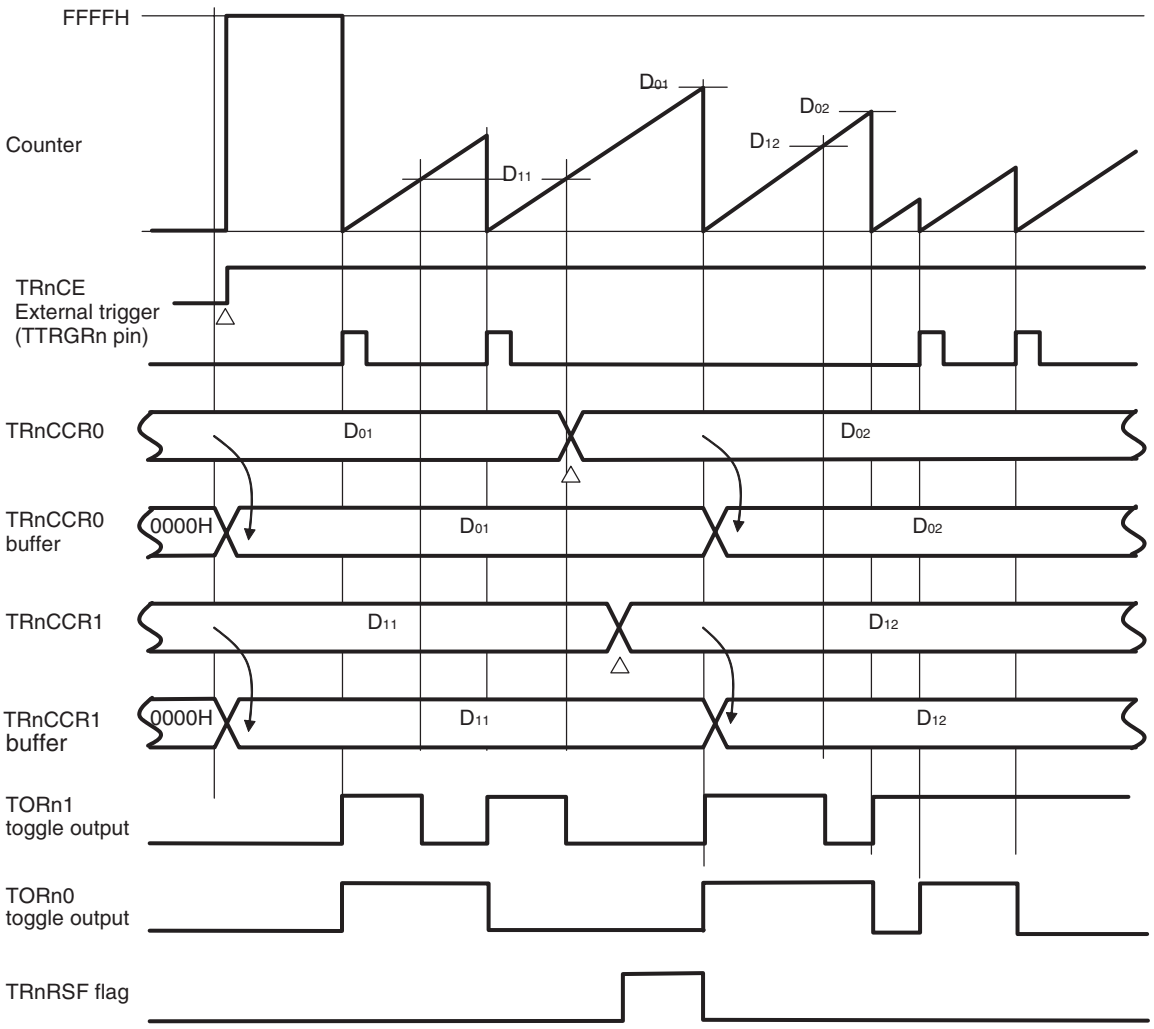


Note: The counter is not cleared upon a match between the counter and the TRnCCR1 to TRnCCR5 buffer register.

Remark: n = 0, 1

Figure 10-45: Basic Operation Timing in External Trigger Pulse Output Mode

(a) When values of TRnCCR0 and TRnCCR1 registers are rewritten, TORn0 and TORn1 are output (TRnOE0, 1 = 1, TRnOL0, 1 = 0)



Remarks: 1. D01, D02: Setting values of TRnCCR0 register (0000H to FFFFH)
D11, D12: Setting values of TRnCCR1 register (0000H to FFFFH)

2. TORn1 (PWM) duty = (setting value of TRnCCR1 register) × (count clock cycle)
TORn1 (PWM) cycle = (setting value of TRnCCR0 register + 1) × (count clock cycle)

3. Pin TORn0 is toggled when the counter is cleared immediately following count start.

4. n = 0, 1

10.10.4 One-shot pulse mode

(1) Outline of one-shot pulse mode

When, in the one-shot pulse mode, the duty is set to the TRnCCR0 register, the output duty delay value is set to the TRnCCR1 to TRnCCR5 registers, and bit TRnCE of the TRnCTL0 register is set to 1, external trigger input (TTRGRn pin) wait results, with the counter remaining stopped at FFFFH. Upon detection of the valid edge of external trigger input (TTRGRn pin), or when bit TRnEST of the TRnCTL0 register is set to 1, count up starts. The TORn1 to TORn5 pins become high level upon a match between the counter and TRnCCR1 to TRnCCR5 registers. Moreover, upon a match between the counter and TRnCCR0 register, the TORn1 to TORn5 pins become low level, and the counter is cleared to 0000H and then stops. The TORn0 pin performs toggle output during the count operation upon a match between the counter and the TRnCCR0 buffer register. Moreover, upon a match between the counter and TRnCCR0 register during count operation, a compare match interrupt (INTTRnCC0) is output, and upon a match between the counter and TRnCCR1 to TRnCCR5 buffer registers, compare match interrupts (INTTRnCC1 to INTTRnCCR5) are output.
The TRnCCR0 and TRnCCR1 registers can be rewritten using the anytime write method, regardless of the value of bit TRnCE.
Even a trigger is input during the counter operation, it is ignored. Be sure to input the second trigger when the counter is stopped at 0000H.
In the one-shot pulse mode, registers TRnCCR0 to TRnCCR3 have their function fixed as compare registers, so the capture function cannot be used.

[One-shot pulse operation flow]

<1> TRnCTL1 register bits TRnMD3 to TRnMD0 = 0011B (One-shot pulse mode)
<2> TRnCCR0 register setting (duty setting), TRnIOC0 register bit TRnOE1 = 1 (TORn1 pin output enable)
<3> TRnCTL0 register bit TRnCE = 1 (counter operation enable):TORn1 = Low-level output
<4> TRnCTL1 register bit TRnEST = 1 or TTRGRn pin edge detection (count-up start): TORn1 = Low-level output
<5> Match between counter value and TRnCCR1 buffer register: TORn1 = High-level output
<6> Match between counter value and TRnCCR0 buffer register:TORn1 = Low-level output, count clear
<7> Count stop: TORn1 = Low-level output
<8> TRnCE = 0 (operation reset)
<9> <1> to <2> can be in any order.

Caution: In the one-shot pulse mode, set bit TRnEEE of the TRnCTL1 register to 0.

(2) One-shot pulse mode operation list

(a) Compare registers

Register	Rewrite Method	Rewrite during Operation	Function
TRnCCR0	Anytime rewrite	Possible	Cycle
TRnCCR1 to TRnCCR3	Anytime rewrite	Possible	Output delay value
TRnCCR4, TRnCCR5	Anytime rewrite	Possible	Output delay value

(b) Input pins

Pin	Function
TIR1m	- (m = 0 to 3)
TTRGR1	-
TEVTR1	-

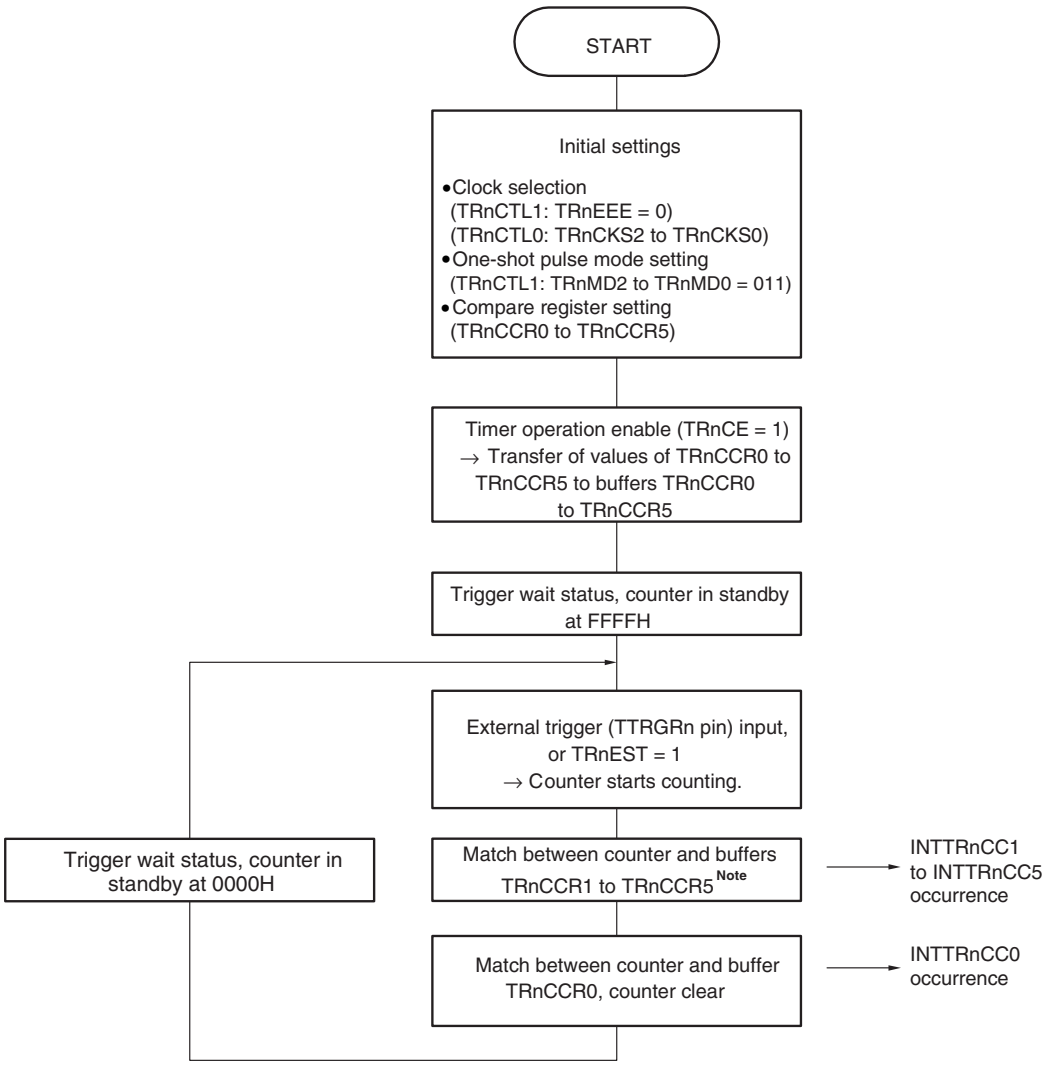
(c) Output pins

Pin	Function
TORn0	Active at count start, inactive upon TRnCCR0 register match
TORnm	Active upon TRnCCRm register match, inactive upon TRnCCR0 register match (m = 1 to 5)
TORn6, TORn7	-

(d) Interrupts

Interrupt	Function
INTTRnCCm	TRnCCRm register compare match
INTTRnOV	-
INTTRnER	-

Figure 10-46: Basic Operation Flow in One-Shot Pulse Mode



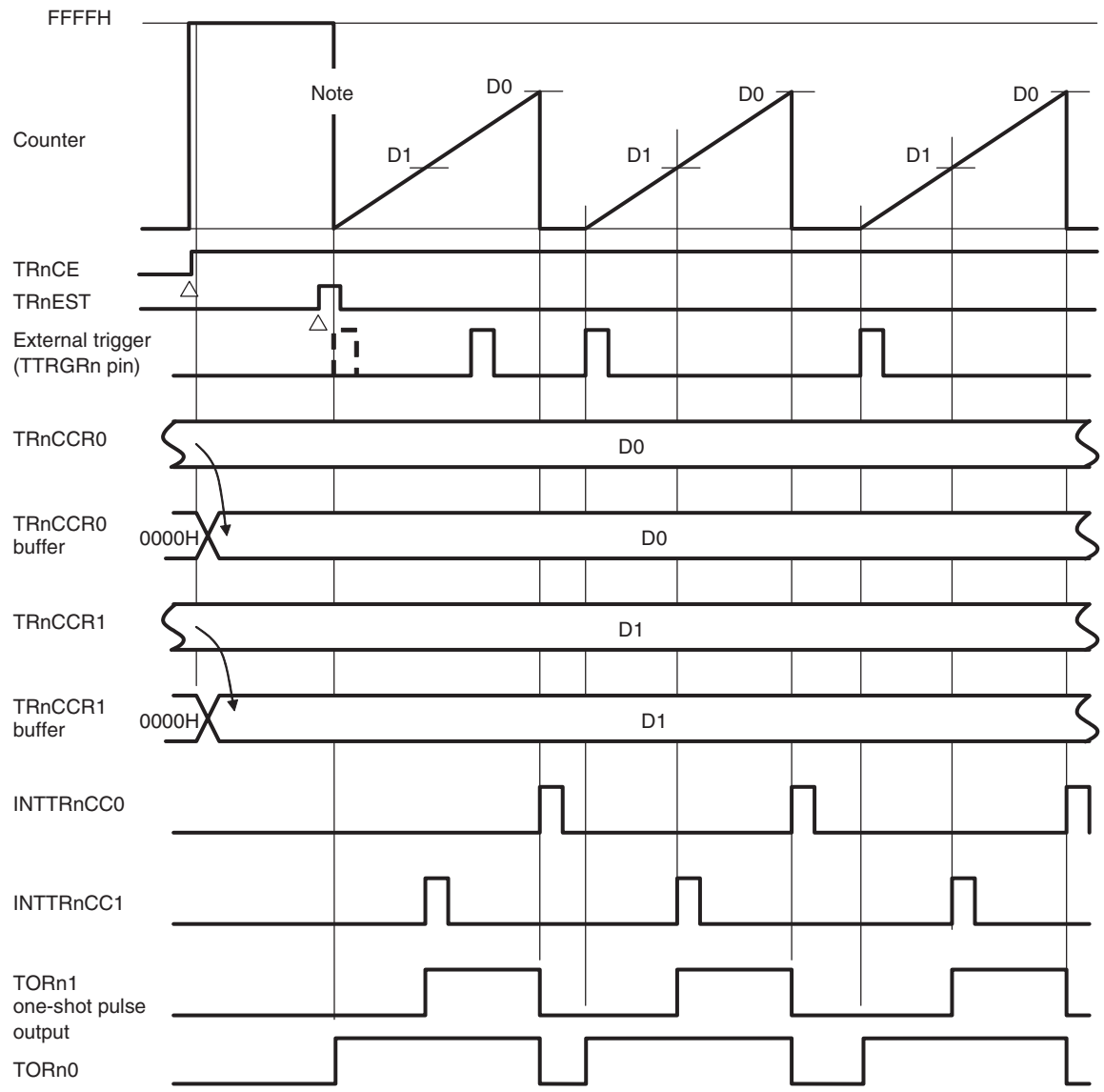
Note: The counter is not cleared upon a match between the counter and the TRnCCR1 to TRnCCR5 buffer registers.

Caution: The counter is not cleared even if trigger input is realized while the counter counts up, and the trigger input is ignored.

Remark: n = 0, 1

Figure 10-47: Basic Operation Timing in One-Shot Pulse Mode

(a) (TRnOE0, 1 = 1, TRnOL0, 1 = 0)



Note: Count up starts when the value of TRnEST becomes 1 or TTRGRn is input.

Remarks: 1. D0: Setting value of TRnCCR0 register (0000H to FFFFH)
D1: Setting value of TRnCCR1 register (0000H to FFFFH)

2. TORn1 (output delay) = (setting value of TRnCCR1 register) × (count clock cycle)
TORn1 (output pulse width) = {(setting value of TRnCCR0 register +1) - (setting value of TRnCCR1 register)} × (count clock cycle)

3. n = 0, 1

10.10.5 PWM mode

(1) Outline of PWM mode

When, in the PWM mode, the duty is set to the TRnCCR1 to TRnCCR5 registers, the cycle is set to the TRnCCR0 register, and TRnCE = 1 is set, variable duty PWM output is performed from pins TORn1 to TORn5.

Simultaneously with the start of count up operation, pins TORn1 to TORn5 becomes high level, and upon a match between the counter and the TRnCCR1 to TRnCCR5 registers, becomes low level. Next, the TORn1 to TORn5 pins become high level upon a match with the TRnCCR0 register. The TORn0 pin performs toggle output upon a match with the TRnCCR0 buffer register.

During count operation, a compare match interrupt (INTTRnCC0) is output upon a match between the counter and TRnCCR0 register, and compare match interrupts (INTTRnCC1 to INTTRnCC5) are output upon a match between the counter and TRnCCR1 to TRnCCR5 registers.

The TRnCCR0 to TRnCCR5 registers can be rewritten during count operation. Compare register reload occurs upon a match between the counter value and the TRnCCR0 buffer register.

However, since the next reload timing becomes valid when the TRnCCR1 register is written to, write the same value to the TRnCCR1 register even when wishing to rewrite only the value of the TRnCCR0 register. Reloading is not performed if only the TRnCCR0 register is rewritten.

In the PWM mode, the TRnCCR0 to TRnCCR3 registers have their function fixed as compare registers, so the capture function cannot be used.

(2) PWM mode operation list

(a) Compare register

Register	Rewrite Method	Rewrite during Operation	Function
TRnCCR0	Reload	Possible	Cycle
TRnCCR1 to TRnCCR3	Reload	Possible	Duty
TRnCCR4, TRnCCR5	Reload	Possible	Duty

(b) Input pins

Pin	Function
TIR1m	- (m = 0 to 3)
TTRGR1	-
TEVTR1	-

(c) Output pins

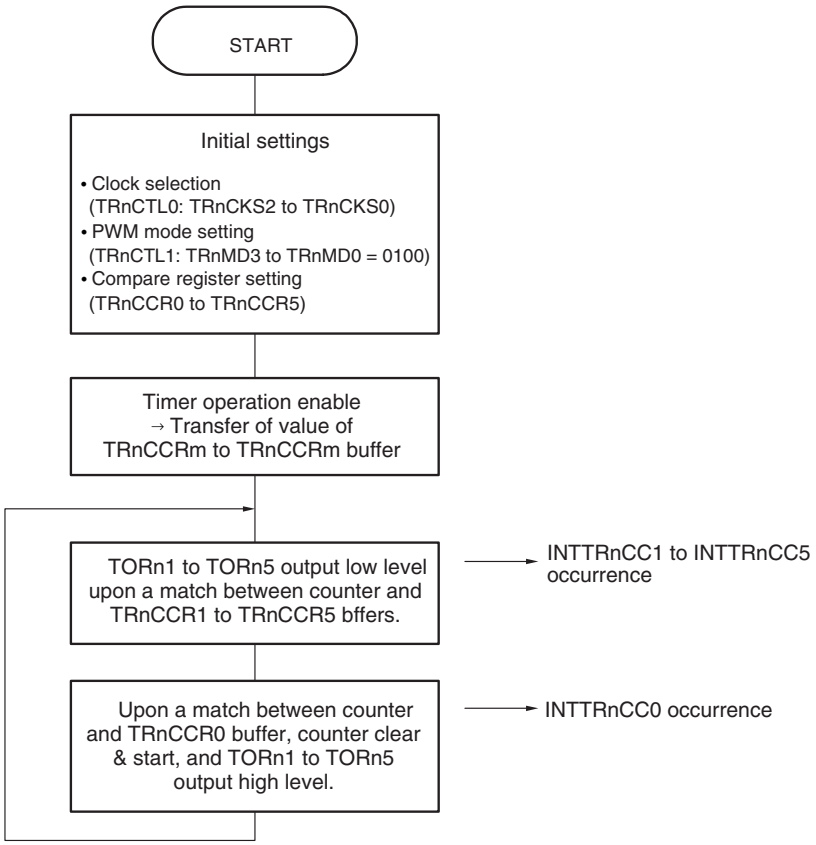
Pin	Function
TORn0	Toggle output upon TRnCCR0 register compare match
TORnm	PWM output upon TRnCCRm register compare match (m = 1 to 5)
TORn6	-
TORn7	Pulse output through A/D conversion trigger

(d) Interrupts

Interrupt	Function
INTTRnCCm	TRnCCRm register compare match
INTTRnOV	-
INTTRnER	Error

Figure 10-48: Basic Operation Mode in PWM Mode (1/2)

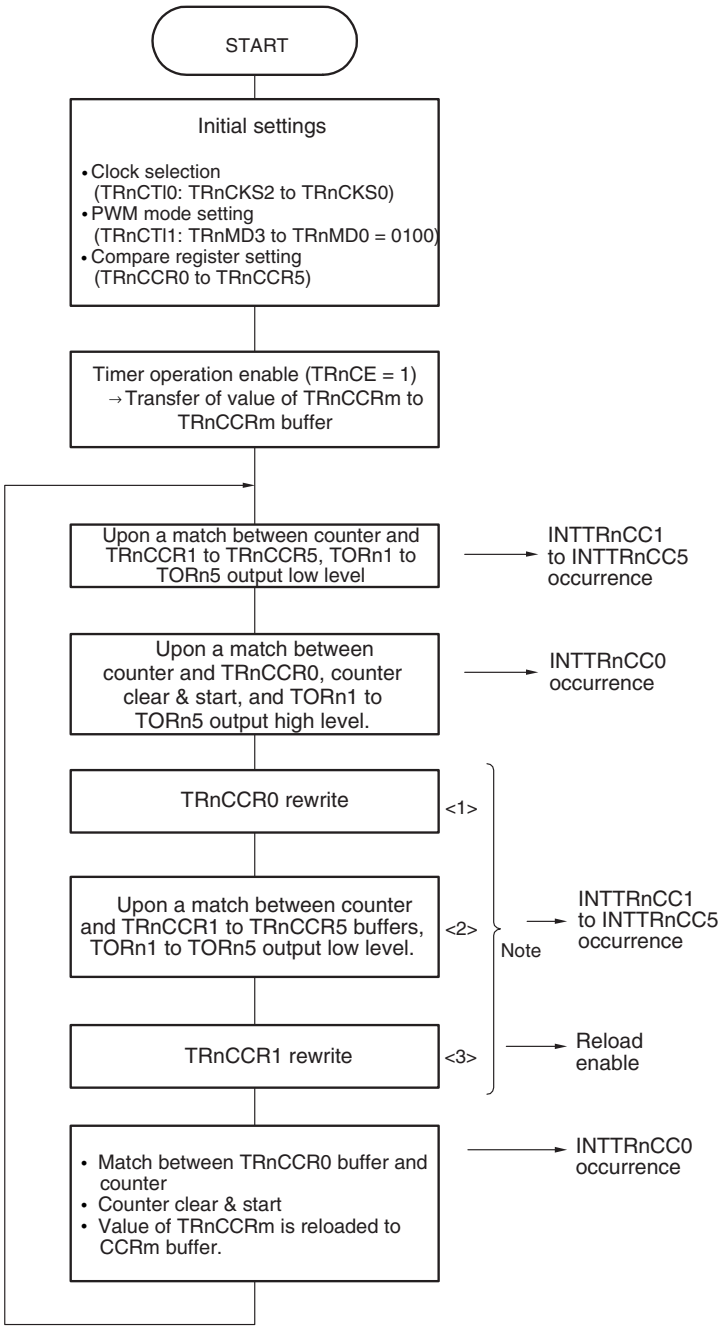
(a) When values of TRnCCR0 to TRnCCR5 registers are rewritten during timer operation



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

Figure 10-48: Basic Operation Flow in PWM Mode (2/2)

(b) When values of TRnCCR0 to TRnCCR5 registers are rewritten during timer operation

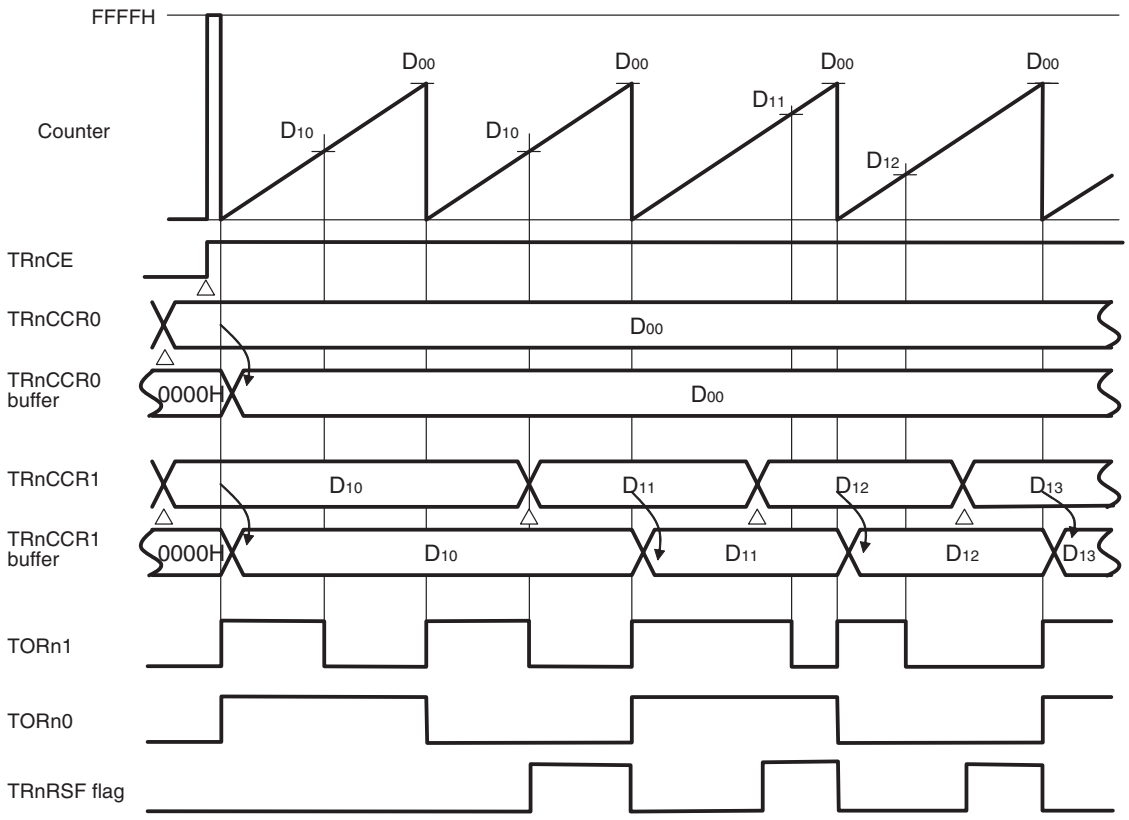


Note: Regarding the sequence, the timing of <2> may differ depending on the <1> or <3> rewrite timing, the value of the TRnCCR1 register, etc., but of <1> and <3>, always make <3> the last.

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

Figure 10-49: Basic Operation Timing in PWM Mode (1/2)

(a) When only value of TRnCCR1 is rewritten, and TORn0 and TORn1 are output (TRnOE0, 1 = 1, TRnOL0, 1 = 0)



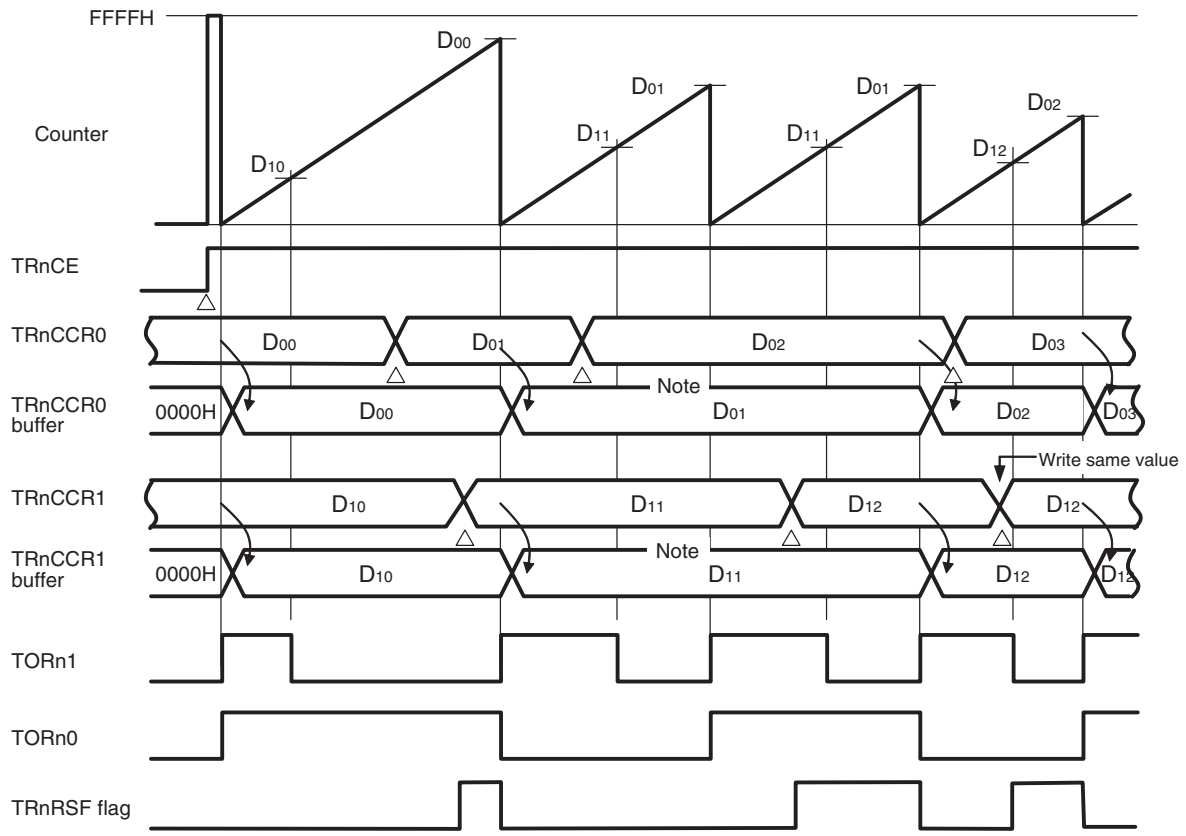
Remarks: **1.** D00: Setting value of TRnCCR0 register (0000H to FFFFH)
D10, D11, D12, D13: Setting values of TRnCCR1 register (0000H to FFFFH)

2. TORn1 (PWM) duty = (setting value of TRnCCR1 register) × (count clock cycle)
TORn1 (PWM) cycle = (setting value of TRnCCR0 register + 1) × (count clock cycle)
TORn0 is toggled immediately following counter start and at (setting value of TRnCCR0 register + 1) × (count clock cycle)

3. n = 0, 1

Figure 10-49: Basic Operation Timing in PWM Mode (2/2)

(b) When values of TRnCCR0 and TRnCCR1 register are rewritten, TORn0 and TORn1 are output (TRnOE0, 1 = 1, TRnOL0, 1 = 0)



Note: The TRnCCR1 register was not written to, so transfer to the TRnCCR0 buffer register was not performed. Held until the next reload timing.

Remarks: 1. D00, D01, D02, D03: Setting values of TRnCCR0 register (0000H to FFFFH)
D10, D11, D12, D13: Setting values of TRnCCR1 register (0000H to FFFFH)

2. The TORn0 and TORn1 pins become high level at timer count start.

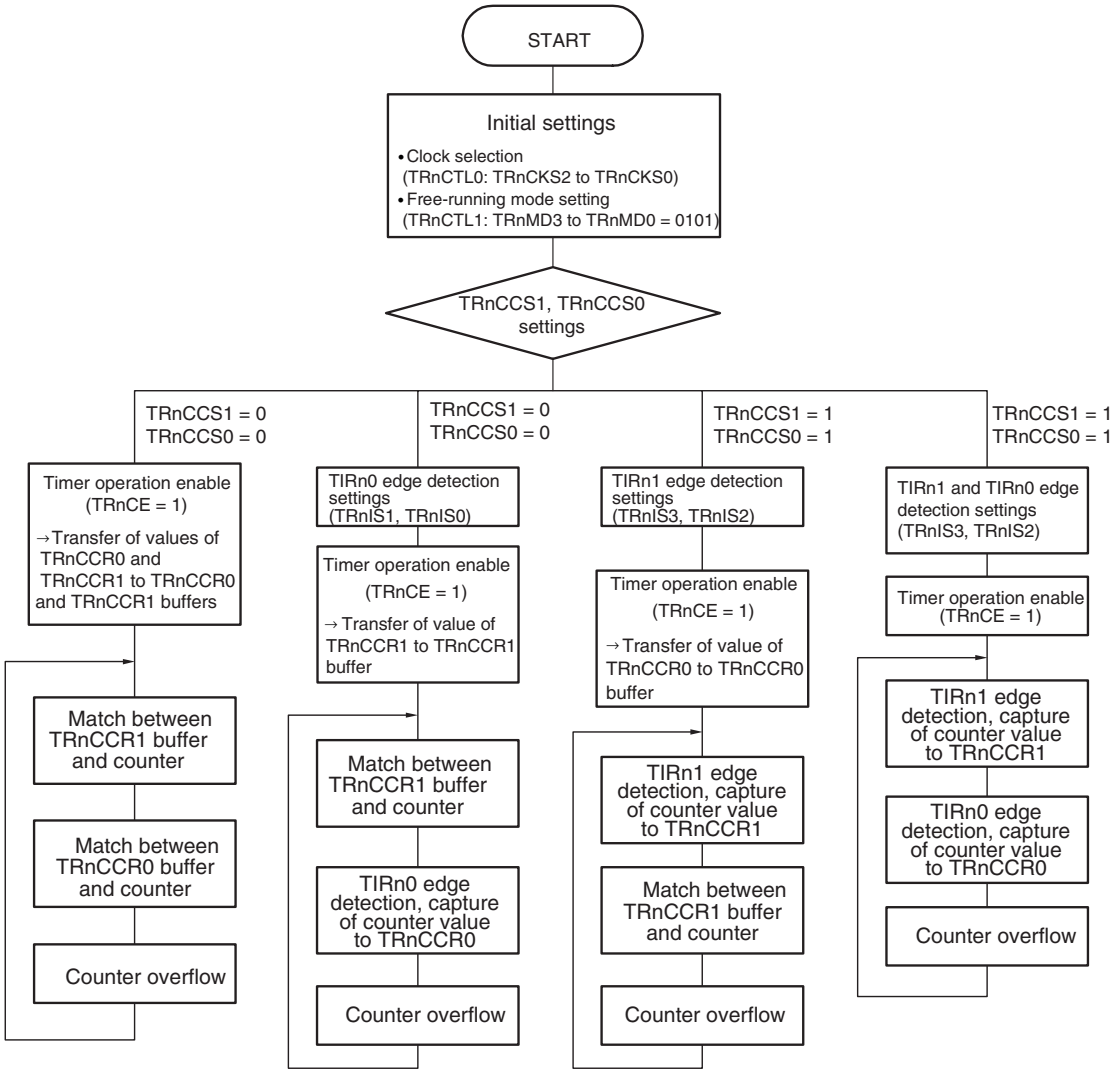
3. n = 0, 1

10.10.6 Free-running mode

(1) Outline of free-running mode

The operation timing of the free-running mode is shown below.
The operation for bits TRnCCS0 to TRnCCS3 of register TRnOPT0 is specified.

Figure 10-50: Basic Operation Flow in Free-Running Mode



Remarks: 1. This is an example when using the TRnCCR0 and TRnCCR1 registers. When using the TRnCCR2 and TRnCCR3 registers, the operation is controlled in the same manner via bits TRnCCS3 and TRnCCS2.

2. n = 0, 1

(2) Free-running mode operation list

(a) Compare registers

Register	Rewrite Method	Rewrite during Operation	Function
TRnCCR0	Anytime rewrite	Possible	Compare value
TRnCCR1 to TRnCCR3	Anytime rewrite	Possible	Compare value
TRnCCR4, TRnCCR5	Anytime rewrite	Possible	Compare value

(b) Input pins

Pin	Function
TIR1m	- (m = 0 to 3)
TTRGR1	-
TEVTR1	-

(c) Output pins

Pin	Function
TORnm	Toggle output upon TRnCCRm register compare match (m = 0 to 5)
TORn6, TORn7	-

(d) Interrupts

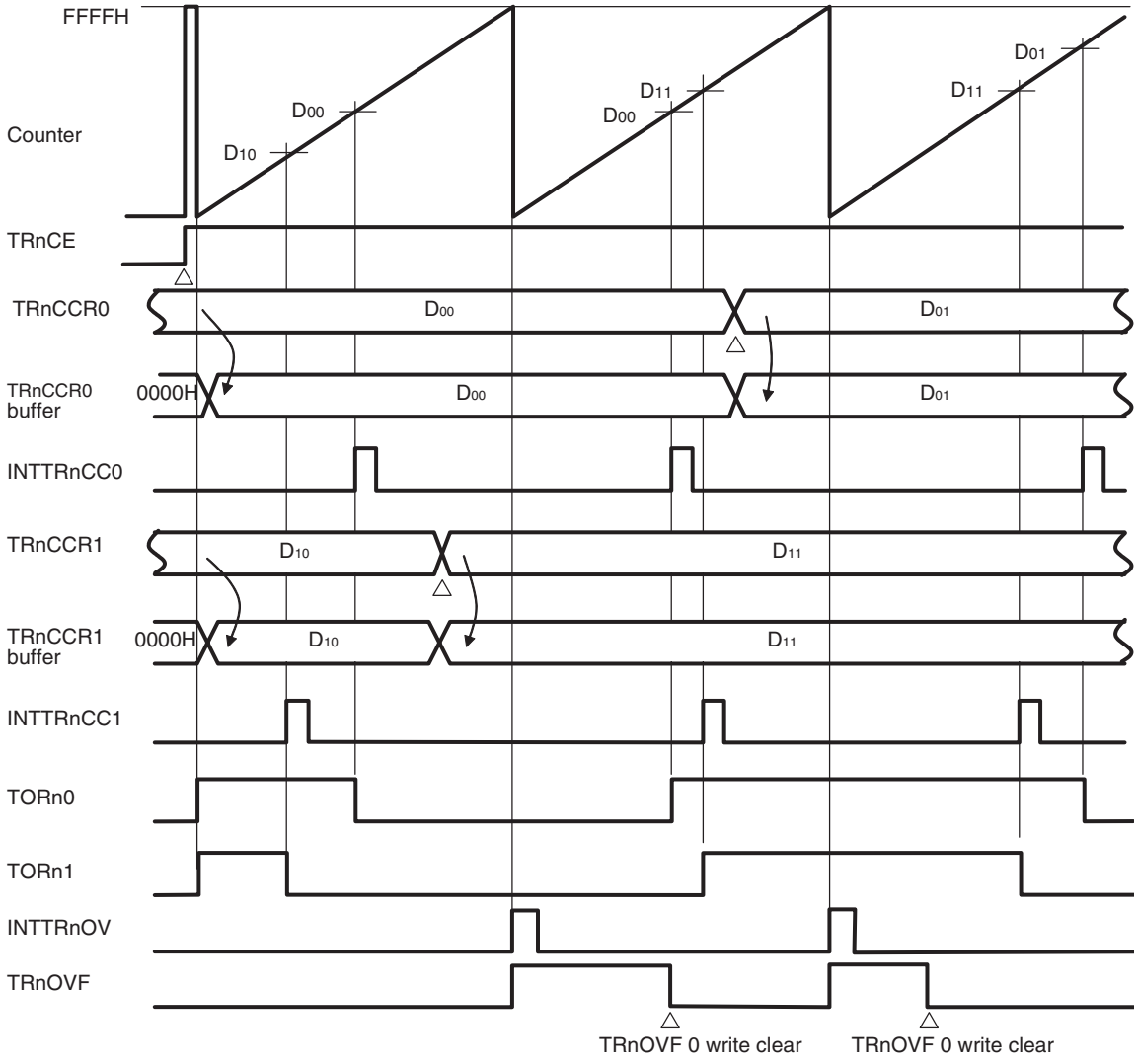
Interrupt	Function
INTTRnCCm	TRnCCRm register compare match (m = 0 to 5)
INTTRnOV	Overflow
INTTRnER	-

(3) Compare function (TRnCCS1 = 0, TRnCCS0 = 0)

When TRnCTL0 register bit TRnCE is set to 1, the counter counts from 0000H to FFFFH. An overflow interrupt (INTTRnOV) is output when the counter value changes from FFFFH to 0000H, and the counter is cleared. The count operation is performed in the free-running mode until TRnCE = 0 is set. Moreover, during count operation, a compare match interrupt (INTTRnCC0) is output upon a match between the counter and TRnCCR0 buffer register, and a compare match interrupt (INTTRnCC1) is output upon a match between the counter and TRnCCR1 buffer register.
The TRnCCR0 and TRnCCR1 registers can be rewritten using the anytime write method, regardless of the value of the TRnCE bit.
The TORn0 and TORn1 pins are toggle output controlled when bits register TRnOE0 and TRnOE1 of the TRnIOC0 register are set to 1.

Figure 10-51: Basic Operation Timing in Free-Running Mode (Compare Function)

When values of TRnCCR0 and TRnCCR1 registers are rewritten, TORn0, TORn1 are output (TRnOE0, 1 = 1, TRnOL0, 1 = 0)



Remarks: 1. D00, D01: Setting values of TRnCCR0 register (0000H to FFFFH)
D10, D11: Setting values of TRnCCR1 register (0000H to FFFFH)

2. TORn0 (toggle) width = (setting value of TRnCCR0 register + 1) × (count clock cycle)

3. TORn1 (toggle) width = (setting value of TRnCCR1 register + 1) × (count clock cycle)

4. Pins TORn0 and TORn1 become high level at count start.

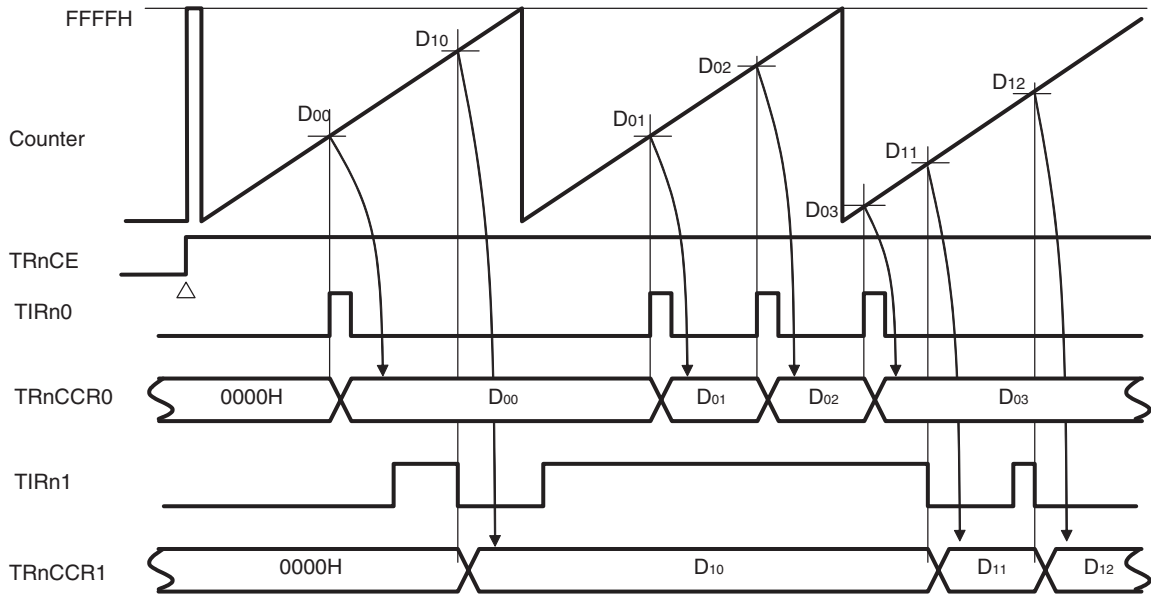
5. n = 0, 1

(4) Capture function (TRnCCS1 = 1, TRnCCS0 = 1)

When TRnCTL0 register bit TRnCE is set to 1, the counter counts from 0000H to FFFFH. An overflow interrupt (INTTRnOV) is output when the value of the counter changes from FFFFH to 0000H, and the counter is cleared. The count operation is performed in the free-running mode until TRnCE = 0 is set. When, during count operation, the counter value is captured to the TRnCCR0 and TRnCCR1 registers through detection of the valid edge of capture input (TIRn1, TIRn0), a capture interrupt (INTTRnCC0, INTTRnCC1) is output.
Regarding capture in the vicinity of overflow (FFFFH), judgment is possible with the overflow flag (TRnOVF). However, judgment with the TRnOVF flag is not possible when the capture trigger interval is such that it includes two overflow occurrences (2 or more free-running cycles).

Figure 10-52: Basic Operation Timing in Free-Running Mode (Capture Function)

When TORn0, TORn1 are not output (TRnOE0, 1 = 0, TRnOL0, 1 = 0)



Remarks: 1. D00, D01: Values captured to TRnCCR0 register (0000H to FFFFH)
D10, D11: Values captured to TRnCCR1 register (0000H to FFFFH)

2. TIRn0: Setting to rising edge detection (TRnIOC1 register bits TRnIS1, TRnIS0 = 01)
TIRn1: Setting to falling edge detection (TRnIOC1 register bits TRnIS3, TRnIS2 = 10)

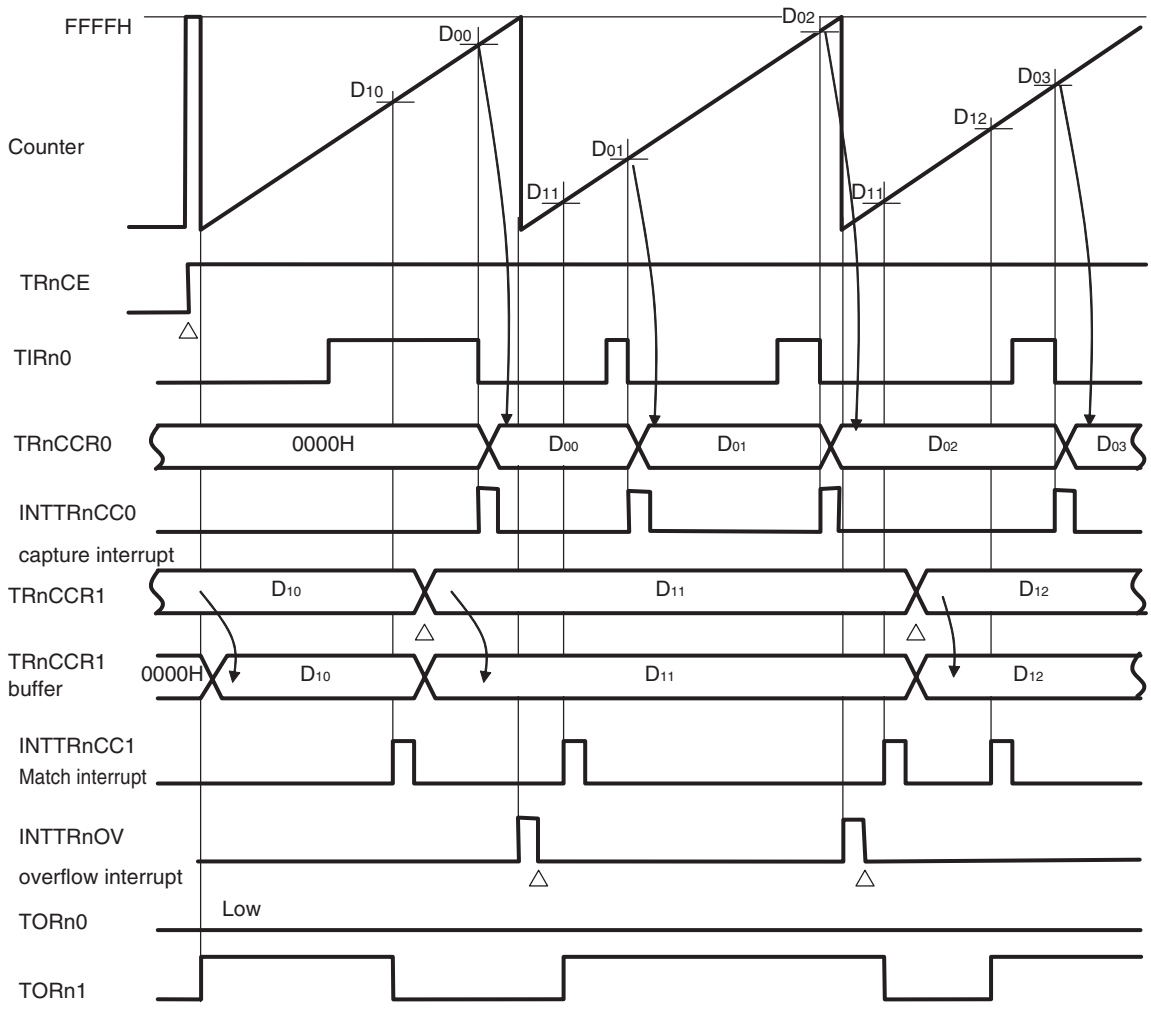
3. n = 0, 1

(5) Compare/capture function (TRnCCS1 = 0, TRnCCS0 = 1)

When TRnCTL0 register bit TRnCE is set to 1, the counter counts from 0000H to FFFFH, an overflow interrupt (INTTRnOV) is output when the value of the counter changes from FFFFH to 0000H, and the counter is cleared. The count operation is performed in the free-running mode until TRnCE = 0 is set. The TRnCCR1 register is used as a compare register, and as the interval function upon a match between the counter and TRnCCR1 register, a compare match interrupt (INTTRnCC1) is output. Since the TRnCCR0 register is set to the capture function, the TORn0 pin cannot be controlled even when TRnIOC0 register bit TRnOE0 is set to 1.

Figure 10-53: Basic Operation Timing in Free-Running Mode (Compare/Capture Function)

When value of TRnCCR1 is rewritten, TORn0, TORn1 are output (TRnOE0, 1 = 1, TRnOL0, 1 = 0)



Remarks: 1. D00, D01: Setting values of TRnCCR1 register (0000H to FFFFH)
D10, D11, D12, D13, D14, D15: Values captured to TRnCCR0 register (0000H to FFFFH)

2. TIRn0: Setting to rising edge detection (TRnIOC1 register bits TtnIS1, TtnIS0 = 11)

3. n = 0, 1

(6) Overflow flag

When, in the free-running mode, the counter overflows from FFFFH to 0000H, the overflow flag (TRnOVF) is set to “1”, and an overflow interrupt (INTTRnOV) is output.
The overflow flag is cleared through 0 write from the CPU.
(The overflow flag is not cleared by just being read.)

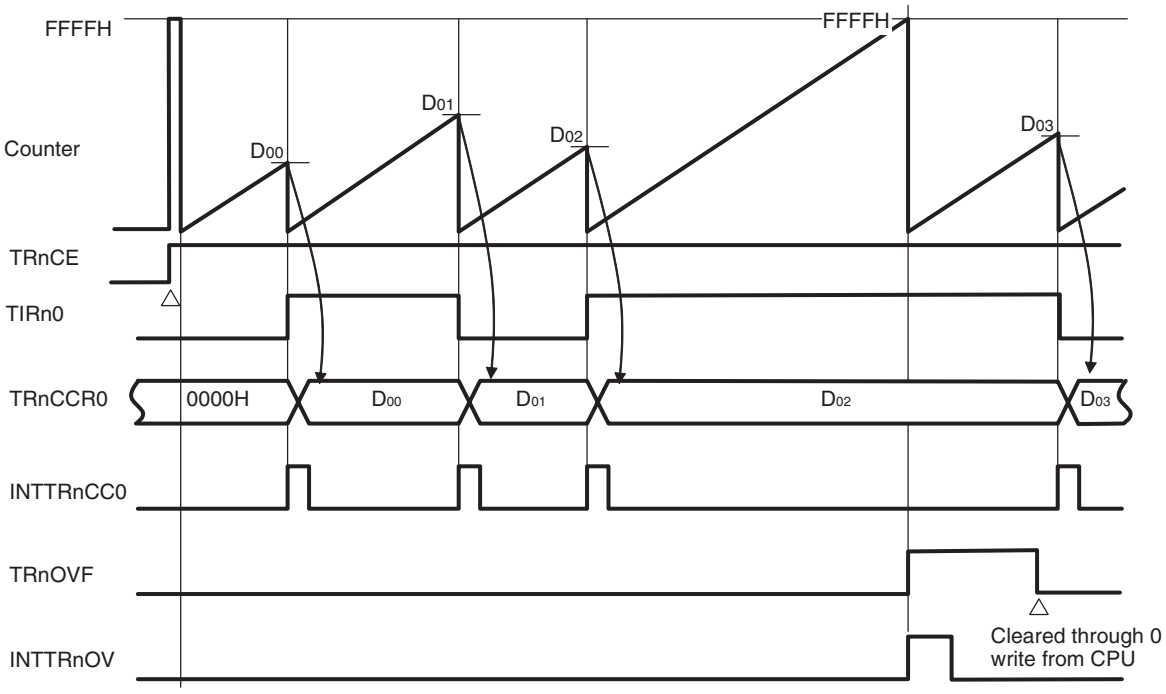
10.10.7 Pulse width measurement mode

(1) Outline of pulse width measurement mode

In the pulse width measurement mode, counting is performed in the free-running mode. The counter value is saved to the TRnCCR0 register, and the counter is cleared to 0000H. As a result, the external input pulse width can be measured. However, when measuring a long pulse width that exceeds counter overflow, perform judgment with the overflow flag. Measurement of pulses during which overflow occurs twice or more is not possible, so adjust the counter's operating frequency. Even in the case of TIRn1 to TIRn3 pin edge detection, pulse width measurement can be similarly performed by using the TRnCCR1 to TRnCCR3 registers.

Figure 10-54: Basic Operation Timing in Pulse Width Measurement Mode

(TRnOE0, 1 = 0, TRnOL0, 1 = 0)



Remarks: **1.** D00, D01, D02, D03: Values captured to TRnCCR0 register (0000H to FFFFH)

2. TIRn0: Setting to rising edge/falling edge (both edges) detection (TRnIOC1 register bits TtnIS1, TtnIS0 = 11)

3. n = 1

(2) Pulse width measurement mode operation list

(a) Compare register

Register	Rewrite Method	Rewrite during Operation	Function
TRnCCR0	Anytime rewrite	Possible	Compare value
TRnCCR1 to TRnCCR3	Anytime rewrite	Possible	Compare value
TRnCCR4, TRnCCR5	Anytime rewrite	Possible	Compare value

(b) Input pins

Pin	Function
TIR1m	Input capture trigger, transfer counter value to TR1CCRm register (m = 0 to 3)
TTRGR1	-
TEVTR1	-

(c) Output pins

Pin	Function
TORn0 to TORn5	-
TORn6, TORn7	-

(d) Interrupts

Interrupt	Function
INTTRnCCm	TIR1m capture (m = 0 to 3)
INTTRnCC4, INTTRnCC5	-
INTTRnOV	Overflow
INTTRnER	-

10.10.8 Triangular wave PWM mode

(1) Outline of triangular wave PWM mode

In the triangular wave PWM mode, similarly to in the PWM mode, when the duty is set to the TRnCCR1 to TRnCCR5 registers, the cycle is set to the TRnCCR0 register, and TRnCE = 1 is set, variable duty and cycle type triangular wave PWM output is performed from pins TORn1 to TORn5. The TORn0 pin is toggle output upon a match with the TRnCCR0 buffer register and upon counter underflow. Upon a match between the counter and TRnCCR0 register during count operation, compare match interrupts (INTTRnCC0 to INTTRnCC5) are output, and upon a match between the counter and TRnCCR1 to TRnCCR5 registers, a compare match interrupt (INTTRnCC1) is output. Moreover, upon counter underflow, an overflow interrupt (INTTRnOV) is output.

The TRnCCR0 to TRnCCR5 registers can be rewritten during count operation. Compare register reload occurs upon a match between the counter value and the TRnCCR0 buffer register.

However, since the next reload timing becomes valid when the TRnCCR1 register is written to, write the same value to the TRnCCR1 register even when wishing to rewrite only the value of the TRnCCR0 register. Reloading is not performed if only the TRnCCR0 register is rewritten. The reload timing is the underflow timing.

In the triangular wave PWM mode, the TRnCCR0 to TRnCCR3 registers have their function fixed as compare registers, so the capture function cannot be used.

Remark: In the triangular wave PWM mode, set the TRnCCR0 register to a value of $0 \leq$ TRnCCR0 $\leq$ FFFEH.

(2) Triangular wave PWM mode operation list

(a) Compare registers

Register	Rewrite Method	Rewrite during Operation	Function
TRnCCR0	Reload	Possible	1/2 of cycle
TRnCCR1 to TRnCCR3	Reload	Possible	1/2 of duty
TRnCCR4, TRnCCR5	Reload	Possible	1/2 of duty

(b) Input pins

Pin	Function
TIR1m	- (m = 0 to 3)
TTRGR1	-
TEVTR1	-

(c) Output pins

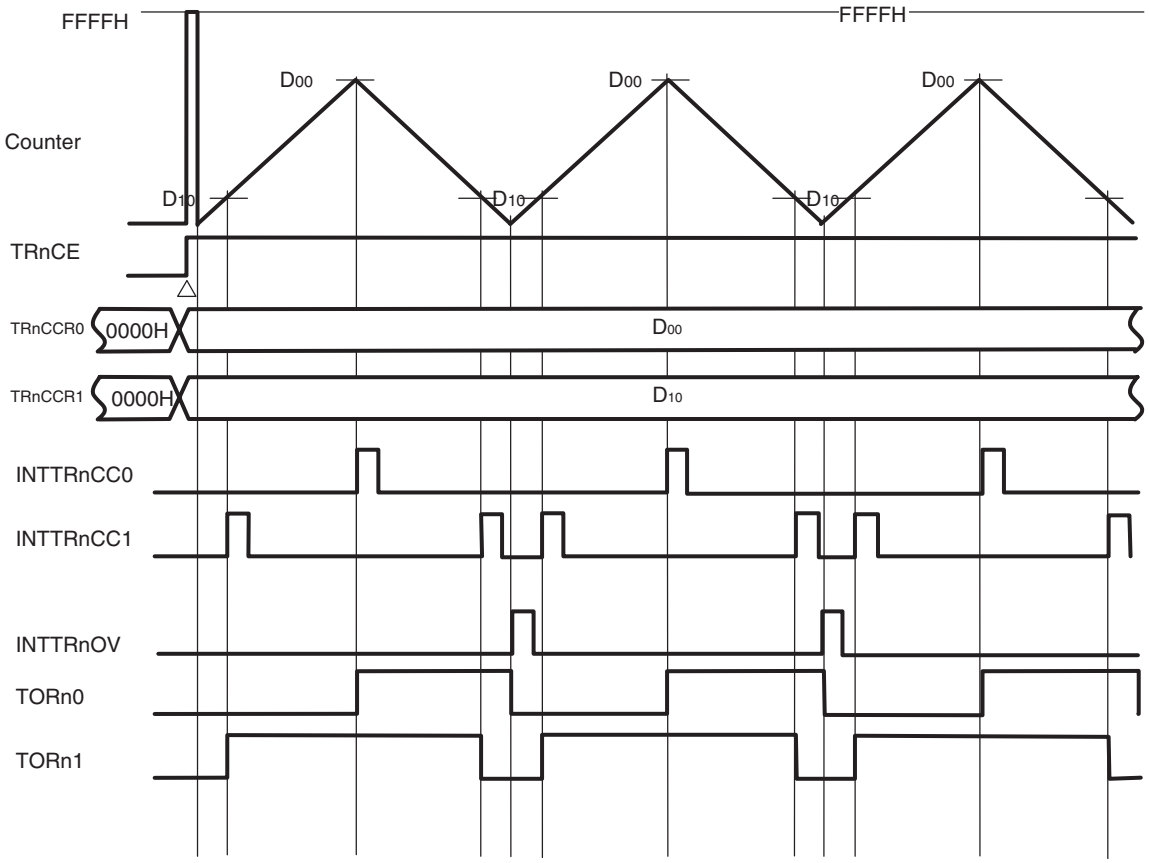
Pin	Function
TORn0	Inactive during counter up count, active during down count
TORnm	PWM output upon TRnCCRm register compare match (m = 0 to 5)
TORn6	-
TORn7	Pulse output through A/D conversion trigger

(d) Interrupts

Interrupt	Function
INTTRnCCm	TRnCCRm register compare match (m = 0 to 5)
INTTRnOV	-
INTTRnER	Error

Figure 10-55: Basic Operation Timing in Triangular Wave PWM Mode

When TORn0, TORn1 are output (TRnOE0, 1 = 1, TRnOL0, 1 = 0)



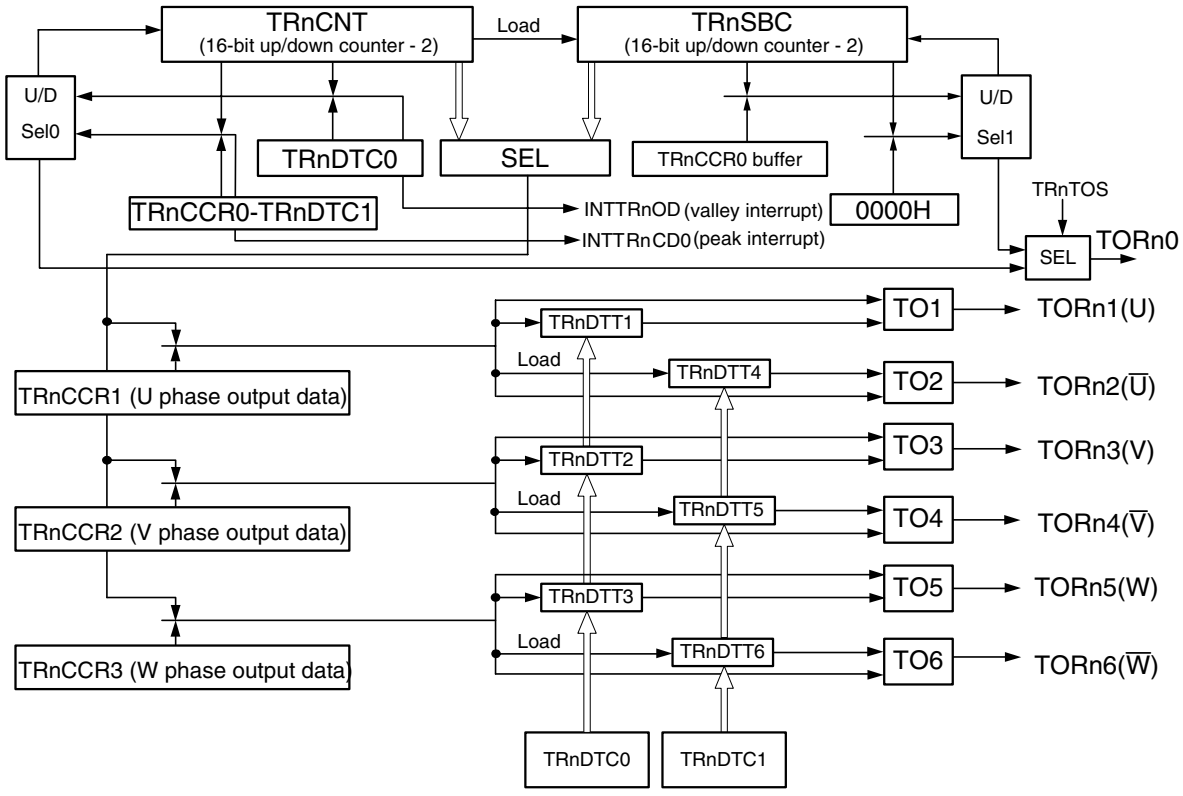
Remark: n = 0, 1

10.10.9 High-accuracy T-PWM mode

(1) Outline of high-accuracy T-PWM mode

The high-accuracy T-PWM mode generates 6-phase PWM using four 16-bit counters (up/down, ±2 counts, 15 real bits) and 16-bit compare registers (LSB = additional pulse control).
The carrier wave cycle calculated with “TRnCCR0-TRnDTC0-TRnDTC1” is set to the TRnCCR0 register. The duty of the U phase, V phase, and W phase voltage data signal is set with the TRnCCR1 to TRnCCR3 registers. The dead time is set with the TRnDTC0 and TRnDTC1 registers, and the TRnDTC0 register can be used to set the inverted phase (OFF) →normal phase (ON) dead time, while the TRnDTC1 register can be used to set the normal phase (OFF) →inverted phase (ON) dead time.
The counter operation consists in performing up count with the TRnDTC0 register value as the minimum value, and upon a match with the maximum value indicated by “TRnCCR0-TRnDTC1”, performing down count.
The 10-bit counters for dead time generation (TRnDTT1 to TRnDTT3) load the setting values of the TRnDTC0 and TRnDTC1 registers upon a match between the counter and the TRnCCR1 to TRnCCR3 registers, and perform down-count.
Upon a match between the 16-bit counter and the TRnCCR1 to TRnCCR3 registers, INTTRnCC1 to INTTRnCC3, which are used as the respective compare match interrupt signals, are output. (In the 0% output vicinity and 100% output vicinity, no interrupt signal may be output.)

Figure 10-56: High-Accuracy T-PWM Mode Block Diagram



(2) High-accuracy T-PWM mode operation list

(a) Compare registers

Register	Rewrite Method	Rewrite during Operation	Function
TRnCCR0	Reload, Anytime rewrite	Possible	Cycle
TRnCCR1 to TRnCCR3	Reload, Anytime rewrite	Possible	PWM duty
TRnCCR4, TRnCCR5	Reload, Anytime rewrite	Possible	PWM duty (selectable as A/D conversion trigger)

(b) Input pins

Pin	Function
TIR1m	-(m = 0 to 3)
TTRGR1	-
TEVTR1	-

(c) Output Pins

Pin	Function
TORn0	Inactive during counter or sub-counter up count, active during down count
TORn1	PWM output upon TRnCCR1 compare match (with dead time)
TORn2	Inverted output to TORn1
TORn3	PWM output upon TRnCCR2 compare match (with dead time)
TORn4	Inverted output to TORn3
TORn5	PWM output upon TRnCCR3 compare match (with dead time)
TORn6	Inverted output to TORn5
TORn7	Pulse output through A/D conversion trigger

(d) Interrupts

Interrupt	Function
INTTRnCCR0	INTTRnCCR0 compare match[Note 1]
INTTRnCC1 to INTTRnCC5	TRnCCR1 to TRnCCR5 compare match
INTTRnOV	Overflow[Note 2]
INTTRnER	Error
INTTRnOD	Through interrupt
INTTRnCD	Peak interrupt

Notes: 1. Only when TRnDTC1 = 000H

2. When TRnCCR0, TRnDTC0, and TRnDTC1 registers are incorrectly set.

(3) High-accuracy T-PWM mode settings

(a) Mode settings

The high-accuracy T-PWM mode is selected by setting TRnCTL1 register bits TRnMD4 to 0 = 1000B.

(b) Output level/output enable settings

Set bits TRnOL0-TRnOL7 and TRnOE0-TRnOE7 of the TRnIOC0, TRnIOC3 registers, to enable output level/output enable.
Pin TORn0 indicates the counter's and sub-counter's up count/down count status. The counter/sub-counter can be switched with TRnOPT7 register bit TRnTOR.
Pin TORn7 is the external A/D conversion output pin. Set this pin as required.

(c) Error interrupt output enable

Set error interrupt output enable upon detection of normal phase/inverted phase simultaneous active. Error interrupt output is enabled by setting TRnIOC4 register bit TRnEOC to "1". In the high-accuracy T-PWM mode, when the dead time setting is other than "000H", the error interrupt (INTTRnER) never goes active, regardless of which value the TRnCCR0 to TRnCCR3 registers are set. However, an error may be detected upon the occurrence of a timer Rn internal circuit fault. If the dead time setting is "000H", a glitch may occur upon occurrence of an error interrupt (INTTRnER) at the normal phase and inverted phase switch timing.

(d) Rewrite timing for registers with reload function

Batch rewrite/anytime rewrite can be set for registers with the reload function. This setting is performed with TRnOPT0 register bit RnCMS. (The default is "0" batch rewrite). To perform batch rewrite, be sure to set TRnOPT1 register bit TRnICE or TRnIOE.
(If bit TRnICE and bit TRnIOE are both "0", the reload timing does not occur.)
If anytime rewrite is selected, unintended output may occur depending on the rewrite timing.
(When using the anytime rewrite function, refer to cautions **(a)** to **(c)** in **10.4.2 (1) Anytime rewrite**.)

(e) Interrupt and thinning out function settings

The interrupt and thinning out function settings are performed with the TRnOPT1 register. If a peak interrupt (INTTRnCD) is required, set bit TRnICE to 1. If a valley interrupt (INTTRnOD) is required, set bit TRnIOE to 1. To use the thinning out function for peak/valley interrupts, perform settings with the TRnID4 to TRnID0 registers.

(f) Reload thinning out function setting

To set the reload timing to the same timing as the interrupt timing, set TRnOPT1 register bit TRnRDE to 1.

(g) A/D conversion trigger output settings

To set A/D conversion trigger 0 (TRnADTRG0 signal), set TRnOPT2 register bits TRnAT05 to TRnAT00.
With bits TRnAT05 to TRnAT00, peak interrupt (INTTRnCD) and valley interrupt (INTTRnOD) enable/disable is performed at the TRnCCR5 register match timing (counter up count/down count), and the TRnCCR4 register match timing (counter up count/down count).
To set A/D conversion trigger 1 (TRnADTRG1 signal), set TRnOPT3 register bits TRnAT15 to TRnAT10.
With bits TRnAT15 to TRnAT10, peak interrupt (INTTRnCD) and valley interrupt (INTTRnOD) enable/disable is performed at the TRnCCR5 register match timing (counter up count/down count), and TRnCCR4 register match timing (counter up count/down count). Set the TRnCCR4 and TRnCCR5 registers' compare values.
For the TRnADTRG0 and TRnADTRG1 signals, also perform the thinning out function setting.

Caution: To use the TORn7 pin, correctly perform the TRnOPT2 and TRnOPT3 register and the TRnCCR4 and TRnCCR5 register settings.

(h) Dead time settings

The dead time settings are performed with the TRnDTC0 and TRnDTC1 registers.
The dead time can be obtained with counter operation clock cycle (P) × TRnDTC0, TRnDTC1.
The time until TORn2,TORn4,TORn6 pin inactive change →TORn1,TORn3,TORn5 pin active change can be set with the TRnDTC0 register.
The time until TORn1,TORn3,TORn5 pin inactive change →TORn2,TORn4,TORn6 pin active change can be set with the TRnDTC1 register.

(i) Carrier wave cycle

For the carrier wave cycle, set the TRnCCR0 register using the following equation.

TRnCCR0 = (carrier wave cycle/ counter operation clock cycle) + TRnDTC1 + TRnDTC0

For the setting value of the TRnCCR0 register, meet the following conditions keeping in mind the dead time.

TRnCCR0 > 3 × MAX (TRnDTC0, TRnDTC1) + MIN (TRnDTC0, TRnDTC1)
TRnCCR0 ≤FFFEH

(MAX(A,B) indicates the larger value of A and B, and MIN(A,B) indicates the smaller value of A and B.)

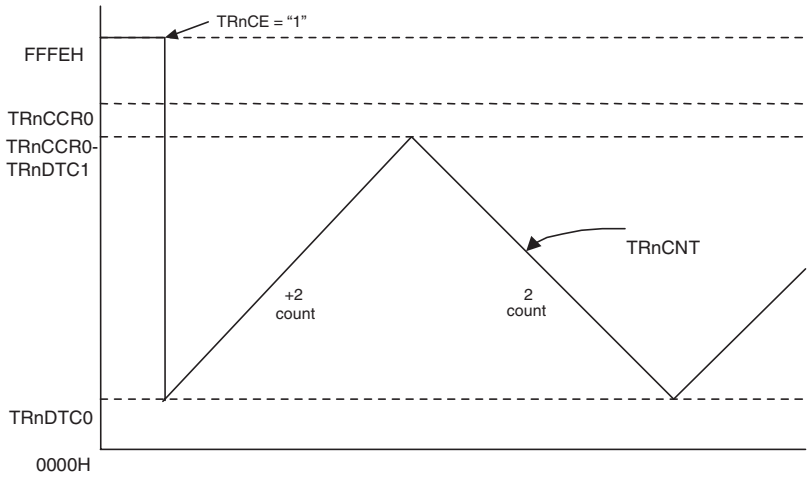
(j) Duty (PWM width) setting

For the duty setting, perform the U phase, V phase, and W phase settings with the TRnCCR1 to TRnCCR3 registers. The setting range of the TRnCCR1 to TRnCCR3 registers is 0000H ≤ TRnCCR1, TRnCCR2, TRnCCR3 ≤TRnCCR0 + 1. Do not set TRnCCR0 + 2 < TRnCCR1, TRnCCR2, TRnCCR3.
LSB (Least Significant Bit) of the TRnCCR1 to TRnCCR3 registers means the additional pulse setting. For example, if TRnCCR1 = 0003H is set, compare to when TRnCCR1 = 0002H is set, the inverted phase (pin TORn2) change is an 1-count clock delay (during counter up count).

(4) Counter operation in high-accuracy T-PWM mode

At initial value FFFEH, the TRnDTC0 value is loaded to the counter immediately after TRnCE = 1 is set, and the counter counts up in +2 steps. Then, upon a match with TRnCCR0 to TRnDTC1, the counter counts down in -2 steps. The counter operation is as follows.

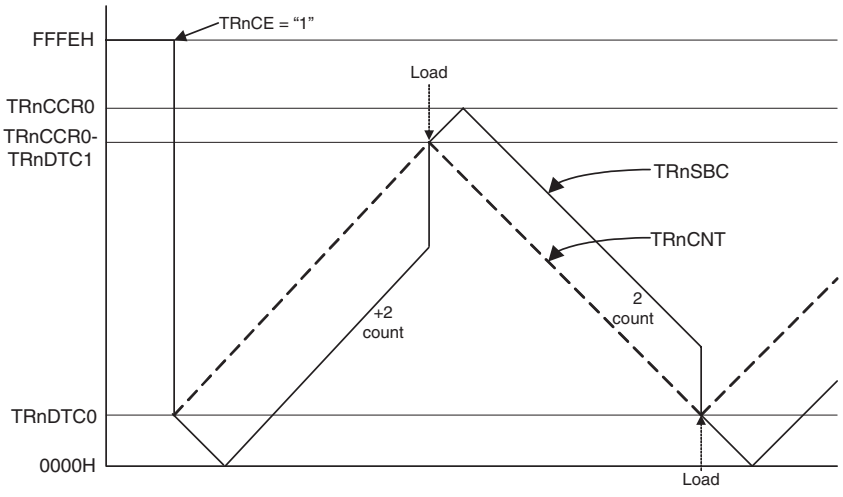
Figure 10-57: Counter Operation in High-Accuracy T-PWM Mode



Remark: Minimum counter value: TRnDTC0
Maximum counter value: TRnCCR0 - TRnDTC1
Carrier wave cycle: (TRnCCR0-TRnDTC0-TRnDTC1) × count clock cycle

At initial value FFFEH, the value of TRnDTC0 register is loaded to the sub-counter immediately after TRnCE = 1 is set. Then, until a match with 0000H, the sub-counter counts down in -2 steps, and the counter value is loaded to the sub-counter at the counter's up count →down count switch timing. The TRnDTC0 register goes on counting up, and upon a match with the TRnCCR0 register, starts counting down in -2 steps. At the same time, upon a match between the counter and the TRnDTC0 register, the counter value is loaded and down count is continued.
The sub-counter operation is as follows.

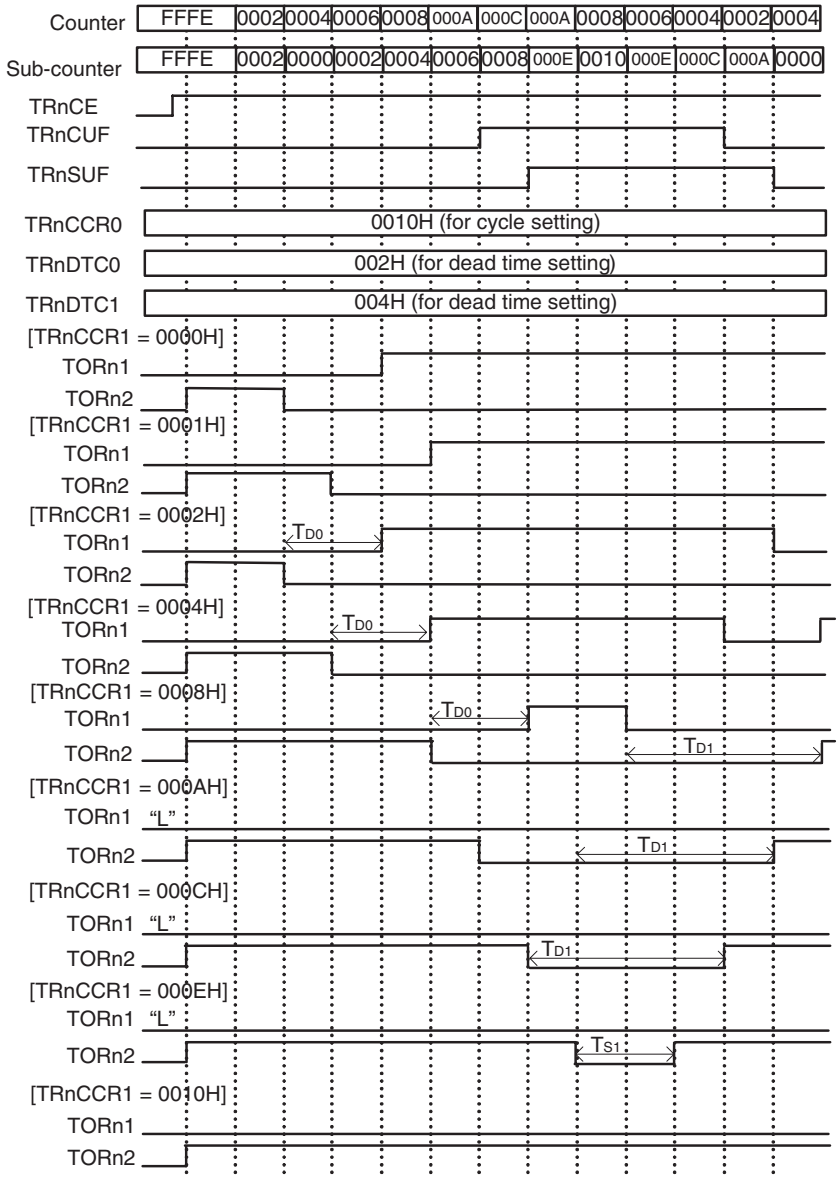
Figure 10-58: Sub-Counter Operation in High-Accuracy T-PWM Mode



(5) Basic operation in high-accuracy T-PWM mode

The Figure 10-59 shows the timing chart when TRnCCR0 = 0010H, TRnDTC0 = 0002H, TRnDTC1 = 0004H, and the TRnCCR1 register is set from 0000H to 0010H (one part only shown). In this example, TRnOL6 to TRnOL1 = 000000B is set.
If TRnCCR1 > TRnDTC0, pin TORn2 changes with the following compare match.
Since TRnCCR1 = (TRnDTC0-0001H) is an additional pulse, compared to when TRnCCR1 = (TRnDTC0 - 0002H), pin TORn2 changes with an 1 count clock delay.

Figure 10-59: Timer Output Example When TRnCE = 1 Is Set (Initial) (High-Accuracy T-PWM Mode)



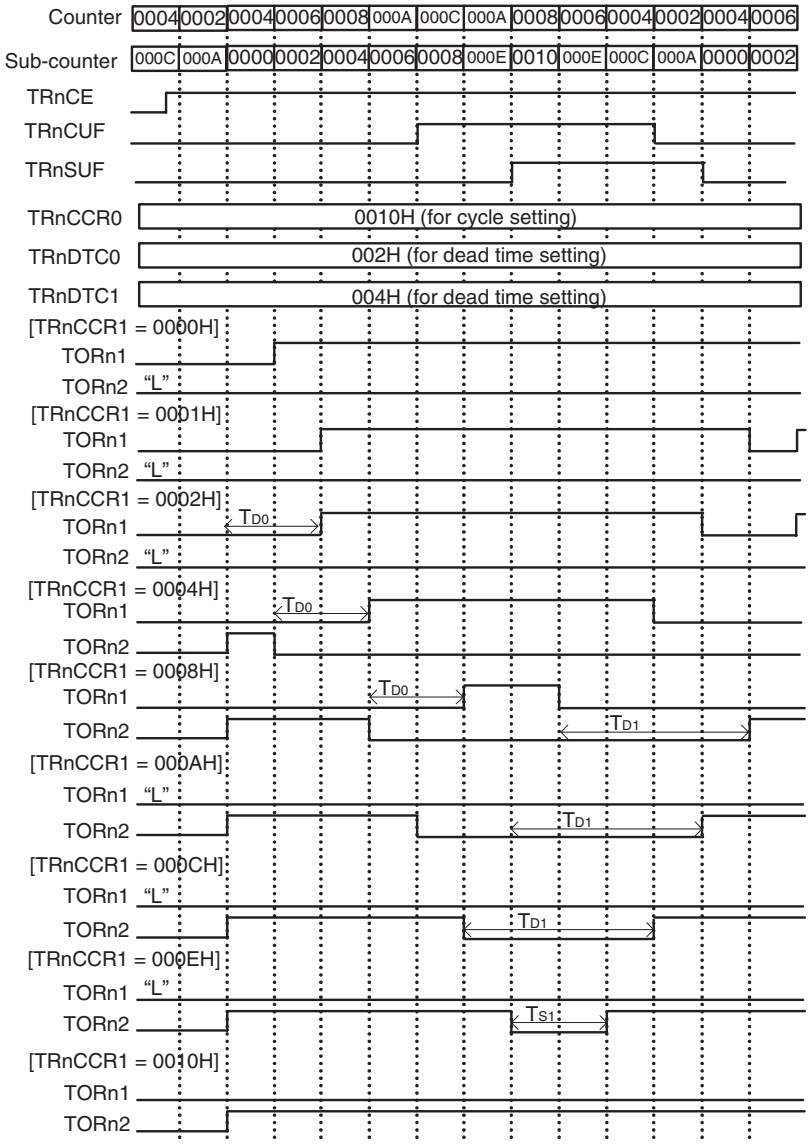
Remarks: 1. TRnCCR0 = 0010H, TRnDTC0 = 0002H, TRnDTC1 = 0004H

2. TD0: Time depends on dead time setting of TRnDTC0 register
TD1: Time depends on dead time setting of TRnDTC1 register
TS1: Time is determined through sub-counter compare, when sub-counter value > counter value

3. n = 0, 1

The Figure 10-60 shows the timing chart when TRnCCR0 = 0010H, TRnDTC0 = 0002H, TRnDTC1 = 0004H, and the TRnCCR1 register is set from 0000H to 0010H (one part only shown). In this example, TRnOL6 to TRnOL1 = 000000B is set.
As can be seen in this figure, a normal phase (pin TORn1) that is active (high level) is output when 0000H ≤TRnCCR1 ≤(TRnCCR0 - TRnDTC0 + 0001H).
Also, the inverted phase (pin TORn0) that is active (high level) is output when (TRnDTC0 + TRnDTC1) < TRnCCR1 ≤TRnCCR0.

Figure 10-60: Timer Output Example During Operation (High-Accuracy T-PWM Mode)



Remarks: 1. TRnCCR0 = 0010H, TRnDTC0 = 0002H, TRnDTC1 = 0004H

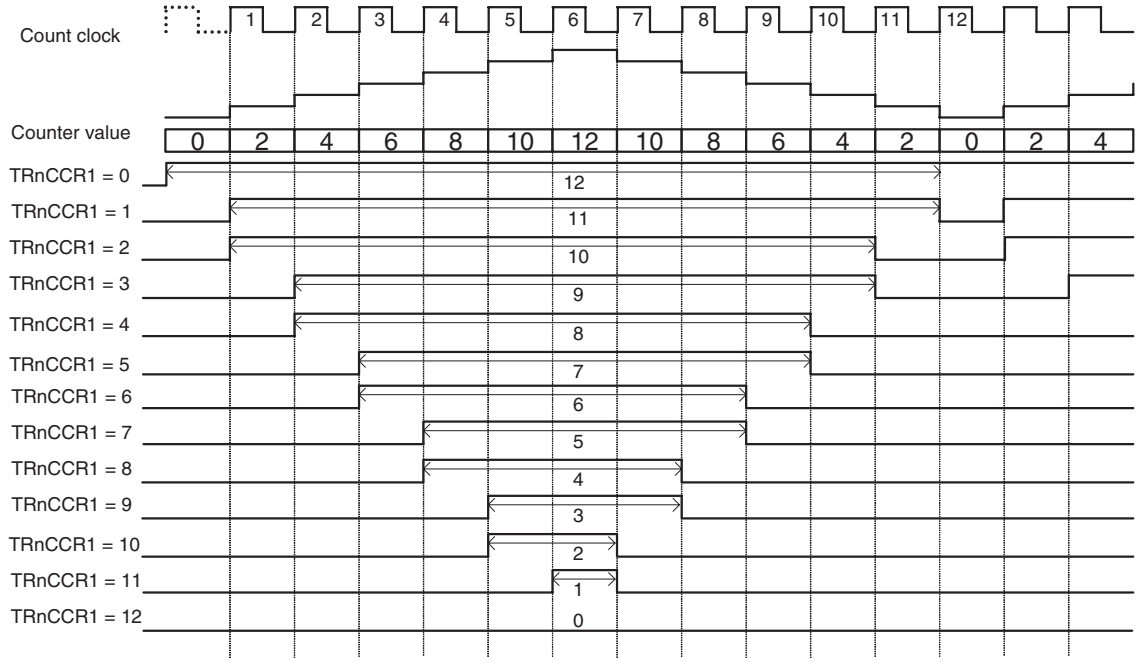
2. TD0: Time depends on dead time setting of TRnDTC0 register
TD1: Time depends on dead time setting of TRnDTC1 register
TS0: Time is determined through sub-counter compare, when sub-counter value < counter value
TS1: Time is determined through sub-counter compare, when sub-counter value > counter value

3. n = 0, 1

(6) Additional pulse control in high-accuracy T-PWM mode

In the high-accuracy T-PWM mode, additional pulse can be set by setting the LSB of the duty setting registers (TRnCCR1 to TRnCCR3) to “1”. With the additional pulse control function, finer duty control can be performed (higher accuracy).
TORn1 pin output examples are provided below for when additional pulse control is and is not performed. The settings used here are TRnCCR = 12, TRnDTC0, and TRnDTC1 = 0.

Figure 10-61: TORn1 Pin Output Example When Performing Additional Pulse Control

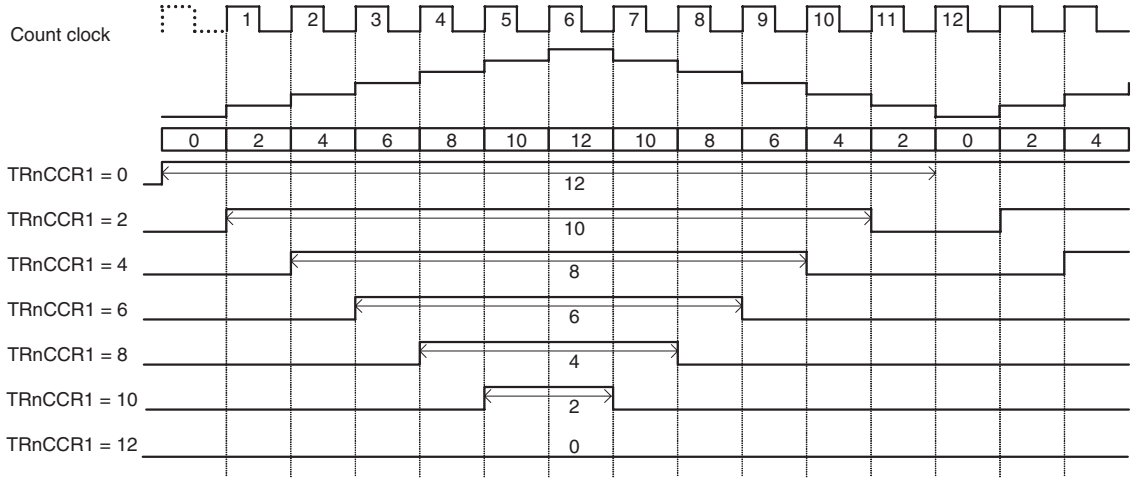


Remarks: 1. TRnCCR0 = 12, TRnDTC0 = 0, TRnDTC1 = 0

2. n = 0, 1

The locations where additional pulse control is performed are when an odd value has been set to the TRnCCR1 register.
In the above figure, the arrows and numbers indicate the duty width of the TORn1 pin output within 1 cycle.
As can be seen in the above figure, when additional pulse control is performed, the output width (duty ratio) of pin TORn1 can be controlled in 1 count clock steps from 12 clocks to 0 clocks.

Figure 10-62: TORn1 Pin Output Example When Additional Pulse Control Is Not Performed



Remarks: 1. TRnCCR0 = 12, TRnDTC0 = 0, TRnDTC1 = 0

2. n = 0, 1

The figure above is an example when additional pulse control is not performed.
In the above figure, the arrows and numbers indicate the duty width of the TORn1 pin output within 1 cycle.
When additional pulse control is not performed, the output width of pin TORn1 can be controlled in 2 count clock steps from 12 clocks to 0 clocks. In this case, the duty change amount is larger compared to when additional pulse control is performed.

(7) Caution on timer output in high-accuracy T-PWM mode

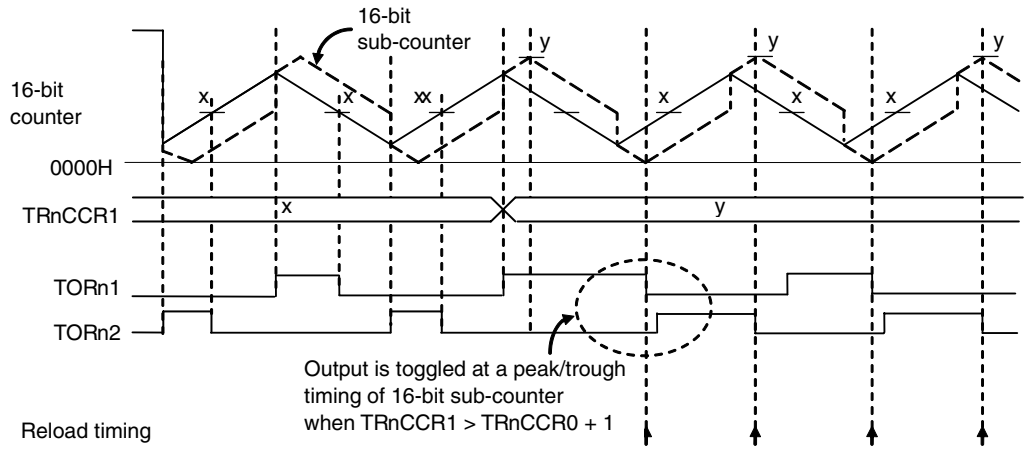
There are cautions for TRnCCR1 to TRnCCR3 as follows when varying 6-phase PWM duty by using reload (batch rewrite).

(a) In case of TRnCCR0 + 2 ≤TRnCCRm (Setting prohibited)

Figure 10-63a shows the case when the value of “TRnCCR0 + 2 or more” is set to the TRnCCR1 register. When the TRnCCR1 register setting is changed like this, a match between the 16-bit counter and TRnCCR1 register does not occur thereafter. Therefore, the TORn1 pin output level is forcibly changed to inactive level at the following 16-bit sub-counter trough timing. Output will be switched at 16-bit sub-counter peak/trough timing after that.

Figure 10-63: Timings of Timer Output in High-accuracy T-PWM mode (1/3)

(a) Output When TRnCCR0 + 2 ≤TRnCCR1



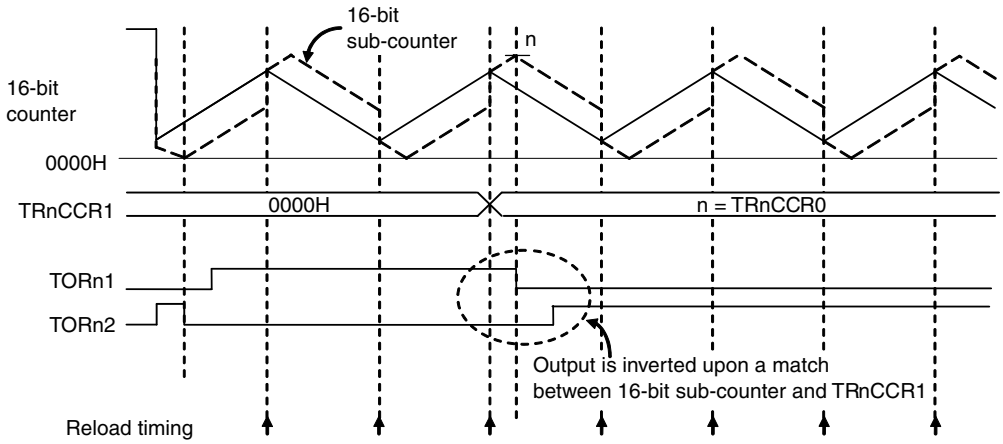
Note: m = 1 to 3

(b) In case of rewriting from TRnCCRm = 0000H to TRnCCRm = TRnCCR0

Figure 10-63b shows the output waveform where the TRnCCR1 register setting is changed from 100% output to 0% output. The TORn1 pin output is inverted upon a match between the TRnCCR1 register and 16-bit sub-counter, and the TORn2 pin output is inverted after the dead time count.

Figure 10-63: Timings of Timer Output in High-accuracy T-PWM mode (2/3)

(b) Output When Rewriting from TRnCCR1 = 0000H to TRnCCR1 = TRnCCR0

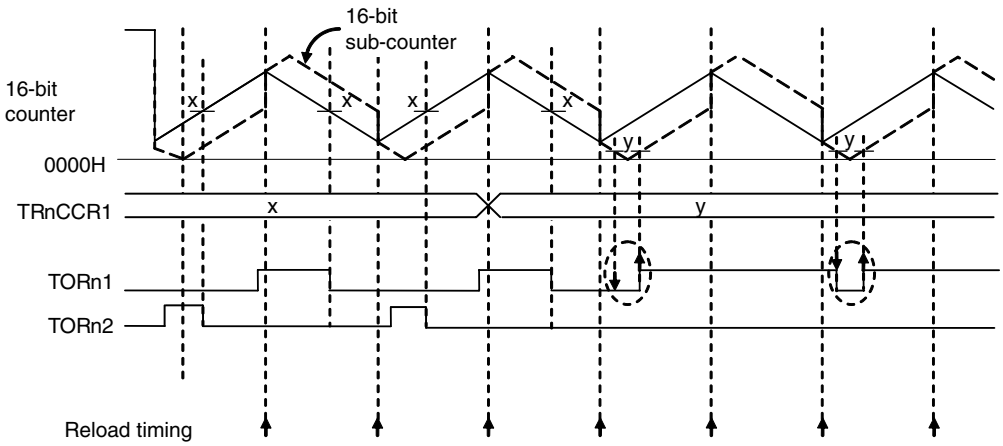


Note: m = 1 to 3

(c) In case of rewriting from "TRnDTC0 + TRnDTC1 < TRnCCRm < TRnCCR0 –TRnDTC0 – TRnDTC1" to "TRnCCRm < TRnDTC0 + TRnDTC1"

Figure 10-63c shows the output waveform when rewriting the TRnCCR1 register from x (TRnDTC0 + TRnDTC1 < x < TRnCCR0 – TRnDTC0 – TRnDTC1) to y (y < TRnDTC0 + TRnDTC1). In this case, the TORn1 pin output becomes active when the TORn1 pin set condition occurs upon a match between the 16-bit counter (or 16-bit sub-counter) and the TRnCCR1 register immediately after reload (batch rewrite).

(c) Output When Rewriting from TRnDTC0 + TRnDTC1 < TRnCCR1 < TRnCCR0 –TRnDTC0 –TRnDTC1 to TRnCCR1 < TRnDTC0 + TRnDTC1



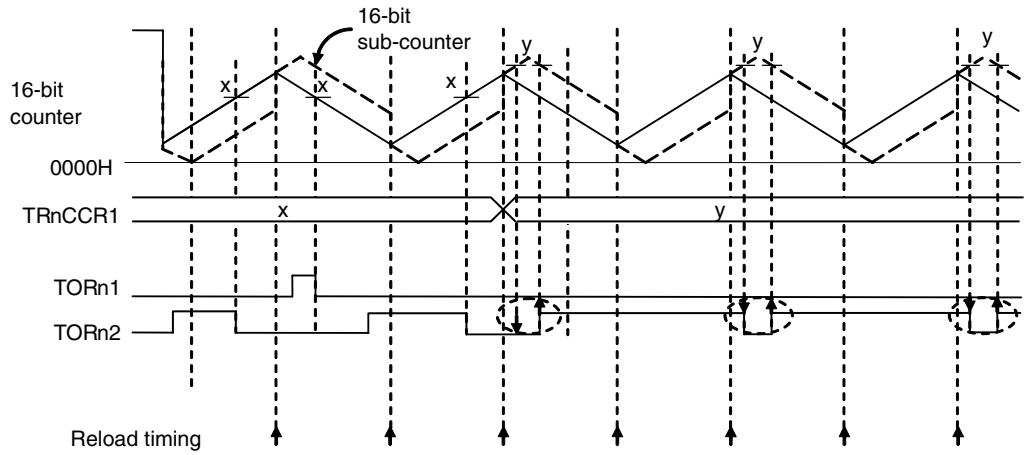
Note: m = 1 to 3

(d) In case of rewriting from "TRnDTC0 + TRnDTC1 < TRnCCRm < TRnCCR0 −TRnDTC0 − TRnDTC1" to "TRnCCR0 −TRnDTC1 + 1 < TRnCCRm < TRnCCR0"

Figure 10-63d shows the output waveform when rewriting the TRnCCR1 register from x (TRnDTC0 + TRnDTC1 < x < TRnCCR0 −TRnDTC0 −TRnDTC1) to y (TRnCCR0 −TRnDTC0 − TRnDTC1 < TRnDTC0 < TRnCCR0). In this case, the TORn2 pin output becomes inactive (high level) when the TORn2 pin set condition occurs upon a match between the 16-bit counter (or 16-bit sub-counter) and TRnCCRm register immediately after batch rewrite.

Figure 10-63: Timings of Timer Output in High-accuracy T-PWM mode (3/3)

(d) Output When Rewriting from "TRnDTC0 + TRnDTC1 < TRnCCR1 < TRnCCR0 −TRnDTC0 − TRnDTC1" to "TRnCCR0 −TRnDTC1 + 1 < TRnCCR1 < TRnCCR0"



Note: m = 1 to 3

(8) Timer output change after compare register updating

Timer output is affected when the compare register value is updated during reload execution. The timer output level is changed at any timing listed in Tables 10-1 and 10-2.

Table 10-1: Positive Phase Operation Condition List

Operation	Symbol	Condition
Set	ST1	Match between counting up near the 16-bit sub-counter trough and compare register values (< TRnDTC0)
Clear	RT1	Match between counting down near the 16-bit sub-counter trough and compare register values (< TRnDTC0)
Set	ST2	At completion of dead time counter (TRnDTC0) operation
Clear	RT2	When 16-bit counter value matches with compare register value during count-down operation
Set	ST3	100% output for PWM duty
Clear	RT3	When no match occurs until 16-bit sub-counter counts down to 0000H
Clear	RT4	TRnCCR0 and TRnDTC0 settings are changed at a reload timing. Though neither a match (nor a match interrupt) occurs between TRnCCR0 and TRnDTC0, the operation is cleared by special processing.
Clear	RT5	The operation is cleared upon a match between peripheral 16-bit sub-counter peak and compare register values in positive phase active level.

Table 10-2: Negative Phase Operation Condition List

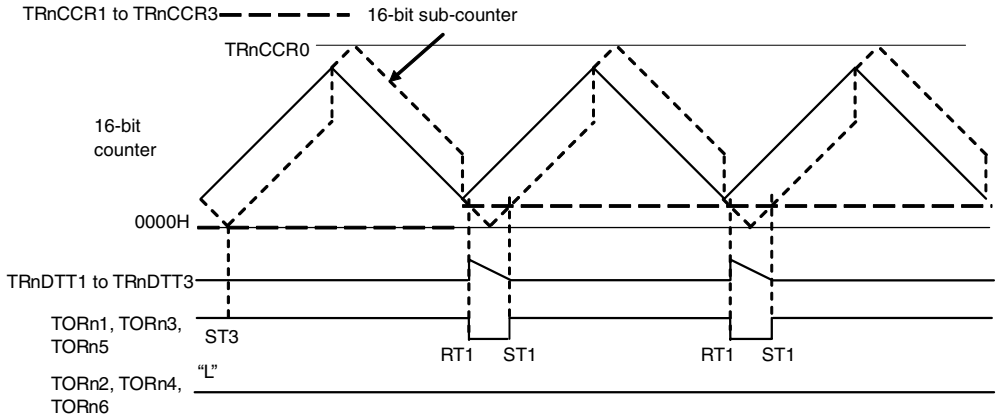
Operation	Symbol	Condition
Set	SB1	Match between counting down near the 16-bit sub-counter peak and compare register values (> TRnCCR0 - TRnDTC1)
Clear	RB1	Match between counting up near the 16-bit sub-counter peak and compare register values (> TRnCCR0 - TRnDTC1)
Set	SB2	At completion of dead time counter (TRnDTC1) operation
Clear	RB2	When 16-bit counter value matches with compare register value during count-up operation
Set	SB3	100% output for PWM duty
Clear	RB3	When no match occurs until 16-bit sub-counter counts up to TRnCCR0
Clear	RB4	TRnCCR0 and TRnDTC0 settings are changed at a reload timing. Though neither a match (nor a match interrupt) occurs between TRnCCR0 and TRnDTC1, the operation is cleared by special processing.
Clear	RB5	The operation is cleared upon a match between peripheral 16-bit sub-counter trough and compare register values in negative phase active level.

Table 10-3: Compare Register Value After Trough Reload (TRnDTC0 < TRnDTC1)

Compare Register Value Immediately Before Trough Reload	Compare Register Value After Trough Reload (TRnDTC0 < TRnDTC1)	Figure No.
0000H	0000H < TRnCCR1 to TRnCCR3 < TRnDTC0	Figureaa
	TRnCCR1 to TRnCCR3 = 0000H, TRnDTC0 + 1	Figureab
	TRnDTC0 + 1 < TRnCCR1 to TRnCCR3 ≤TRnDTC0 × 2	Figure 10-64c
	TRnDTC0 × 2 < TRnCCR1 to TRnCCR3 < TRnCCR0 −TRnDTC0 −TRnDTC1	Figure 10-64d
	TRnCCR0 −TRnDTC0 −TRnDTC1 ≤TRnCCR1 to TRnCCR3 < TRnCCR0 −TRnDTC1	Figure 10-64e
	TRnCCR0 −TRnDTC1 ≤TRnCCR1 to TRnCCR3 < TRnCCR0	Figure 10-64f
	TRnCCR1 to TRnCCR3 = TRnCCR0	Figure 10-64g

Figure 10-64: Timer Output Change after Compare Register Updating Timings (1/3)

(a) TRnCCR1 to TRnCCR3 = 0000H →0000H < TRnCCR1 to TRnCCR3 < TRnDTC0



(b) TRnCCR1 to TRnCCR3 = 0000H →TRnCCR1 to TRnCCR3 = TRnDTC0, TRnDTC0 + 1

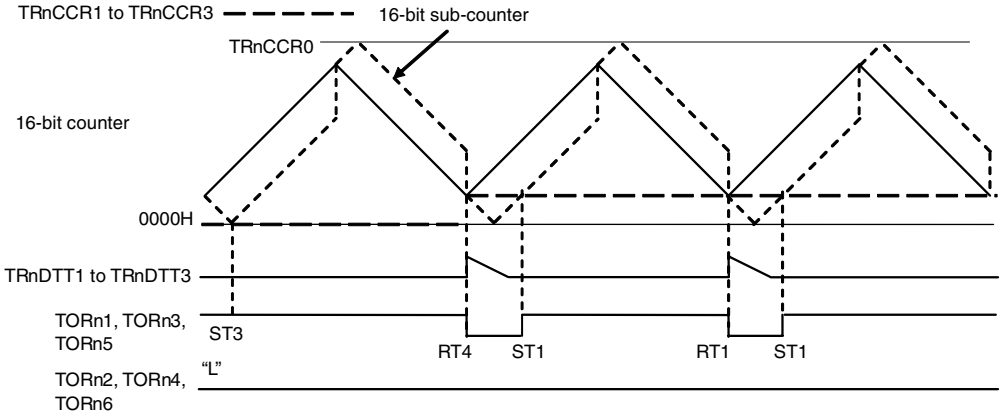
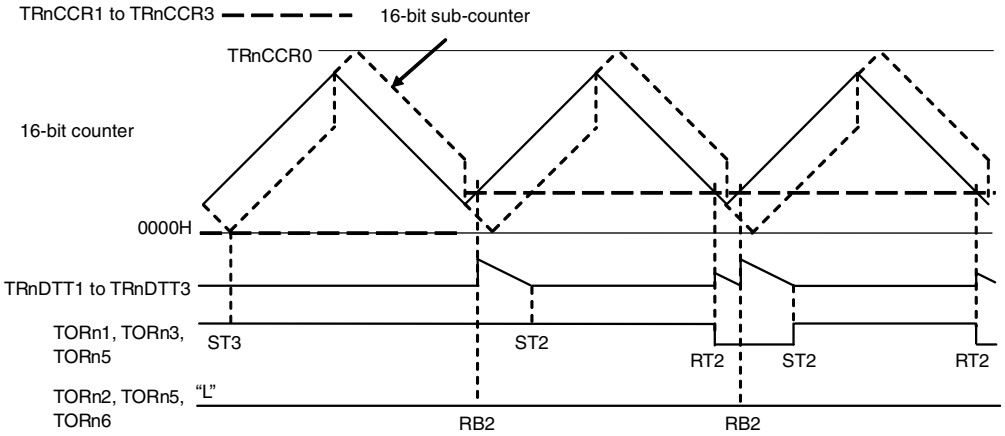


Figure 10-64: Timer Output Change after Compare Register Updating Timings (2/3)

(c) TRnCCR1 to TRnCCR3 = 0000H →TRnDTC0 < TRnCCR1 to TRnCCR3 < TRnDTC0 × 2



When the values of TRnCCR1 to TRnCCR3 are changed from "0000H ≤TRnCCR1 to TRnCCR3 < TRnDTC0" to "TRnDTC0 < TRnCCR1 to TRnCCR3 < TRnDTC0 × 2", the positive phase will be 100% output for one cycle, as shown in Figure 10-64c.
To prevent this phenomenon, change "0000H ≤ TRnCCR1 to TRnCCR3 < TRnDTC0" to "TRnDTC0 < TRnCCR1 to TRnCCR3 < TRnDTC × 2" through TRnDTC0, or directly change "0000H ≤TRnCCR1 to TRnCCR3 < TRnDTC0" to "TRnDTC0 × 2 ≤TRnCCR1 to TRnCCR3".

(d) TRnCCR1 to TRnCCR3 = 0000H →TRnDTC0 × 2 < TRnCCR1 to TRnCCR3 < TRnCCR0 − TRnDTC1 −TRnDTC0

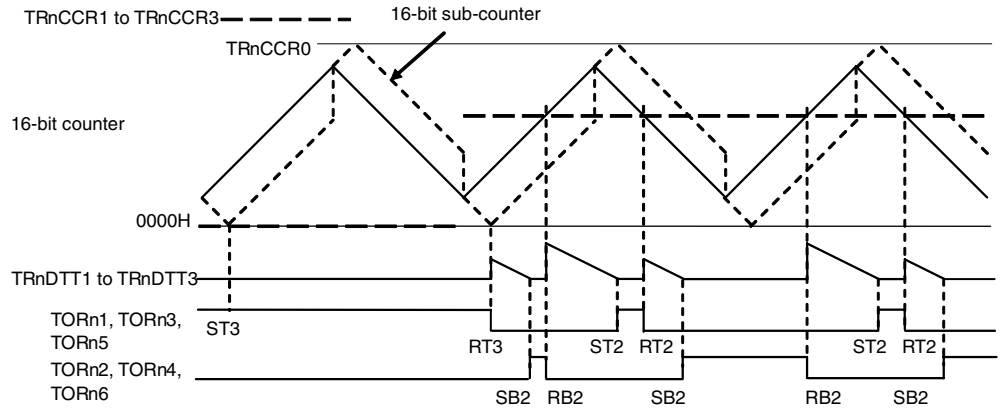
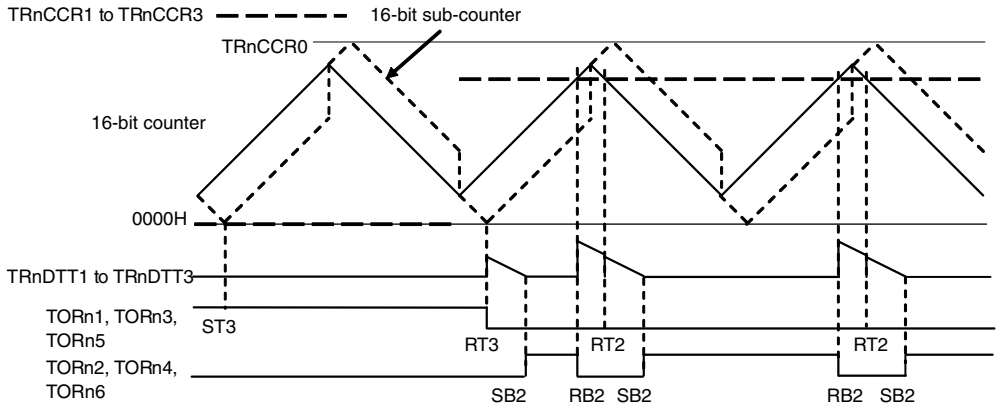
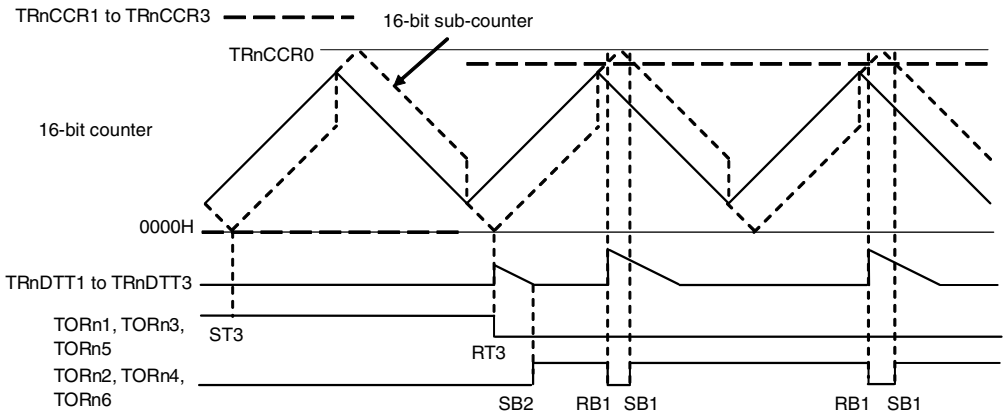


Figure 10-64: Timer Output Change after Compare Register Updating Timings (3/3)

(e) TRnCCR1 to TRnCCR3 = 0000H →TRnCCR0 –TRnDTC1 –TRnDTC0 < TRnCCR1 to TRnCCR3 < TRnCCR0 –TRnDTC1



(f) TRnCCR1 to TRnCCR3 = 0000H →TRnCCR0 –TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0



(g) TRnCCR1 to TRnCCR3 = 0000H →TRnCCR0 –TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0

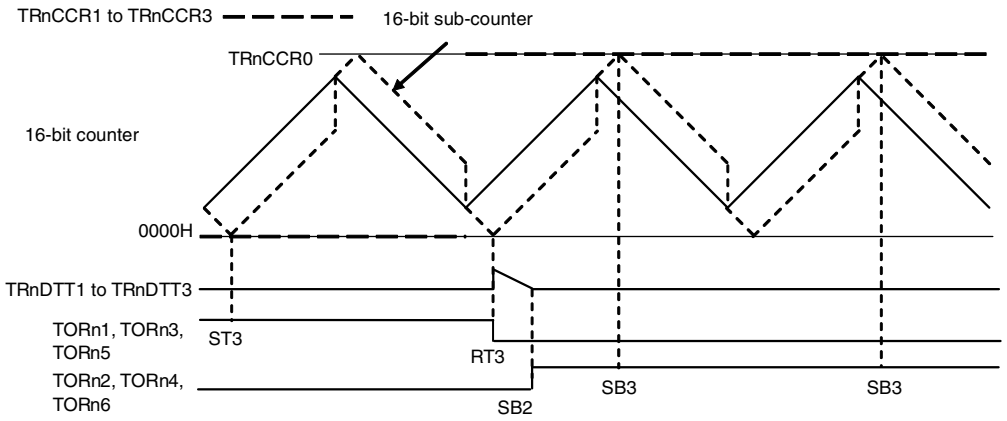
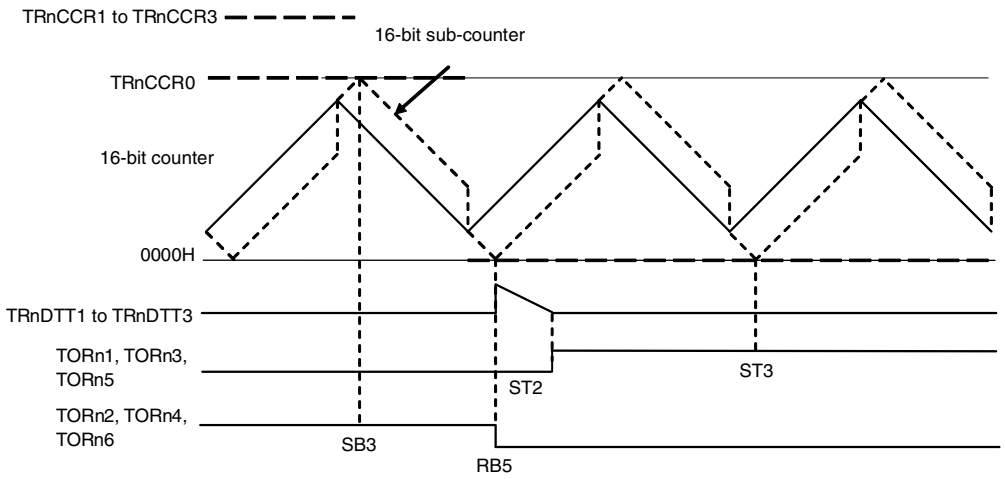


Table 10-4: Compare Register Value After Trough Reload

Compare Register Value Immediately Before Trough Reload	Compare Register Value After Trough Reload	Figure No.
TRnCCR0	TRnCCR1 to TRnCCR3 = 0000H	Figure 10-65a
	0000H < TRnCCR1 to TRnCCR3 < TRnDTC0	Figure 10-65b
	TRnCCR1 to TRnCCR3 = TRnDTC0, TRnDTC0 + 1	Figure 10-65c
	TRnDTC0 + 1 < TRnCCR1 to TRnCCR3 < TRnDTC0 + TRnDTC1	Figure 10-65d
	TRnDTC0 + TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0 –TRnDTC0 –TRnDTC1	Figure 10-65e
	TRnCCR0 –TRnDTC0 –TRnDTC1 ≤TRnCCR1 to TRnCCR3 < TRnCCR0 –TRnDTC1	Figure 10-65f
	TRnCCR0 –TRnDTC1 ≤TRnCCR1 to TRnCCR3 < TRnCCR0	Figure 10-65g

Figure 10-65: Compare Register Value After Trough Reload Timing (1/3)

(a) TRnCCR1 to TRnCCR3 = TRnCCR0 →TRnCCR1 to TRnCCR3 = 0000H



(b) TRnCCR1 to TRnCCR3 = TRnCCR0 →0000H < TRnCCR1 to TRnCCR3 < TRnDTC0

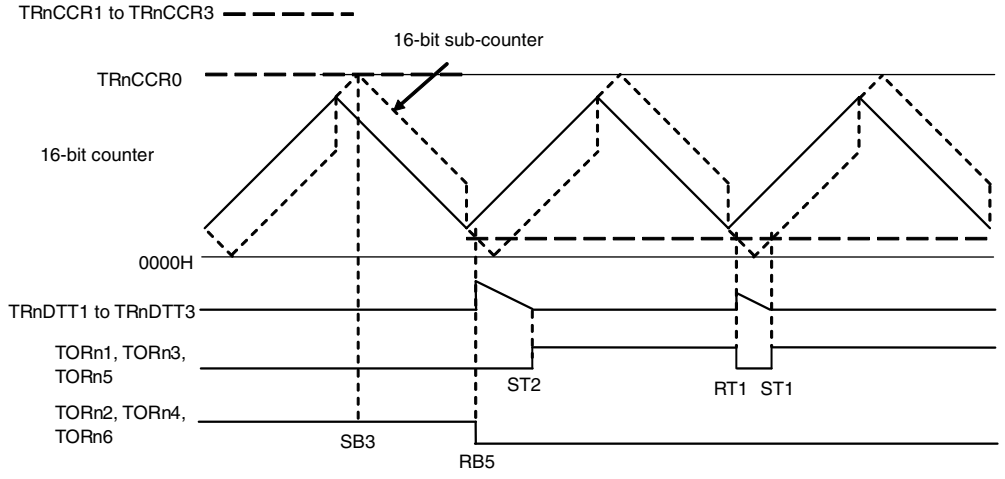
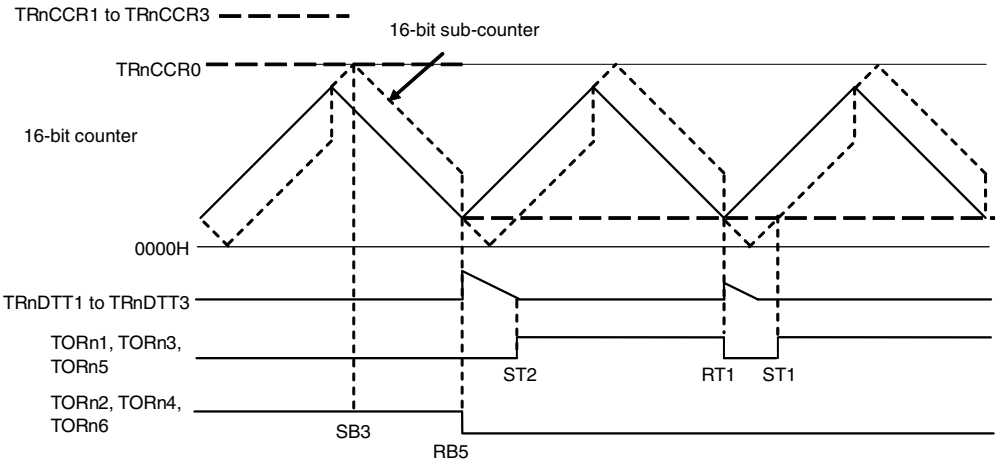
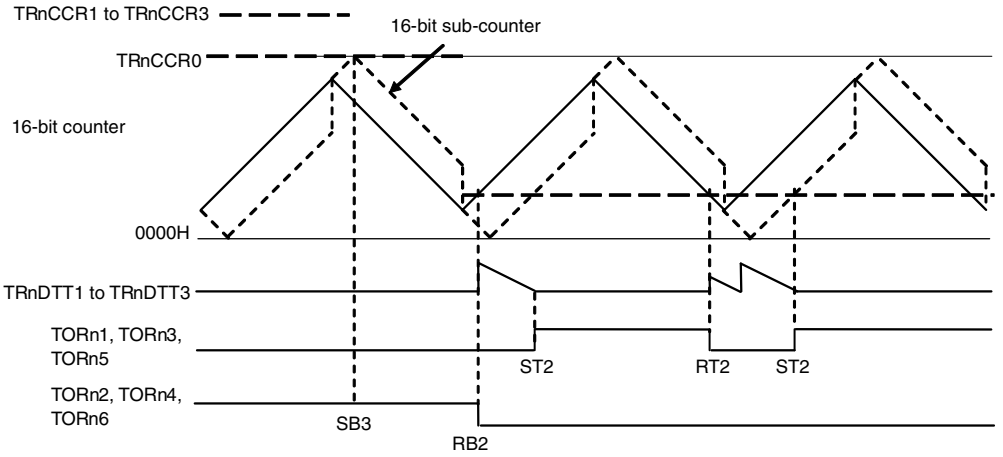


Figure 10-65: Compare Register Value After Trough Reload Timing (2/3)

(c) TRnCCR1 to TRnCCR3 = TRnCCR0 →TRnCCR1 to TRnCCR3 = TRnDTC0, TRnDTC0 + 1



(d) TRnCCR1 to TRnCCR3 = TRnCCR0 →TRnDTC0 + 1 < TRnCCR1 to TRnCCR3 ≤TRnDTC0 + TRnDTC1



(e) TRnCCR1 to TRnCCR3 = TRnCCR0 →TRnDTC0 + TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0 –TRnDTC1 –TRnDTC0

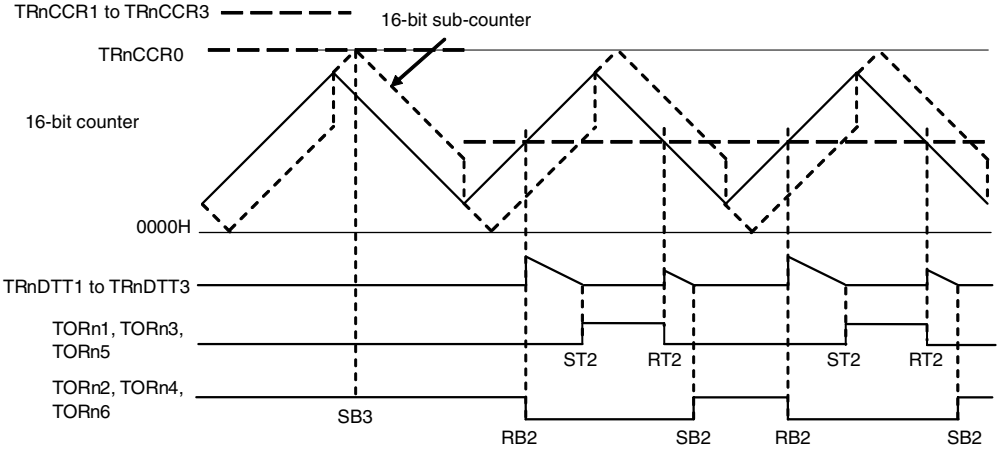
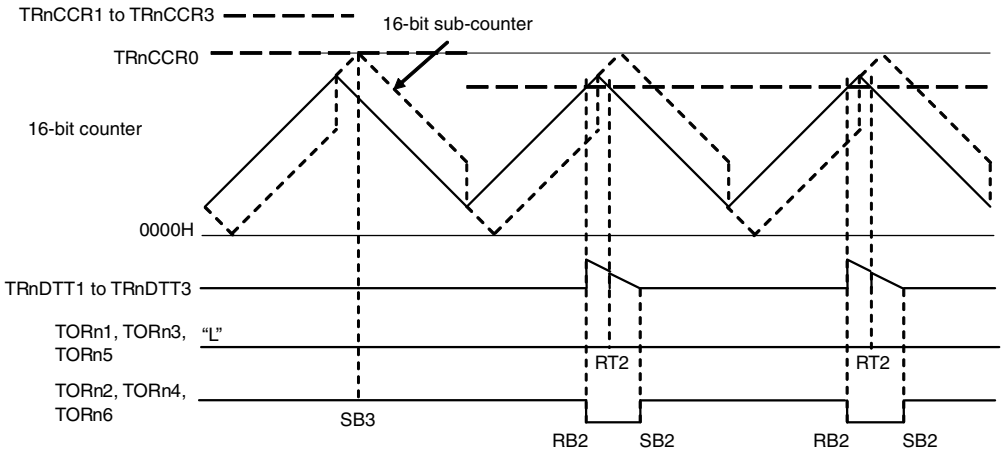


Figure 10-65: Compare Register Value After Trough Reload Timing (3/3)

(f) TRnCCR1 to TRnCCR3 = TRnCCR0 →TRnCCR0 –TRnDTC1 –TRnDTC0 < TRnCCR1 to TRnCCR3 < TRnCCR0 –TRnDTC1



(g) TRnCCR1 to TRnCCR3 = TRnCCR0 →TRnDTC0 –TRnDTC1 –TRnDTC0 < TRnCCR1 to TRnCCR3 < TRnCCR0

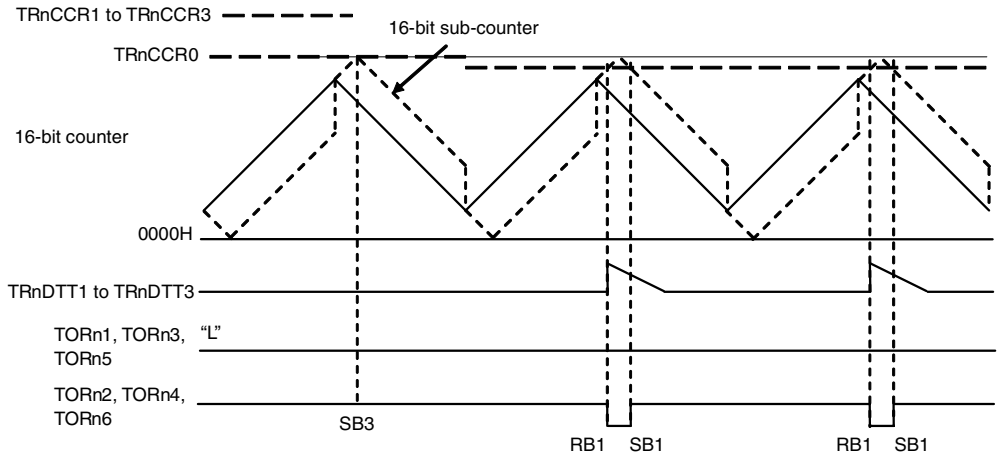
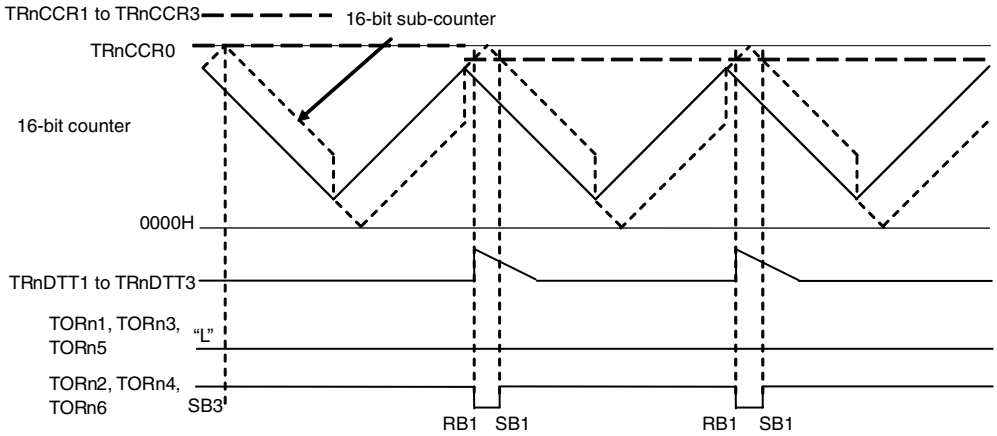


Table 10-5: Compare Register Value After Trough Reload (TRnDTC1 < TRnDTC0)

Compare Register Value Immediately Before Peak Reload	Compare Register Value After Trough Reload (TRnDTC1 < TRnDTC0)	Figure No.
TRnCCR0	TRnCCR0 –TRnDTC1 ≤TRnCCR1 to TRnCCR3 < TRnCCR0	Figure 10-66a
	TRnCCR1 to TRnCCR3 < TRnCCR0 –TRnDTC1	Figure 10-66b
	TRnCCR0 –TRnDTC1 × 2 ≤TRnCCR1 to TRnCCR3 < TRnCCR0 –TRnDTC1	Figure 10-66c
	TRnDTC0 + TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0 –TRnDTC1 × 2	Figure 10-66d
	TRnDTC0 + 1 < TRnCCR1 to TRnCCR3 < TRnDTC0 + TRnDTC1	Figure 10-66e
	0000H < TRnCCR1 to TRnCCR3 ≤TRnDTC0 + TRnDTC1	Figure 10-66f
	TRnCCR1 to TRnCCR3 = 0000H	Figure 10-66g

Figure 10-66: Compare Register Value After Trough Reload (TRnDTC1 < TRnDTC0) (1/3)

(a) TRnCCR1 to TRnCCR3 = TRnCCR0 →TRnCCR0 −TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0



(b) TRnCCR1 to TRnCCR3 = TRnCCR0 →TRnCCR1 to TRnCCR3 = TRnDTC0 −TRnDTC1

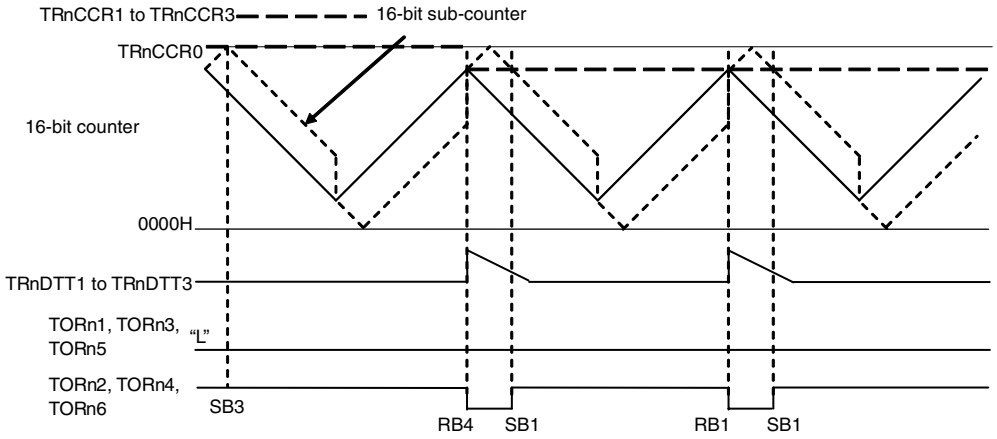
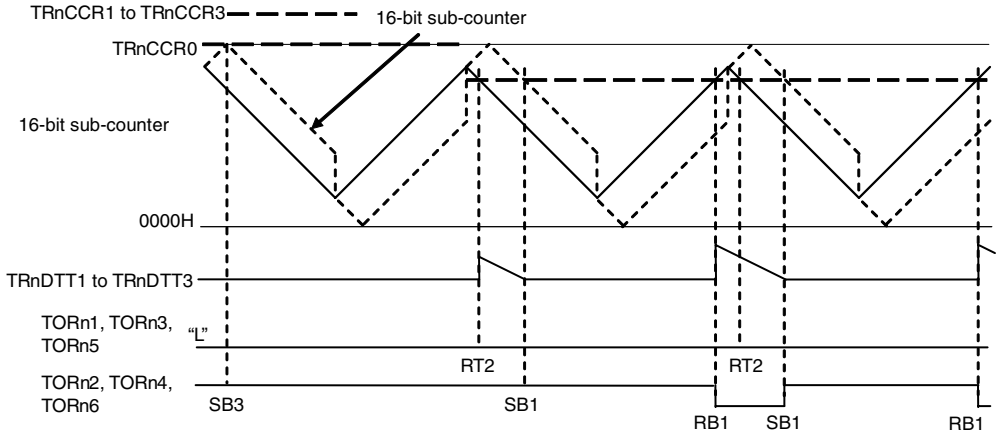


Figure 10-66: Compare Register Value After Trough Reload (TRnDTC1 < TRnDTC0) (2/3)

(c) TRnCCR1 to TRnCCR3 = TRnCCR0 →TRnCCR0 −TRnDTC1 x 2 < TRnCCR1 to TRnCCR3 < TRnCCR0 −TRnDTC1

When the values of TRnCCR1 to TRnCCR3 are changed from "TRnCCR0 − TRnDTC1 < TRnCCR1 to TRnCCR3 ≤TRnCCR0" to "TRnCCR0 −TRnDTC1 × 2 < TRnCCR1 to TRnCCR3 < TRnCCR0 −TRnDTC1", the negative phase will be 100% output for one cycle, as shown in figure below.

To prevent this phenomenon, change "TRnCCR0 − TRnDTC1 < TRnCCR1 to TRnCCR3 ≤ TRnCCR0" to "TRnDTC0 < TRnCCR1 to TRnCCR3 < TRnDT1 × 2" through "TRnCCR0 − TRnDTC1", or directly change "TRnCCR0 −TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0" to "TRnCCR1 to TRnCCR3 ≤TRnCCR0 −TRnDTC1 × 2".



(d) TRnCCR1 to TRnCCR3 = TRnCCR0 →TRnDTC0 + TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0 −TRnDTC1 x 2

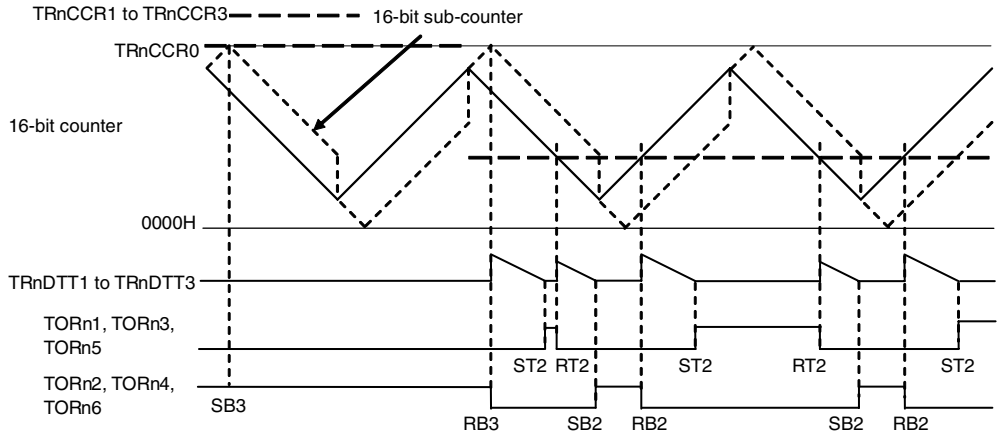
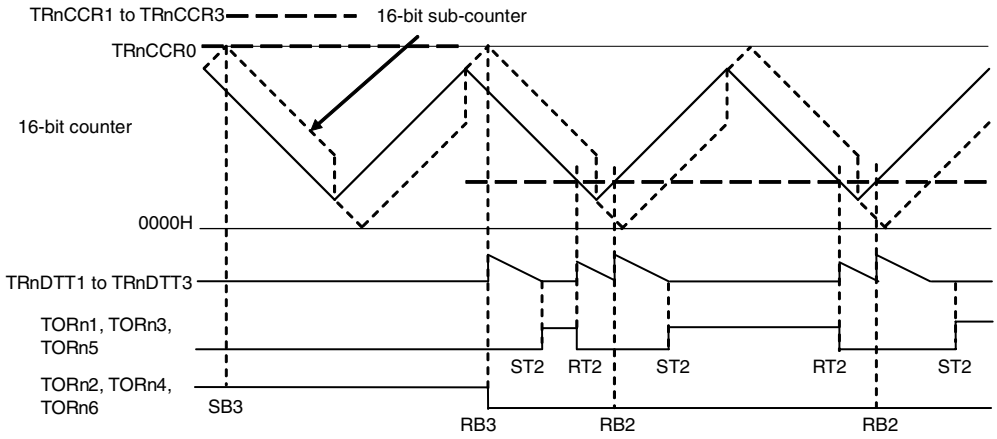
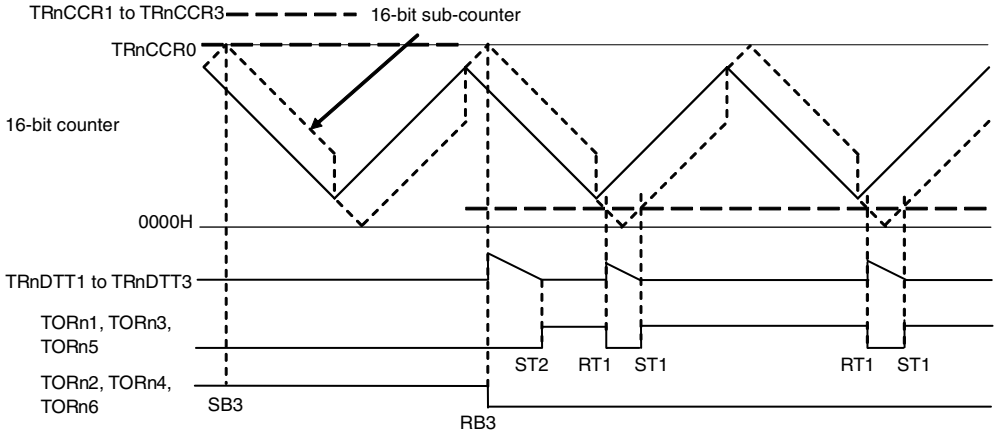


Figure 10-66: Compare Register Value After Trough Reload (TRnDTC1 < TRnDTC0) (3/3)

(e) TRnCCR1 to TRnCCR3 = TRnCCR0 →TRnDTC0 + 1 < TRnCCR1 to TRnCCR3 ≤TRnDTC0 + TRnDTC1



(f) TRnCCR1 to TRnCCR3 = TRnCCR0 →0000H < TRnCCR1 to TRnCCR3 ≤TRnDTC0 + 1



(g) TRnCCR1 to TRnCCR3 = TRnCCR0 →TRnCCR1 to TRnCCR3 = 0000H

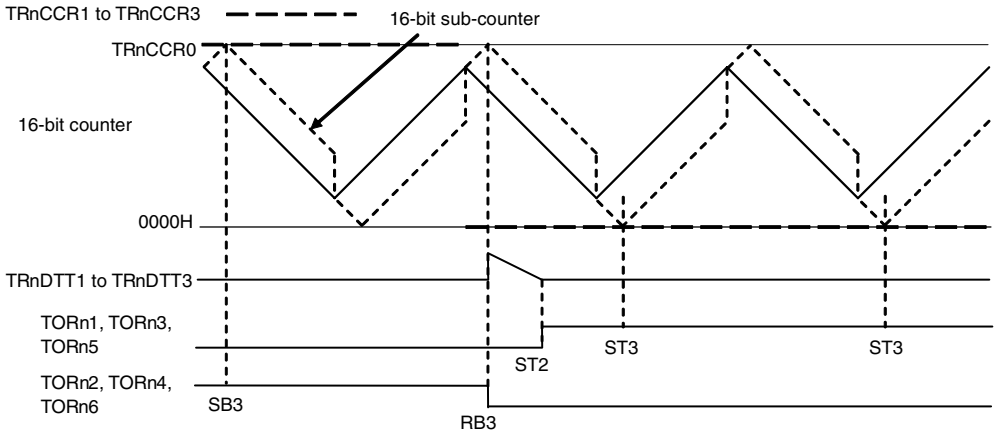
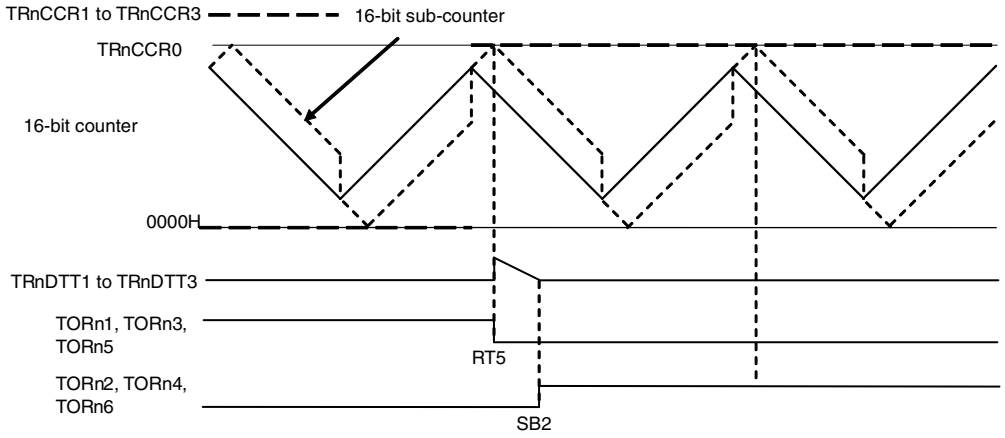


Table 10-6: Compare Register Value After Trough Reload

Compare Register Value Immediately Before Peak Reload	Compare Register Value After Trough Reload	Figure No.
0000H	TRnCCR1 to TRnCCR3 = TRnCCR0	Figure 10-67a
	TRnCCR0 –TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0	Figure 10-67b
	TRnCCR1 to TRnCCR3 = TRnCCR0 –TRnDTC1	Figure 10-67c
	TRnCCR0 –TRnDTC0 –TRnDTC1 ≤TRnCCR1 to TRnCCR3 < TRnCCR0 –TRnDTC1	Figure 10-67d
	TRnDTC0 + TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0 –TRnDTC0 –TRnDTC1	Figure 10-67e
	TRnDTC0 + 1 < TRnCCR1 to TRnCCR3 ≤TRnDTC0 + TRnDTC1	Figure 10-67f
	0000H < TRnCCR1 to TRnCCR3 ≤TRnDTC0 + 1	Figure 10-67g

Figure 10-67: Compare Register Value After Trough Reload (1/3)

(a) TRnCCR1 to TRnCCR3 = 0000H →TRnCCR1 to TRnCCR3 = TRnCCR0



(b) TRTRnCCR1 to TRnCCR3 = 0000H →TRnCCR0 –TRnDTC1 < TRTRnCCR1 to TRnCCR3 < TRnCCR0

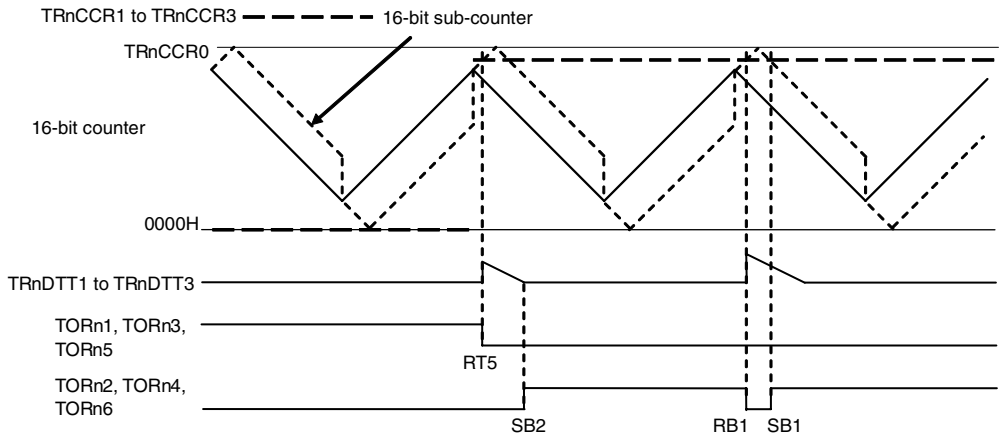
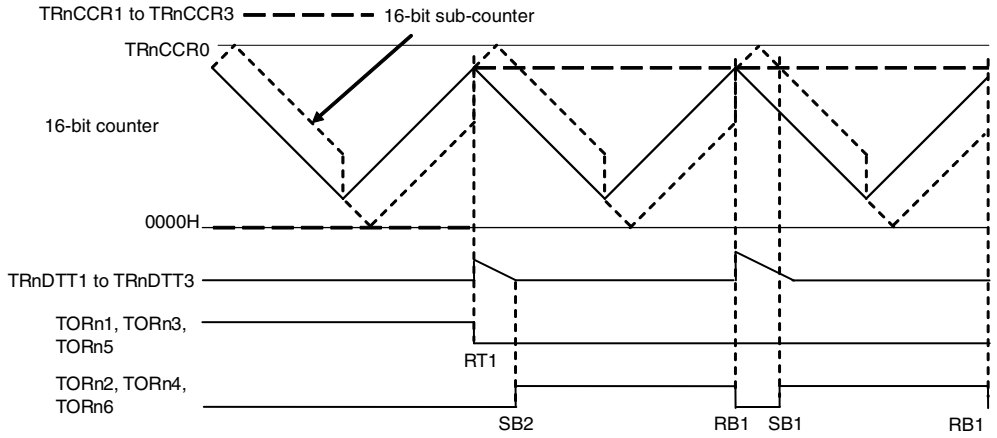
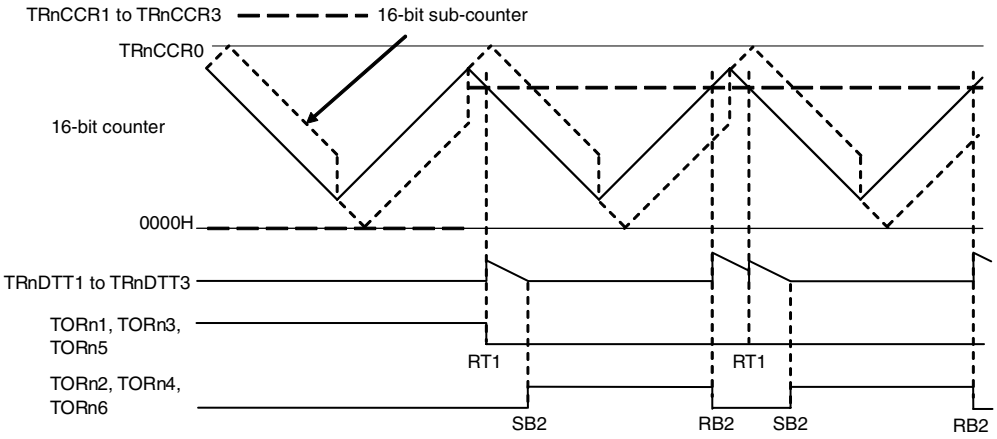


Figure 10-67: Compare Register Value After Trough Reload (2/3)

(c) TRnCCR1 to TRnCCR3 = 0000H →TRnCCR1 to TRnCCR3 = TRnCCR0 −TRnDTC1



(d) TRnCCR1 to TRnCCR3 = 0000H →TRnCCR0 −TRnDTC0 −TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0 −TRnDTC1



(e) TRnCCR1 to TRnCCR3 = 0000H →TRnDTC0 + TRnDTC1 < TRnCCR1 to TRnCCR3 ≤ TRnCCR0 −TRnDTC0 −TRnDTC1

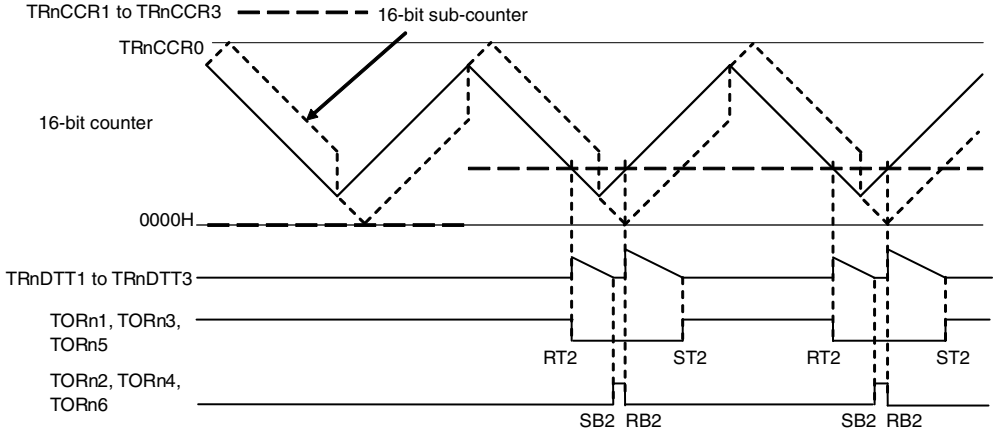
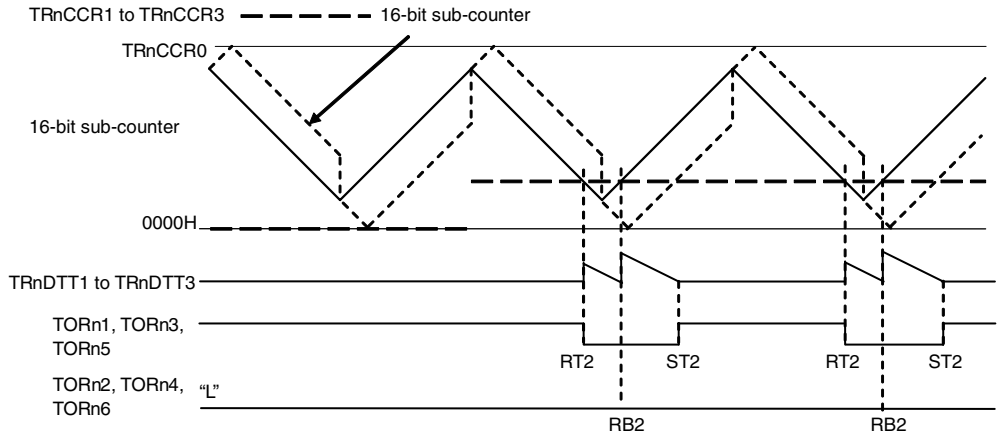
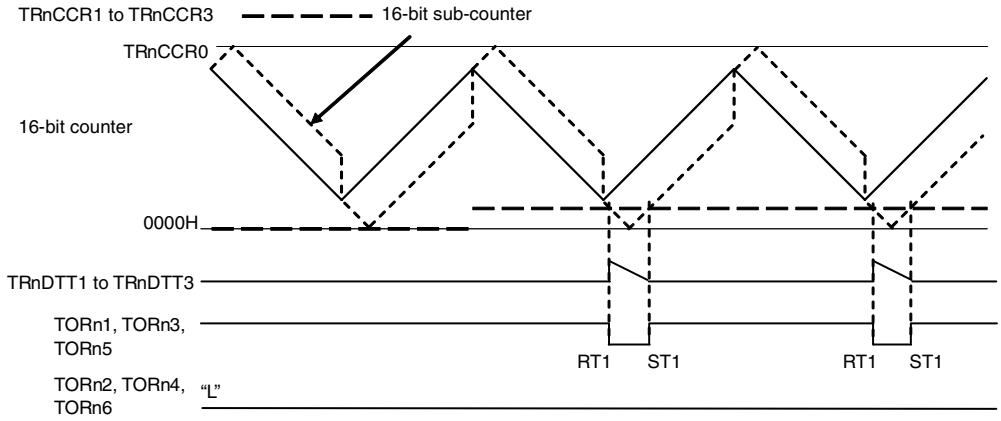


Figure 10-67: Compare Register Value After Trough Reload (3/3)

(f) TRnCCR1 to TRnCCR3 = 0000H →TRnDTC0 + 1 < TRnCCR1 to TRnCCR3 < TRnDTC0 + TRnDTC1



(g) TRnCCR1 to TRnCCR3 = 0000H →0000H < TRnCCR1 to TRnCCR3 ≤TRnDTC0 + 1



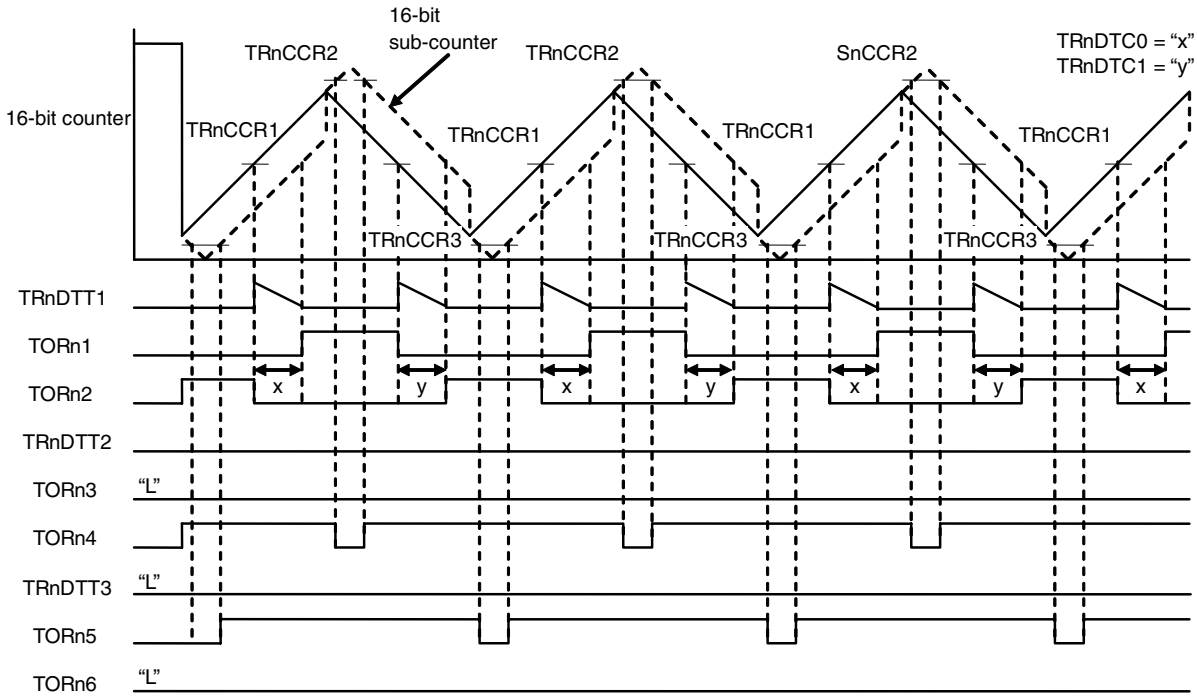
(9) Dead time control in high-accuracy T-PWM mode

In the high-accuracy T-PWM mode, the TRnCCR1 to TRnCCR3 registers are used for duty setting and the TRnCCR0 register is used for cycle setting. By using these four registers, duty variable type 6-phase PWM waveform can be output. To implement dead time control, there are three 10-bit down-counters that synchronously operate with the count clock of the 16-bit counter, and two dead time setting registers (TRnDTC0, TRnDTC1).

The TRnDTC0 register is used to set the dead time from when a negative phase changes to inactive until a positive phase changes to active. The TRnDTC1 register is used to set the dead time from when a positive phase changes to inactive until a negative phase changes to active.

The output waveform in case of TRnDTC0 = x, TRnDTC1 = y is shown below.

Figure 10-68: Output Waveform Example When Dead Time Is Set



(10) Cautions on dead time control in high-accuracy T-PWM mode

(a) Rewriting of TRnDTC0 and TRnDTC1 registers

The setting of the dead time in the TRnDTC0, TRnDTC1 registers can be rewritten during operation. Note the following cautions when rewriting the dead time setting during operation.

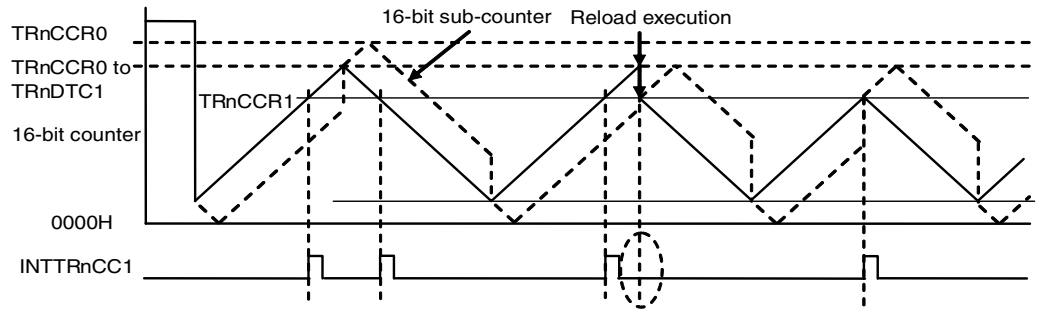
Cautions: 1. Rewrite the TRnDTC0 and TRnDTC1 registers when using the reload function (TRnCMS = 0).

2. When the TRnDTC0 and TRnDTC1 registers are rewritten, carrier-wave cycles will be changed. In cases where carrier-wave cycles should not be changed, rewrite the TRnCCR0 register value at the same time as changing the TRnDTC0 and TRnDTC1 registers.

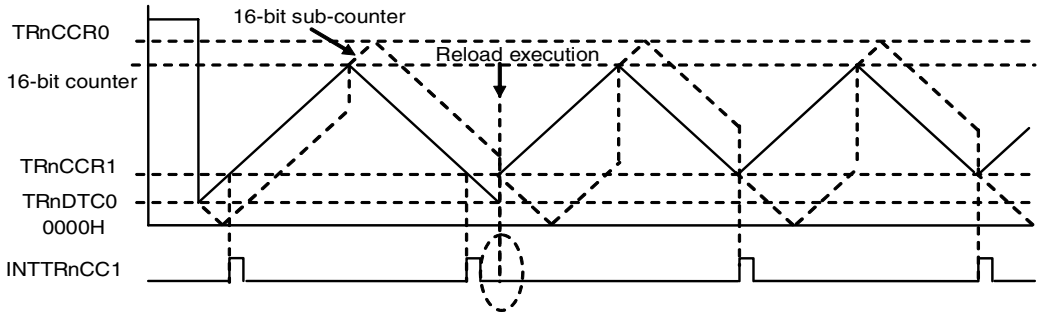
3. Rewriting is prohibited when TRnCMS = 1.

4. In case of changing TRnCCR0 and TRnCCR1 at a 16-bit counter peak:
Match interrupts (INTTRnCC1 to INTTRnCC5) will not occur immediately after reload execution if the values set in the TRnCCR1 to TRnCCR5 register matches with and TRnCCR0 −TRnDTC1 (the new maximum value of main counter) after updating.

Figure 10-69: Dead Time Control in High-Accuracy T-PWM Mode



5. In case of changing TR0DTC0 at a 16-bit counter trough:
Match interrupts (INTTRnCC1 to INTTRnCC5) will not occur immediately after reload execution if the values set in the TRnCCR1 to TRnCCR5 register match with TR0DTC0 (the new minimum value of main counter) after updating.



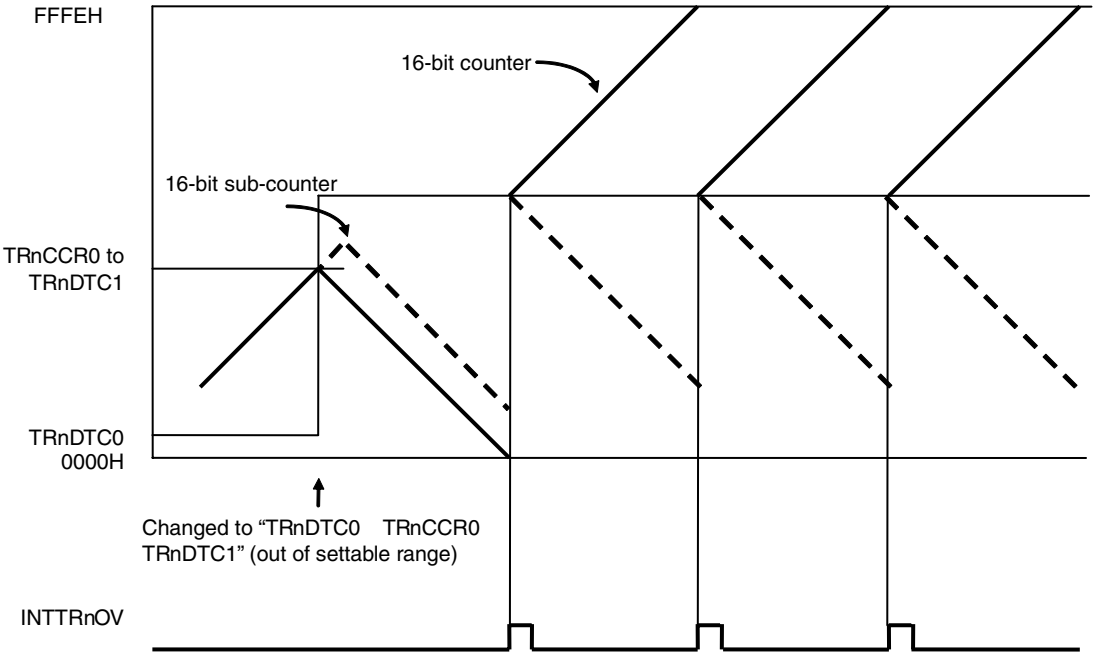
(11) Caution on rewriting cycles in high-accuracy T-PWM mode

In high-accuracy T-PWM mode, setting conditions for the TRnCCR0, TRnDTC0, and TRnDTC1 registers are as follows.

- 3 × MAX (TRnDTC0, TRnDTC1) + MIN (TRnDTC0, TRnDTC1) < TRnCCR0
  0002H < TRnCCR0 ≤FFFEH
- MAX (A, B) indicates the greater value of A and B, and MIN (A, B) indicates the smaller value of A and B.

Figure 10-70 shows an operation example when the setting range is exceeded.
This example shows the case where the TRnDTC0 register is set out of the range "TRnDTC0 ≥ TRnCCR0 −TRnDTC1". Though the 16-bit counter executes count-down operation, the count-down operation is executed from 0000H because no match occurs. In this case, the count operation continues by loading the TRnDTC0 register setting value. However, no match with TRnCCR0 −TRnDTC1 occurs in the count-up operation, thus the 16-bit counter overflows. In this case, the count operation continues by loading the TRnDTC0 register setting value again.
An overflow interrupt (INTTRnOV) occurs when the 16-bit counter loads the TRnDTC0 register setting value from 0000H or when an overflow occurs at FFFEH, and then the TRnOVF flag is set. An overflow interrupt (INTTRnOV) does not occur if the TRnCCR0, TRnDTC0, and TRnDTC1 registers are set correctly, so this can be used for detecting incorrect settings.

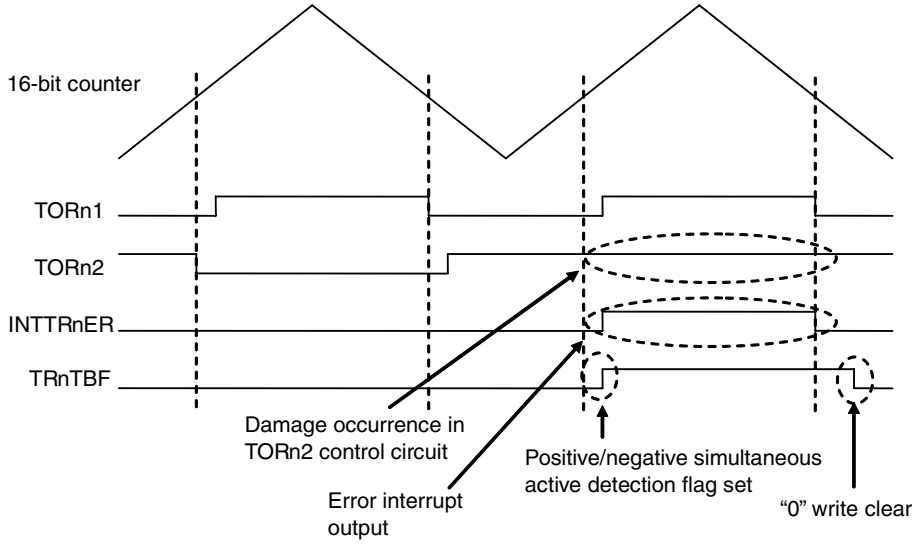
Figure 10-70: Operation Example Setting Is Out of Range



(12) Error interrupt (INTTRnER) in high-accuracy T-PWM mode

The positive/negative simultaneous active detection function can be used in the high-accuracy T-PWM mode. Error interrupts (INTTRnER) do not occur in the high-accuracy T-PWM mode. In case of occurrence, the internal circuits may be damaged.

Figure 10-71: Error Interrupt Operation Example

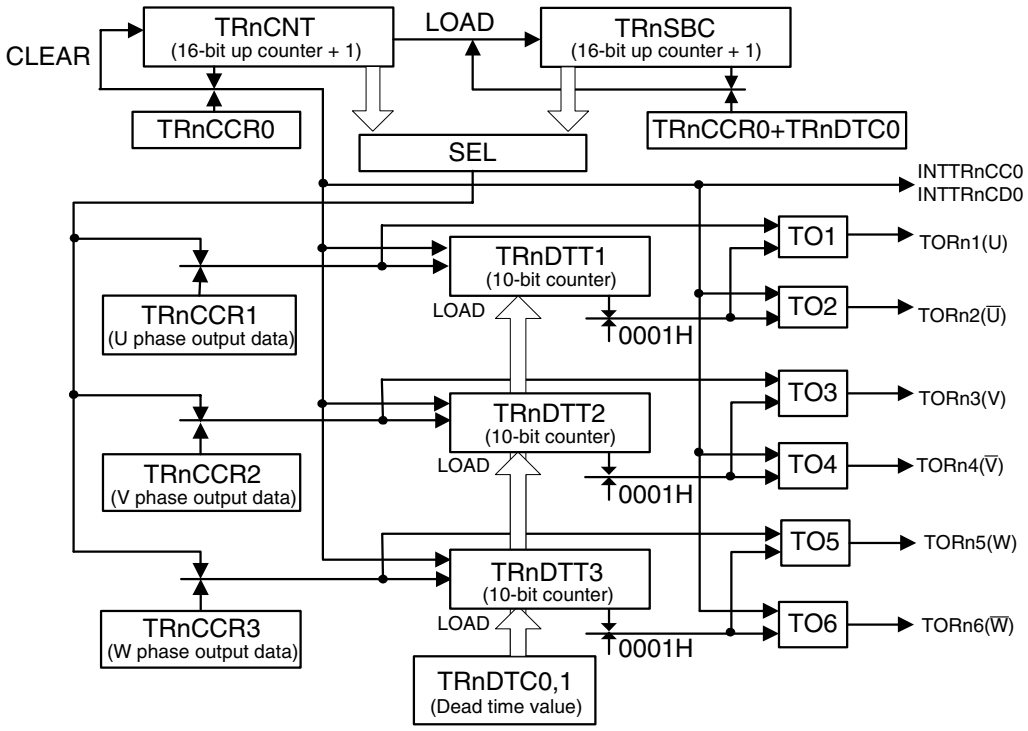


10.10.10 PWM mode with dead time

(1) Outline of PWM mode with dead time

In the PWM mode with dead time, 6-phase PWM is generated using the 16-bit counter's saw tooth wave operation and four 16-bit counters. The counter's maximum value is set with the TRnCCR0 register. The duties of the U phase, V phase, and W phase voltage data signals are set with the TRnCCR1 to TRnCCR3 registers. The dead time is set with the TRnDTC0 and TRnDTC1 registers, and the dead time for inverted phase →normal phase and the dead time for normal phase →inverted phase can be independently set with the TRnDTC0 register and TRnDTC1 register, respectively. The counter's operation consists in performing up count with 0000H as the minimum value, and when the maximum value (cycle) indicated by the TRnCCR0 register is matched, the counter is cleared (0000H), and the counter continues up-count operation.
The 10-bit dead time counters (TRnDTT1 to TRnDTT3) reload the setting value of the TRnDTC0 and TRnDTC1 registers upon a match between the counter and the TRnCCR1 to TRnCCR3 registers, and perform down count.
Upon a match between the 16-bit counter and the TRnCCR0 to TRnCCR3 registers, the corresponding compare match interrupts (INTTRnCC1 to INTTRnCC3) are output.

Figure 10-72: Block Diagram in PWM Mode With Dead Time



(2) PWM mode with dead time operation list

(a) Compare registers

Register	Rewrite Method	Rewrite during Operation	Function
TRnCCR0	Reload	Possible	Cycle
TRnCCR1 to TRnCCR3	Reload	Possible	PWM duty
TRnCCR4, TRnCCR5	Reload	Possible	PWM duty

(b) Input pins

Pin	Function
TIR1m	- (m = 0 to 3)
TTRGR1	-
TEVTR1	-

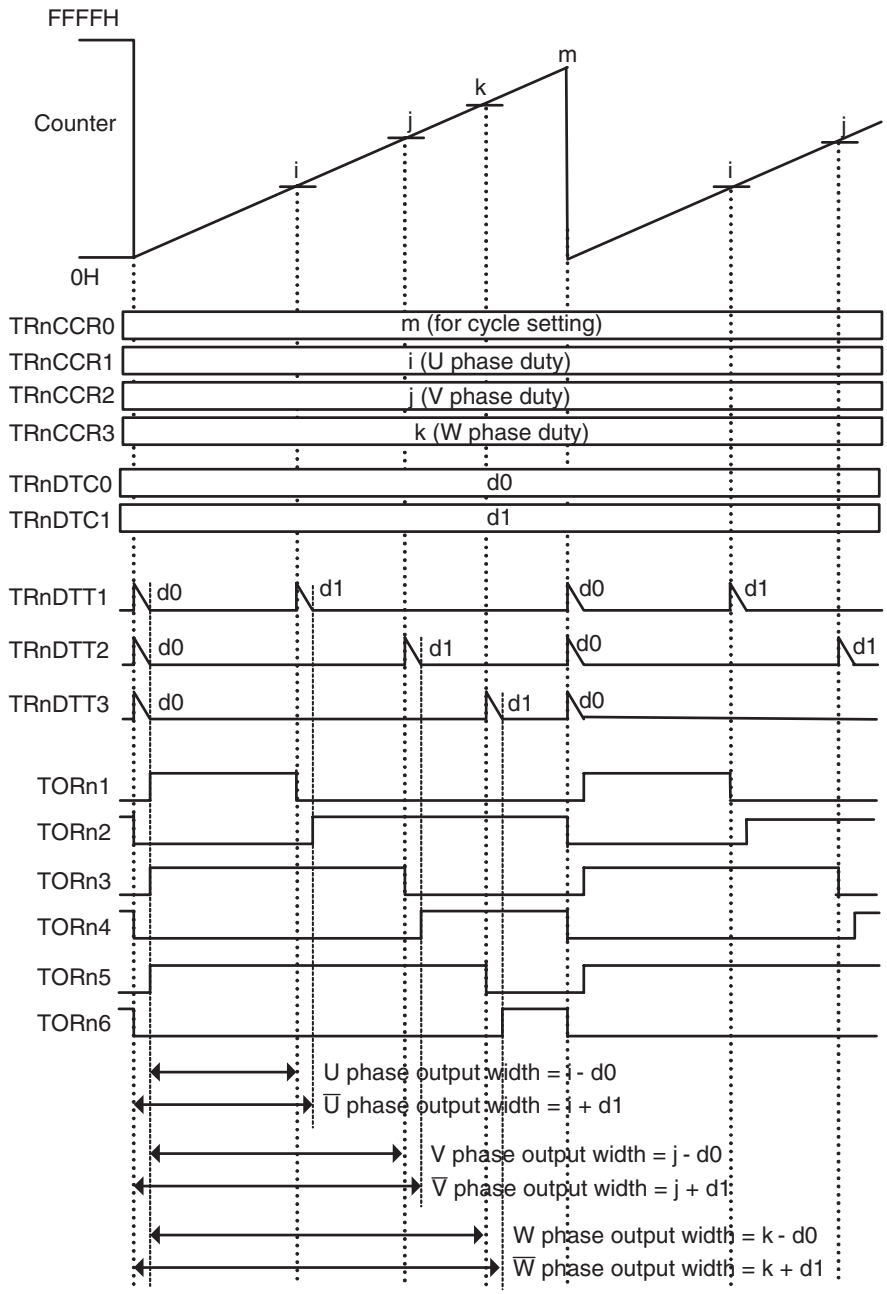
(c) Output pins

Pin	Function
TORn0	Toggle output upon TRnCCR0 register compare match
TORn1	PWM output (with dead time) upon TRnCCR1 register compare match
TORn2	Inverted phase output to TORn1
TORn3	PWM output (with dead time) upon TRnCCR2 register compare match
TORn4	Inverted phase output to TORn3
TORn5	PWM output (with dead time) upon TRnCCR3 register compare match
TORn6	Inverted phase output to TORn5
TORn7	Pulse output through A/D conversion trigger

(d) Interrupts

Interrupt	Function
INTTRnCCm	TRnCCRm register compare match (m = 0 to 5)
INTTRnOV	-
INTTRnER	Error

Figure 10-73: Output Waveform Example in PWM Mode with Dead Time



Remarks: 1. The maximum value that can be set to the TRnCCR1 to TRnCCR3 registers is TRnCCR0 + TRnDTC0.

2. If “0000H” is set to the TRnCCR1 to TRnCCR3 registers, PWM is output with 0% duty.

3. If TRnCCR0 + TRnDTC0 is set to the TRnCCR1 to TRnCCR3 registers, PWM is output with 100% duty.

4. The maximum value of the TRnCCR0 register is FFFFH - TRnDTC0.

5. Perform setting so as to satisfy condition FFFFH > TRnCCR0 + TRnDTC0.

(3) PWM mode with dead time settings

(a) Mode setting

The PWM mode with dead time is set by setting TRnCTL1 register bits TRnMD4 to TRnMD0 = 1001.

(b) Output level/output enable settings

Output level/output enable is set by setting the TRnOL0 to TRnOL7 and TRnOE0 to TRnOE7 bits of the TRnIOC0 and TRnIOC3 registers.
Pin TORn0 performs toggle output upon cycle match (match between the counter and the TRnCCR0 register).
Pin TORn7 is the output for A/D conversion. Set this pin as required.

(c) Error output enable

Set error output enable when normal phase/inverted phase simultaneous active is detected. Error output is enabled by setting TRnIOC4 register bit TRnEOC to 1. Moreover, the pin for detecting simultaneous active can also be set, by setting TRnIOC4 register bits TRnTBA2 to TRnTBA0. In the PWM mode with dead time, INTTRnER does not become active, regardless of which value the user sets to the TRnCCR0 to TRnCCR3 registers, except when the dead time setting is 0. When an error occurs, this indicates an internal circuit fault.

(d) Interrupt and thinning out function settings

A peak interrupt (INTTRnCD) occurs upon a match between the TRnCCR0 register and the counter (bit TRnIOE control is invalid). To output a peak interrupt, set TRnICE = 1. Use of the thinning out function for peak interrupts is done with the TRnID4 to TRnID0 registers.

(e) Reload thinning out function setting

To set the reload timing to the same timing as the interrupt timing, set TRnOPT1 register bit TRnRDE to 1. The reload timing occurs when TRnICE = 1.

(f) A/D conversion trigger output setting

A/D conversion trigger 0 (TRnADTRG0 signal) is set with TRnOPT2 register bits TRnAT04, TRnAT02, and TRnAT01. The TRnCCR5 register match timing, TRnCCR4 register match timing, and peak interrupt (INTTRnCD) enable/disable settings are performed with bits TRnAT04, TRnAT02, and TRnAT01.
Do not set TRnAT05, TRnAT03, and TRnAT00 to “1”.
A/D conversion trigger 1 (TRnADTRG1 signal) is set with TRnOPT3 register bits TRnAT14, TRnAT12, and TRnAT11. The TRnCCR5 register match timing, TRnCCR4 register match timing, and peak interrupt (INTTRnCD) enable/disable settings are performed with bits TRnAT14, TRnAT12, and TRnAT11.
Do not set bits TRnAT15, TRnAT13, and TRnAT10 to “1”.
Set the compare values of the TRnCCR4 and TRnCCR5 registers.

(g) Dead time settings

The dead time settings are performed with the TRnDTC0 and TRnDTC1 registers. The dead time can be obtained with count clock cycle × TRnDTC0,TRnDTC1. The time until TORn2, TORn4, TORn6 pin inactive change →TORn1, TORn3, TORn5 pin active change can be set with the TRnDTC0 register. The time until TORn1,TORn3,TORn5 pin inactive change →TORn2, TORn4, TORn6 pin active change can be set with the TRnDTC1 register.

(h) PWM cycle, duty (PWM width) setting

The duty is set with the TRnCCR1 to TRnCCR3 registers. The setting range of the TRnCCR1 to TRnCCR3 registers is

0000H ≤TRnCCRm ≤(TRnCCR0 + TRnDTC0)

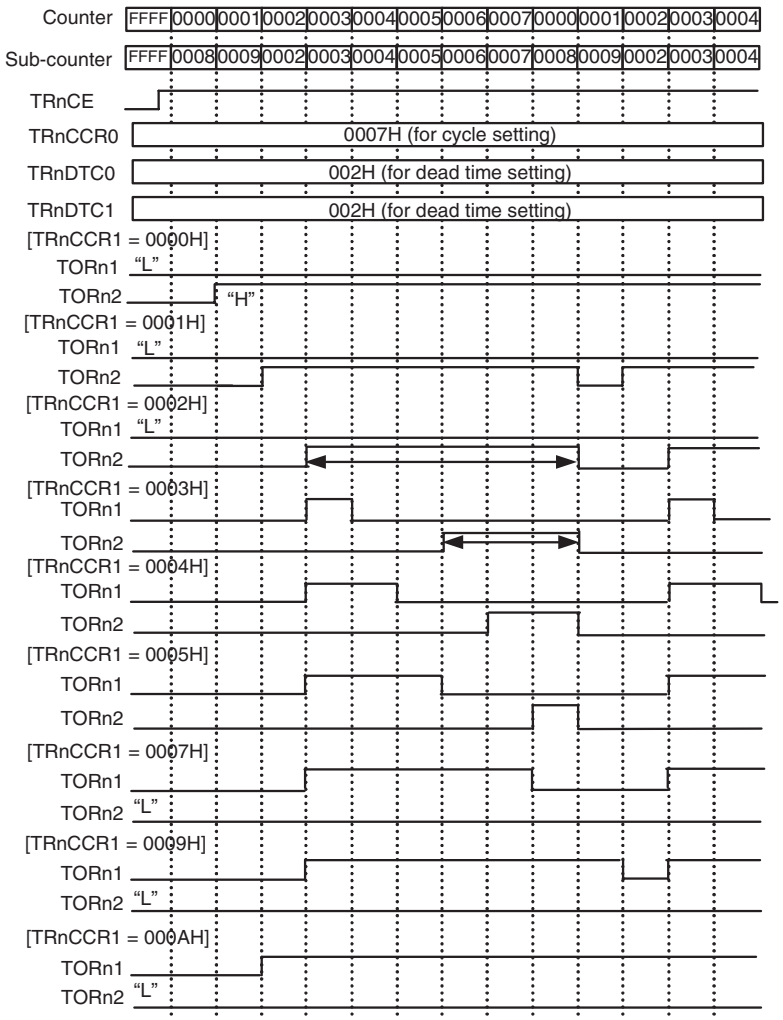
The TRnCCR0 and TRnDTC0 registers must be set so as to satisfy TRnCCR0 + TRnDTC0 < FFFFH.

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

(4) Operation in PWM mode with dead time

The figure shows the timing chart when TRnCCR0 = 0007H, TRnDTC0 = 0002H, TRnDTC1 = 0002H, and the TRnCCR0 register is set to 0000H to 0007H (one part).
When the compare value of the TRnCCR1 register is incremented/decremented by 1 at a time, the PWM width is incremented/decremented 1 count clock at a time, but at the points indicated by arrows in the figure, incrementing/decrementing is done by TRnDTC1+1 count clock. This occurs when the TRnCCR1 register is rewritten from the setting value of the TRnDTC0 register to TRnDTC0+0001H (because dead time control is required).

Figure 10-74: Timer Output Example When TRnCE = 1 Is Set (Initial) (PWM mode with Dead Time)

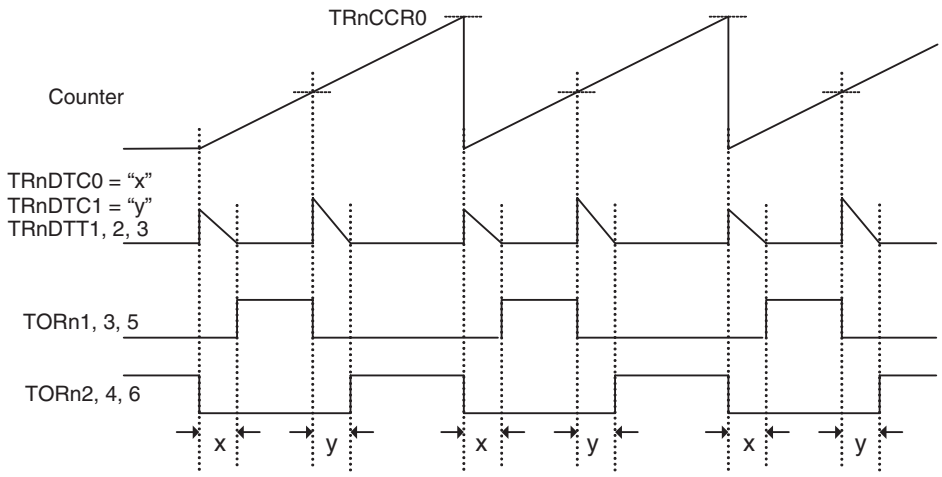


(5) Dead time control in PWM mode with dead time

In the PWM mode with dead time, compare registers (TRnCCR1 to TRnCCR3) are used as the duty setting registers, and another compare register (TRnCCR0) is used as the cycle setting register. Through the use of these four registers, a variable duty 6-phase PWM waveform is output. To realize dead time control, three 10-bit down counters that operate in synchronization with the counter's count clock, and dead time setting registers (TRnDTC0, TRnDTC1) are provided. The TRnDTC0 register is used to set the dead time from when the inverted phase becomes inactive to when the normal phase becomes active, and the TRnDTC1 register is used to set the dead time from when the normal phase becomes inactive to when the inverted phase becomes active.

The following figure shows an output example when TRnDTC0 = x, TRnDTC1 = y.

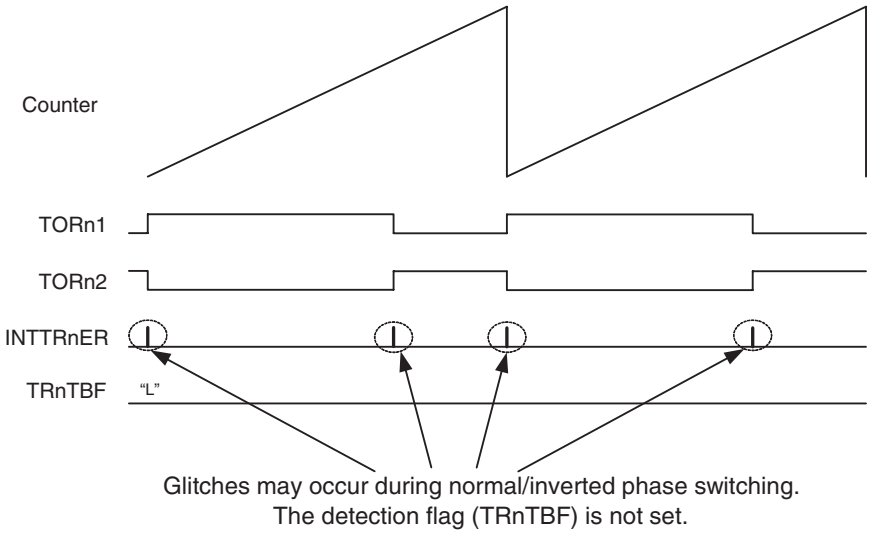
Figure 10-75: Output Waveform Example in PWM Mode with Dead Time



(6) Error interrupt (INTTRnER) in PWM mode with dead time

In the PWM mode with dead time, the normal phase/inverted phase simultaneous active detection function can be used. When using the PWM mode with dead time, no error interrupt (INTTRnER) is output as long as no hardware fault occurs (except when TRnDTC0, TRnDTC1 = 0000H is set). Also, when TRnDTC0, TRnDTC1 = 000H is set, glitches may occur upon error interrupt (INTTRnER) output. In this case, the occurrence of glitches during error interrupt (INTTRnER) output can be prevented by setting bit TRnEOC to 0.

Figure 10-76: Error Interrupt (INTTRnER) in PWM Mode with Dead Time



[MEMO]

Chapter 11 16-bit Timer/Event Counter T

11.1 Features

Timer T (TMT) is a 16-bit timer/event counter that provides general-purpose functions.
Timer T can perform the following operations.

- Interval timer function
- External event count function
- One-shot pulse output function
- External trigger pulse function
- 16-bit accuracy PWM output function
- Free-running function
- Pulse width measurement function
- 2-phase encoder function
- Triangular wave PWM output function
- Offset trigger generation function

11.2 Function Outline

- Capture trigger input signal × 2
- Encoder input signal × 2
- Encoder clear signal × 1
- External trigger input signal × 1
- External event input × 1
- Readable counter × 1
- Count write buffer × 1
- Capture/compare reload register × 2
- Capture/compare match interrupt × 2
- Timer Output (TO) × 2
- Capture/compare match signal × 2
- Overflow interrupt × 1
- Encoder clear interrupt × 1

11.3 Configuration

Timer T is configured of the following hardware.

Table 11-1: Timer T Configuration

Item	Configuration
Counter	16-bit counter
Registers	TMTn capture/compare registers 0, 1 (TTnCCR0, TTnCCR1) TMTn counter read buffer register (TTnCNT) TMTn counter write buffer register (TTnTCW) TTnCCR0 buffer register, TTnCCR1 buffer register
Timer input pins	7 (TITn0, TITn1, TEVTTn, TTRGTn, TENCTn0, TENCTn1, TECRTn)**Note**
Timer output pins	2 (TOTn0, TOTn1)**Note**
Timer input signals	
Timer output signals	TTnEQC0, TTnEQC1
Control registers	TMTn control registers 0, 1 (TTnCTL0 to TTnCTL2) TMTn I/O control registers 0 to 2 (TTnIOC0 to TTnIOC3) TMTn option registers 0, 1 (TTnOPT0 to TTnOPT2)
Interrupts	Compare match interrupt (INTTTnCC0, INTTTnCC1) Overflow interrupt (INTTTnOV) Encoder clear interrupt (INTTTnEC)

Note: Alternate-function pins

Remark: n = 0, 1

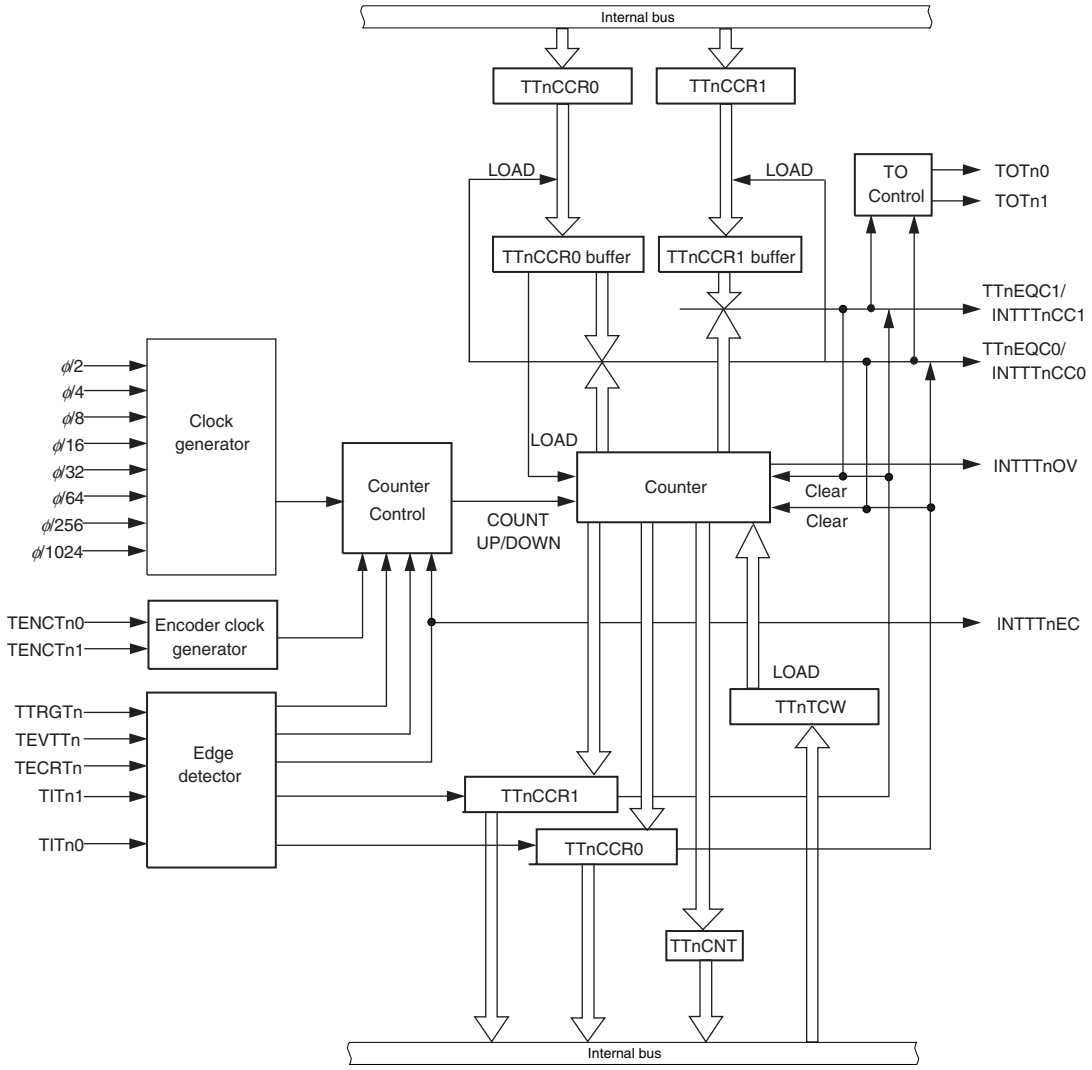
Table 11-2: List of Timer T Registers

Address	Register Name	Symbol	R/W	Manipulable Bit Units			After Reset
				1	8	16	
FFFFF690H	TMT0 control register 0	TT0CTL0	R/W	×	×		00H
FFFFF691H	TMT0 control register 1	TT0CTL1	R/W	×	×		00H
FFFFF692H	TMT0 control register 2	TT0CTL2	R/W	×	×		00H
FFFFF693H	TMT0 I/O control register 0	TT0IOC0	R/W	×	×		00H
FFFFF694H	TMT0I/O control register 1	TT0IOC1	R/W	×	×		00H
FFFFF695H	TMT0 I/O control register 2	TT0IOC2	R/W	×	×		00H
FFFFF696H	TMT0 I/O control register 3	TT0IOC3	R/W	×	×		00H
FFFFF697H	TMT0 option register 0	TT0OPT0	R/W	×	×		00H
FFFFF698H	TMT0 option register 1	TT0OPT1	R/W	×	×		00H
FFFFF699H	TMT0 option register 2	TT0OPT2	R/W	×	×		00H
FFFFF69AH	TMT0 capture/compare register 0	TT0CCR0	R/W			×	0000H
FFFFF69CH	TMT0 capture/compare register 1	TT0CCR1	R/W			×	0000H
FFFFF69EH	TMT0 counter read buffer register	TT0CNT	R			×	0000H**Note**
FFFFF990H	TMT0 counter write buffer register	TT0TCW	R/W			×	0000H
FFFFF6A0H	TMT1 control register 0	TT1CTL0	R/W	×	×		00H
FFFFF6A1H	TMT1 control register 1	TT1CTL1	R/W	×	×		00H
FFFFF6A2H	TMT1 control register 2	TT1CTL2	R/W	×	×		00H
FFFFF6A3H	TMT1 I/O control register 0	TT1IOC0	R/W	×	×		00H
FFFFF6A4H	TMT1I/O control register 1	TT1IOC1	R/W	×	×		00H
FFFFF6A5H	TMT1 I/O control register 2	TT1IOC2	R/W	×	×		00H
FFFFF6A6H	TMT1 I/O control register 3	TT1IOC3	R/W	×	×		00H
FFFFF6A7H	TMT1 option register 0	TT1OPT1	R/W	×	×		00H
FFFFF6A8H	TMT1 option register 1	TT1OPT1	R/W	×	×		00H
FFFFF6A9H	TMT1 option register 2	TT1OPT2	R/W	×	×		00H
FFFFF6AAH	TMT1 capture/compare register 0	TT1CCR0	R/W			×	0000H
FFFFF6ACH	TMT1 capture/compare register 1	TT1CCR1	R/W			×	0000H
FFFFF6AEH	TMT1 counter read buffer register	TT1CNT	R			×	0000H**Note**
FFFFF9A0H	TMT1 counter write buffer register	TT1TCW	R/W			×	0000H

Note: When TTnCE = 0

Remark: n = 0, 1

Figure 11-1: Block Diagram of Timer T



Remark: n = 0, 1
m = 0, 1

(1) TMTn capture/compare register 0 (TTnCCR0)

The TTnCCR0 register is a 16-bit register that functions both as a capture register and as a compare register.
This register can be read and written in 16-bit units only.
Reset input clears this register to 0000H.

Figure 11-2: TMTn Capture/Compare Register 0 (TTnCCR0)

After reset: 0000H R/W Address: TT0CCR0 FFFFF69AH, TT1CCR0 FFFFF6AAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTnCCR0																

The capture and compare functions are as follows in each mode.

Table 11-3: Capture/Compare Functions in Each Mode

Operation Mode	Capture/Compare Setting of TTnCCR0 Register	Rewriting Method during Compare	Counter Clear Function
Interval mode	Compare only	Anytime write	Compare match
External event count mode	Compare only	Anytime write	Compare match
External trigger pulse output mode	Compare only	Batch write (Reload)	Compare match
One-shot pulse mode	Compare only	Anytime write	Compare match
PWM mode	Compare only	Batch write (Reload)	Compare match
Free-running mode	Capture/compare selectable	Anytime write	-
Pulse width measurement mode	Capture only	-	External input (TITn0 pin)
Triangular wave PWM mode	Compare only	Batch write (Reload)**Note 1**	Compare match
Encoder compare mode	Compare only	Anytime write	Depends on set condition**Note 2**
Offset trigger generation mode	Capture only	-	External input (TITn0 pin)

Notes: 1. The batch write reload timing is the counter underflow timing only.

2. The condition is set with the TTnECM0 and TTnECM1 bits of the TTnCTL2 register.

Remark: n = 0, 1

- Use as compare register
  When TTnCE = 1, the TTnCCR0 register rewrite method differs according to the operation mode.
  Refer to **Table 11-3: Capture/Compare Functions in Each Mode**.
  (For details about the compare register rewrite operation, refer to **11.5.2 Method for writing to compare register**.)

- Use as capture register
  The counter value is saved to the TTnCR0 register upon TITn0 pin input edge detection. The function to clear counters following capture differs according to the operation mode.
  Refer to **Table 11-3: Capture/Compare Functions in Each Mode**.

(2) TMTn capture/compare register 1 (TTnCCR1)

The TTnCCR1 register is a 16-bit register that functions both as a capture register and a compare register.
This register can be read and written in 16-bit units only.
Reset input clears this register to 0000H.

Figure 11-3: TMTn Capture/Compare Register 1 (TTnCCR1)

After reset: 0000H R/W Address: TT0CCR1 FFFFF69CH, TT1CCR1 FFFFF6ACH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTnCCR1																

The capture/compare functions in each operation mode are as follows.

Table 11-4: Capture/Compare Functions in Each Mode

Operation Mode	Capture/Compare Setting of TTnCCR1 Register	Rewriting Method during Compare	Counter Clear Function
Interval mode	Compare only	Anytime write	-
External event count mode	Compare only	Anytime write	-
External trigger pulse output mode	Compare only	Batch write (Reload)	-
One-shot pulse mode	Compare only	Anytime write	-
PWM mode	Compare only	Batch write (Reload)	-
Free-running mode	Capture/compare selectable	Anytime write	-
Pulse width measurement mode	Capture only	-	External input (TITn1 pin)
Triangular wave PWM mode	Compare only	Batch write (Reload)**Note 1**	-
Encoder compare mode	Compare only	Anytime write	Depends on set conditions**Note 2**
Offset trigger generation mode	Compare only	Batch write (Reload)**Note 3**	-

Notes: 1. The batch write reload timing is the counter underflow occurrence timing only.

2. The conditions are set with bits TTnECM0 and TTnECM1 of TTnCTL2 register.

3. The batch write reload timing is the counter's 0000H clear timing only.

Remark: n = 0, 1

- Use as compare register
  When TTnCE = 1, the write method of register TTnCCR1 differs according to the operation mode.
  Refer to **Table 11-4: Capture/Compare Functions in Each Mode**.
  (For details about the compare register rewrite operation, refer to **11.5.2 Method for writing to compare register**.)

- Use as capture register
  The counter value upon TITn1 pin input edge detection is saved to the TTnCCR1 register. The function to clear the counter following capture also differs according to the mode.
  Refer to **Table 11-4: Capture/Compare Functions in Each Mode**.

(3) TMTn counter write buffer register (TTnTCW)

The TTnTCW register is a write buffer register that can write the counter value.
The setting value is valid only in the encoder compare mode, encoder capture mode. In all other modes, the setting value is invalid.
This register can be read and written in 16-bit units.
Reset input clears this register to 0000H.

Remark: When TTnECC of register TTnCTL2 = 0, the setting value is loaded to the counter when the TTnCE bit is set (to 1). (When TTnECC = 1, the counter holds its value, so it is not reloaded.)

Figure 11-4: TMTn Counter Write Buffer Register (TTnTCW)

After reset: 0000H R/W Address: TT0TCW FFFFF990H, TT1TCW FFFFF9A0H

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

(4) TMTn counter read buffer register (TTnCNT)

The TTnCNT register is a read buffer register that can read the counter value.
This register can be read in 16-bit units only.
Reset input clears this register to 0000H.

Remark: When, in the encoder compare mode, encoder capture mode, the value of the TTnCE bit is changed from “1” to “0”, the value that can be read by the TTnCNT register differs according to the following conditions.

- When bit TTnECC of the TTnCTL2 register = 0, 0000H can be read.
- When bit TTnECC = 1, the value held when bit TTnCE was cleared to "0" can be read.

Figure 11-5: TMTn Counter Read Buffer Register (TTnCNT)

After reset: 0000H R/W Address: TT0CNT FFFFF69EH, TT1CNT FFFFF6AEH

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

11.4 Control Registers

(1) TMTn control register 0 (TTnCTL0)

TTnCTL0 is an 8-bit register that controls the operation of TMTn.
This register can be read and written in 8-bit or 1-bit units.
Reset input clears this register to 00H. Reset input clears this register to 00H.
When TTnCE = 1, only the TTnCE bit of the TTnCTL0 register can be changed. Perform write access to the other bits using the same values.

Figure 11-6: TMTn Control Register 0 (TTnCTL0) (1/2)

After reset: 00H R/W Address: TT0CTL0 FFFFF690H, TT1CTL0 FFFFF6A0H

	7	6	5	4	3	2	1	0
TTnCTL0 (n = 0, 1)	TTnCE	0	0	0	0	TTnCKS2	TTnCKS1	TTnCKS0

TRnCE	TMTn Operation Control
0	Internal operating clock operation disabled (TMTn reset asynchronously)
1	Internal operating clock operation enabled

When bit TTnCE is set to “0”, the internal operation clock of TMTn stops (fixed to low level), and TMTn is reset asynchronously.
When bit TTnCE is set to “1”, the internal operation of TMTn is enabled from when bit TTnCE was set to “1” and count-up is performed. The time until count-up is as listed in Table TMTn Count Clock and Time Until Count-Up.

Remarks: 1. In the encoder compare mode, encoder capture mode, the functions that are reset when TTnCE = 0 and TTnECC = 1 are as follows.

- Compare match detector (interrupt output low level)
- Timer output (Output inactive level)
- Edge detector for other than pins TENCTn0, TENCTn1, and TECRTn

2. The following functions are not reset.

- Counter
- Flags in TTnOPT1 register
- TTnCCR0 buffer, TTnCCR1 buffer register, counter read buffer register
- TENCTn0, TENCTn1, TECRTn pin edge detector

3. In modes other than the above, (in which TTnECC is fixed to 0), the functions that are reset by TTnCE = 0 are as follows.

- Internal registers other than registers that can be written from the CPU, and internal latch circuits

Remark: n = 0, 1

Figure 11-6: TMTn Control Register 0 (TTnCTL0) (2/2)

TTnCKS2	TTnCKS1	TTnCKS0	Internal Count Clock Selection
0	0	0	$f_{XX}/2$
0	0	1	$f_{XX}/4$
0	1	0	$f_{XX}/8$
0	1	1	$f_{XX}/16$
1	0	0	$f_{XX}/32$
1	0	1	$f_{XX}/64$
1	1	0	$f_{XX}/256$
1	1	1	$f_{XX}/1024$

Table 11-5: TMTn Count Clock and Count Delay

Count Clocks	TTnCKS2	TTnCKS1	TTnCKS0	Count Delay	
				Minimum	Maximum
$f_{XX}/2$	0	0	0	3 base clocks	4 base clocks
$f_{XX}/4$	0	0	1		
$f_{XX}/8$	0	1	0		
$f_{XX}/16$	0	1	1	4 base clocks	5 base clocks + 1 count clock
$f_{XX}/32$	1	0	0		
$f_{XX}/64$	1	0	1		
$f_{XX}/256$	1	1	0		
$f_{XX}/1024$	1	1	1		

Remarks: **1.** f_{XX}: System clock

2. f_{TMTn}: Base clock of TMTn ($f_{TMTn} = f_{XX}/2$)

3. n = 0, 1

(2) TMTn control register 1 (TTnCTL1)

The TTnCTL1 register is an 8-bit register that controls the operation of TMTn.
This register can be read and written in 8-bit or 1-bit units.
Reset input clears this register to 00H.
Set the TTnCTL1 register when TTnCE = 0. When TTnCE = 1, the bits other than bit TTnEST (TTnEEE, TTnMD3 to TTnMD0, TTnSYE) can be write accessed using the same value.

Caution: In the one-shot pulse mode and external trigger pulse output mode, write access using "1", the same value as that of bit TTnEST, functions as one trigger.

Figure 11-7: TMTn Control Register 1 (TTnCTL1) (1/2)

After reset: 00H R/W Address: TR0CTL1 FFFFF691H, TR1CTL1 FFFFF6A1H

	7	6	5	4	3	2	1	0
TTnCTL1 (n = 0, 1)	0	TTnEST	TTnEEE	0	TTnMD3	TTnMD2	TTnMD1	TTnMD0

TTnEST	Software Trigger Control
0	No operation
1	Enable software trigger control
• In one-shot pulse mode (One-shot pulse software trigger) Can be made to function as a software trigger by setting TTnETS to 1 when TTnCE = 1. Always write TTnEST = 1 when TTnCE = 1. • In external trigger pulse output mode (Pulse output software trigger) **Remark:** "0" is always read out from the TTnEST bit.	

TTnEEE	Count Clock Selection
0	Use of clock selected with bits TTnCKS2 to TTnCKS0 of TTnCTL0 register
1	Use of external clock (TEVTTn pin input edge)
Specification of the valid edge when TTnEEE = 1 (external clock: TEVTTn pin) is set with bits TTnEES1 and TTnEES0 of TTnIOC2 register.) **Remark:** The setting of bit TTnEEE is invalid in the external event count mode, encoder compare mode, encoder capture mode, encoder capture/compare mode.	

Caution: Rewrite the TTnEEE bit only when TTnCE = 0. (The same value can be written when TTnCE = 1.) The operation is not guaranteed if rewriting is performed when TTnCE = 1. If rewriting was mistakenly performed, set TTnCE = 0 and then set the bit again.

Remark: n = 0, 1

Figure 11-7: TMTn Control Register 1 (TTnCTL1) (2/2)

TTnMD3	TTnMD2	TTnMD1	TTnMD0	Timer Mode
0	0	0	0	Interval mode
0	0	0	1	External event count mode
0	0	1	0	External trigger pulse output mode
0	0	1	1	One-shot pulse mode
0	1	0	0	PWM mode
0	1	0	1	Free-running mode
0	1	1	0	Pulse width measurement mode
0	1	1	1	Triangular wave PWM mode
1	0	0	0	Encoder compare mode
1	1	0	0	Offset trigger generation mode
Other than above				Setting prohibited

Caution: Rewrite the TTnMD3 to TTnMD0 bits only when TTnCE = 0. (The same value can be written when TTnCE = 1.) The operation is not guaranteed if rewriting is performed when TTnCE = 1. If rewriting was mistakenly performed, set TTnCE = 0.

Remark: n = 0, 1

(3) TMTn control register 2 (TTnCTL2)

The TTnCTL2 register is an 8-bit register that controls the operation of TMTn.
This register can be read and written in 8-bit or 1-bit units.
Reset input clears this register to 00H.
The settings of the TTnCTL2 register are valid only in the encoder compare mode. The settings of this register are invalid in all other modes.
Set the TTnCTL2 register when TTnCE = 0. When TTnCE = 1, write access to the TTnCTL2 register can be performed with the same value.

Figure 11-8: TMTn Control Register 2 (TTnCTL2) (1/2)

After reset: 00H R/W Address: TR0CTL1 FFFFF692H, TR1CTL1 FFFFF6A2H

	7	6	5	4	3	2	1	0
TTnCTL2 (n = 0, 1)	TTnECC	0	0	TTnLDE	TTnECM1	TTnECM0	TTnUDS1	TTnUDS0

TTnECC	Selection of Initialization/Hold of Counter Value when TTnCE = 0
0	Initialize counter value when TTnCE = 0
1	Hold counter value when TTnCE = 0
When TTnECC = 0, setting TTnCE = 0 causes the counter to be reset to FFFFH, the capture registers (TTnCCR0/TTnCCR1) to be reset to 0000H, and the encoder-dedicated flags (TTnEOF/TTnEUF/TTnESF) to be reset to 0. When TTnECC = 0, the value of the TTnTCW register is loaded to the counter when TTnCE is set from 0 to 1. When TTnECC = 1, setting TTnCE = 0 causes the values of the counter, capture registers (TTnCCR0/TTnCCR1), and encoder dedicated flags (TTnEOF/TTnEUF/TTnESF) to be held. When TTnECC = 1, the value of the TTnTCW register is not loaded to the counter. **Remark:** The setting of bit TTnECC is valid in the encoder compare mode.		

TTnLDE	Encoder Load Enable
0	Disable transfer of compare setting value to counter
1	Enable transfer of compare setting value (TTnCCR0) to counter when underflow occurs
Remark:	The setting of bit TTnLDE is valid in the encoder compare mode and bits TTnECM1 and TTnECM0 are set as follows. • TTnECM1 = 0, TTnECM0 = 0 or 1

TTnECM1	Encoder Clear Mode on Match of Counter and TTnCCR1 Register
0	No clear condition
1	When the counter and TTnCCR1 register match, clear the counter if the next count is a down count (TTnESF = 1)
Remark:	The setting of bit TTnECM1 is valid in the encoder compare mode.

Remark: n = 0, 1

Figure 11-8: TMTn Control Register 2 (TTnCTL2) (2/2)

TTnECM0	Encoder Clear Mode on Match of Counter and TTnCCR0 Register
0	No clear condition
1	When the counter and TTnCCR0 register match, clear the counter if the next count is a down count (TTnESF = 0)
Remark:	The setting of bit TTnECM0 is valid in the encoder compare mode.

TTnUDS1	TTnUDS0	Encoder Operation Mode
0	0	Upon detection of the valid edge of the A phase of encoder input (TENCTn0 pin), the following count operation is performed in the B phase of encoder input. • When "high", count down. • When "low", count up.
0	1	Count up upon detection of valid edge of A phase of encoder input (TENCTn0 pin). Count down upon detection of valid edge of B phase of encoder input (TENCTn1 pin).
1	0	Count up at rising edge of A phase of encoder input (TENCTn0 pin). Count down at falling edge of A phase of encoder input. However, count operation is performed only when B phase of encoder input (TENCTn1 pin) is “low”.
1	1	Detection of both edges of phase A of encoder input (TENCTn0 pin)/phase B of encoder input (TENCTn1 pin). Judgment of count operation based on combination of detection edge and input level.

Remarks: 1. When bits TTnUDS1 and TTnUDS0 are set to 10B or 11B, the settings of bits TTnEIS1 and TTnEIS0 of the TTnIOC3 register are invalid, and these bits are fixed to the setting for detection of both edges.

2. n = 0, 1

(4) TMTn I/O control register 0 (TTnIOC0)

The TTnIOC0 register is an 8-bit register that controls timer output (TOTn0 and TOTn1 pins). This register can be read and written in 8-bit or 1-bit units.
Reset input clears this register to 00H.
Set the TTnIOC0 register when TTnCE = 0. When TTnCE = 1, write access to the TTnIOC0 register can be performed using the same value.

Figure 11-9: TMTn I/O Control Register 0 (TTnIOC0)

After reset: 00H R/W Address: TR0IOC0 FFFFF693H, TR1IOC0 FFFFF6A3H

	7	6	5	4	3	2	1	0
TTnIOC0 (n = 0, 1)	0	0	0	0	TTnOL1	TTnOE1	TTnOL0	TTnOE0

TTnOLm	Timer Output Level Setting (TOTnm pin)
0	Normal output (Low level, when output is inactive.)
1	Inverted output (High level, when output is inactive.)

TTnOEm	Timer Output Control (TOTnm pin)
0	Timer output disabled (TOTnm pin output is fixed to inactive level.)
1	Timer output enabled (A pulse can be output from the TOTnm pin.)

Remark: n = 0, 1
m = 0, 1

(5) TMTn I/O control register 1 (TTnIOC1)

The TTnIOC1 register is an 8-bit register that controls the valid edge of capture input (TITn1 and TITn0 pins).

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

Reset input clears this register to 00H.

Set the TTnIOC1 register when TTnCE = 0. When TTnCE = 1, write access to the TTnIOC1 register can be performed using the same value.

Figure 11-10: TMTn I/O Control Register 1 (TTnIOC1)

After reset: 00H R/W Address: TR0IOC1 FFFFF694H, TR1IOC1 FFFFF6A4H

	7	6	5	4	3	2	1	0
TTnIOC1 (n = 0, 1)	0	0	0	0	TTnIS3	TTnIS2	TTnIS1	TTnIS0

TTnIS3	TTnIS2	Capture Input (TITn1) Valid Edge Setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection

Capture operation is performed and capture interrupt (INTTTnCC1) is output upon edge detection.

Remark: The setting of bits TTnIS3 and TTnIS2 are valid in the free-running mode and pulse width measurement mode.

TTnIS1	TTnIS0	Capture Input (TITn0) Valid Edge Setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection

Capture operation is performed and capture interrupt (INTTTnCC0) is output upon edge detection.

Remark: The setting of bits TTnIS1 and TTnIS0 are valid in the free-running mode and pulse width measurement mode.

Remark: n = 0, 1

(6) TMTn I/O control register 2 (TTnIOC2)

The TTnIOC2 register is an 8-bit register that controls the valid edge of external event count input (TEVTTn pin) and external trigger input (TTRGTn pin).
This register can be read and written in 8-bit or 1-bit units.
Reset input clears this register to 00H.
Set the TTnIOC2 register when TTnCE = 0. When TTnCE = 1, write access to the TTnIOC2 register can be performed using the same value.

Figure 11-11: TMTn I/O Control Register 2 (TTnIOC2)

After reset: 00H R/W Address: TR0IOC2 FFFFF695H, TR1IOC2 FFFFF6A5H

	7	6	5	4	3	2	1	0
TTnIOC2 (n = 0, 1)	0	0	0	0	TTnEES1	TTnEES0	TTnETS1	TTnETS0

TT1EES1	TT1EES0	External Event Counter Input (TEVTTn) Valid Edge Setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection
Remark:	The settings of bits TTnEES1 and TTnEES0 are valid in the external event count mode, or when bit TTnEEE of the TTnCTL1 register = 1.	

TT1ETS1	TT1ETS0	External Trigger Input (TTRGTn) Valid Edge Setting
0	0	No edge detection (capture operation invalid)
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both, rising and falling edge detection
Remark:	The settings of bits TTnETS1 and TTnETS0 are valid in the external trigger pulse output mode and the one-shot pulse mode.	

Remark: n = 0, 1

(7) TMTn I/O control register 3 (TTnIOC3)

The TTnIOC3 register is an 8-bit register that controls the valid edge of encoder clear input (TECRTn pin) and encoder input (TENCTn1 and TENCTn0 pins).
This register can be read and written in 8-bit or 1-bit units.
Reset input clears this register to 00H.
Set the TTnIOC3 register when TTnCE = 0. When TTnCE = 1, write access to the TTnIOC2 register can be performed using the same value.

Figure 11-12: TMTn I/O Control Register 3 (TTnIOC3) (1/2)

After reset: 00H R/W Address: TR0IOC3 FFFFF696H, TR1IOC3 FFFFF6A6H

	7	6	5	4	3	2	1	0
TTnIOC3 (n = 0, 1)	TTnSCE	TTnZCL	TTnBCL	TTnACL	TTnECS1	TTnECS0	TTnEIS1	TTnEIS0

TTnSCE	Selects the encoder counter clear method
0	Clear upon detection of edge of TECRTn pin
1	Clear upon match of clear condition level
When TTnSCE = 1, the counter is cleared to 0000H if all the conditions set with bits TTnZCL, TTnBCL, and TTnACL are matched. When TTnSCE = 1, the settings of bits TTnECS1 and TTnECS0 are invalid, so no encoder clear interrupt (INTTTnEC) is output. When TTnSCE = 0, the settings of bits TTnZCL, TTnBCL, and TTnACL are invalid. The settings of bits TTnECS1 and TTnECS0 become valid, and the encoder clear interrupt (INTTTnEC) is output. **Caution: When TTnSCE = 1, be sure to set bits TTnUDS1, and TTnUDS0 of the TTnCTL2 register to 10B or 11B.**		

TTnZCL	Sets the clear level for the Z phase of encoder input (TECRTn pin)
0	Clear condition = Low level
1	Clear condition = High level
Remark:	The TTnZCL bit is valid when TTnSCE = 1.

TTnBCL	Sets the clear level for the B phase of encoder input (TENCTn1 pin)
0	Clear condition = Low level
1	Clear condition = High level
Remark:	The TTnBCL bit is valid when TTnSCE = 1.

TTnACL	Sets the clear level for the A phase of encoder input (TENCTn0 pin)
0	Clear condition = Low level
1	Clear condition = High level
Remark:	The TTnACL bit is valid when TTnSCE = 1.

Remark: n = 0, 1

Figure 11-12: TMTn I/O Control Register 3 (TTnIOC3) (2/2)

TTnECS1	TTnECS0	Set the valid edge of encoder clear input (TECRTn pin)
0	0	No edge detection
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both rising and falling edge detection

The encoder clear interrupt (INTTTnEC) is output upon detection of the valid edge set with bits TTnECS1, TTnECS0.

Caution: When TTnSCE = 1, the encoder clear interrupt (INTTTnEC) is not output.

Remark: Bits TTnECS1 and TTnECS0 are valid in the encoder compare mode and when TTnSCE = 0.

TTnEIS1	TTnEIS0	Set the valid edge of the encoder input signal (TENCTn1/TENCTn0 pins)
0	0	No edge detection
0	1	Rising edge detection
1	0	Falling edge detection
1	1	Both rising and falling edge detection

Remark: Bits TTnEIS1 and TTnEIS0 are valid when bits TTnUDS1 and TTnUDS0 of register TTnCTL2 are “00B” or “01B”.

Remark: n = 0, 1

(8) TMTn option register 0 (TTnOPT0)

The TTnOPT0 register is an 8-bit register that sets the capture/compare operation and detects overflow.

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

Reset input clears this register to 00H.

Set the bits of the TTnOPT0 register other than TTnOVF when TTnCE = 0. When TTnCE = 1, write access of bits of the TTnOPT0 register other than TTnOVF can be performed using the same value.

Figure 11-13: TMTn Option Register 0 (TTnOPT0)

After reset: 00H R/W Address: TR0IOC3 FFFFF697H, TR1IOC3 FFFFF6A7H

	7	6	5	4	3	2	1	0
TTnOPT0 (n = 0, 1)	0	0	TTnCCS1	TTnCCS0	0	0	0	TTnOVF

TTnCCS1	Specifies the operation mode of register TTnCCR1
0	Operation as compare register
1	Operation as capture register
Remark:	The setting of bit TTnCCS1 is valid in the free-running mode only.

TTnCCS0	Specifies the operation mode of register TTnCCR0
0	Operation as compare register
1	Operation as capture register
Remark:	The setting of bit TTnCCS0 is valid in the free-running mode only.

TTnOVF	Flag that indicates TMTn overflow
0	No overflow occurrence after timer restart or flag reset
1	Overflow occurrence

In the free-running mode, pulse width measurement mode, and offset trigger generation mode, if the counter value is counted up from FFFFH, overflow occurs, the TTnOVF flag is set (1), and the counter is cleared to 0000H. The counter is also cleared by writing 0. At the same time that the TTnOVF flag is set (1), an overflow interrupt (INTTTnOV) occurs. If 0 is written to the TTnOVF flag, or if TTnECC = 0 and TTnCE = 0 are set, the counter is cleared.

Remark: Overflow does not occur during compare match & clear operation for counter value FFFFH and compare value FFFFH.

Cautions: 1. If overflow occurs in the encoder compare mode, the encoder-dedicated overflow flag (TTnEOF) is set, and the overflow flag (TTnOVF) is not set. At this time, the overflow interrupt (INTTTnOV) is output.

2. When TTnOVF = 1, the TTnOVF flag is not cleared even if the TTnOVF flag and TTnOPT0 register are read.

3. The TTnOVF flag can be read and written, but even if 1 is written to the TTnOVF flag from the CPU, this is invalid.

Remark: n = 0, 1

(9) TMTn option register 1 (TTnOPT1)

The TTnOPT1 register is an 8-bit register that detects encoder-dedicated underflow, overflow, and counter up/down operation.

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

Reset input clears this register to 00H.

The setting of the TTnOPT1 register is valid only in the encoder compare mode. In all other modes, the setting value is invalid.

Figure 11-14: TMTn Option Register 1 (TTnOPT1) (1/2)

After reset: 00H R/W Address: TR0IOC3 FFFFF698H, TR1IOC3 FFFFF6A8H

	7	6	5	4	3	2	1	0
TTnOPT1 (n = 0, 1)	0	0	0	0	0	TTnEUF	TTnEOF	TTnESF

TTnEUF	Indication of Encoder Underflow
0	No underflow indicated
1	Indicates counter underflow in the encoder compare mode

If the counter value is counted down from 0000H, underflow occurs, the OVF flag is set (to 1), and the counter is set to FFFFH. When the TTnEUF flag is set (to 1), an overflow interrupt (INTTTnOV) occurs at the same time.

The TTnEUF flag is cleared (to 0) under the following conditions.

- When 0 is written by CPU instruction
- When TTnCE = 0 is set while TTnECC = 0

Cautions: 1. The TTnEUF flag is not cleared even if it is read.

2. The TTnEUF flag can be read and written, but even if 1 is written to the TTnEUF flag, this is invalid.

Remark: When bit TTnECC of the TTnCTL2 register is 1, the flag status is held even if the value of bit TTnCE is changed from 1 to 0.

Remark: n = 0, 1

Figure 11-14: TMTn Option Register 1 (TTnOPT1) (2/2)

TTnEOF	Indication of Encoder Overflow
0	No overflow indicated
1	Indicates counter overflow in the encoder compare mode

If the counter value is counted up from FFFFH, overflow occurs, the OVF flag is set (1), and the counter is cleared to 0000H. At the same time that the TTnEOF flag is set (1), an overflow interrupt (INTTTnOV) occurs. However, the TTnOVF flag is not set (to 1).

The TTnEOF flag is cleared (0) under the following conditions.

- When 0 is written by CPU instruction
- When TTnCE = 0 is set while TTnECC = 0

Cautions: 1. The TTnEOF flag is not cleared even if it is read.

2. The TTnEOF flag can be read and written, but even if 1 is written to the TTnEOF flag from the CPU, this is invalid.

Remark: When bit TTnECC of the TTnCTL2 register is 1, the flag status is held even if the value of bit TTnCE is changed from 1 to 0.

TTnESF	Indication of Encoder Count Direction
0	Indicates the up count operation of the counter in the encoder compare mode.
1	Indicates the down count operation of the counter in the encoder compare mode.

The TTnESF flag is cleared (to 0) under the following conditions.

- When TTnCE = 0 is set while TTnECC = 0

Remark: When bit TTnECC of the TTnCTL2 register is 1, the flag status is held even if the value of bit TTnCE is changed from 1 to 0.

Remark: n = 0, 1

(10) TMTn option register 2 (TTnOPT2)

The TTnOPT2 register is an 8-bit register that indicates the reload request status when performing write access to compare registers using the reload method.
This register can only be read in 8-bit or 1-bit units.
Reset input clears this register to 00H.
The read contents of the TTnOPT2 register are valid only in the external trigger pulse mode, PWM MODE, and offset trigger generation using the reload method. In all other modes, the read contents are 0.

Figure 11-15: TMTn Option Register 2 (TTnOPT2)

After reset: 00H R/W Address: TR0IOC3 FFFFF699H, TR1IOC3 FFFFF6A9H

	7	6	5	4	3	2	1	0
TTnOPT2 (n = 0, 1)	0	0	0	0	0	0	0	TTnRSF

TTnRSF	Reload Status Flag
0	No reload request, or reload completed
1	Reload request was output
It indicates that the data to be transferred next is held pending in the TTnCCR0 and TTnCCR1 registers. The TTnRSF flag is set (1) by writing to the TTnCCR1 register, and it is cleared (0) upon reload completion. **Caution: When TTnRSF = 1, do not perform write access to the TTnCCR0 and TTnCCR1 registers.**	

Remark: n = 0, 1

11.5 Basic Operation

11.5.1 Basic counter operation

This section describes the basic operation of the counter. For details, refer to chapter **11.6 Operation in Each Mode**.

(1) Counter start operation

(a) Encoder compare mode

The count operation is controlled by the phases of pins TENCTn0 and TENCTn1.
When TTnCE = 0 and TTnECC = 0, the counter is initialized by the TTnTCW register and the count operation is started. (The setting value of the TTnTCW register is loaded to the counter at the timing when TTnCE changes from 0 to 1.)

(b) Triangular wave PWM MODE

The counter starts counting from initial value FFFFH.
It counts up FFFFH, 0000H, 0001H, 0002H, 0003H…
Following count up operation, the counter counts down upon a match with the TTnCCR0 register.

(c) Modes other than the above

The counter starts counting from initial value FFFFH.
It counts up FFFFH, 0000H, 0001H, 0002H, 0003H…

(2) Counter clear operation

There are the following five counter clear causes.

- Clear through match between counter value and compare setting value.
- Capture and clear through capture input
- Counter clear through encoder clear input (TECRTn pin)
- Counter clear through match with clear condition level
- Clear through clear signal input (TTnSYCI) for synchronization function during slave operation

Remark: n = 0, 1

Table 11-6: Counter Clear Operation

Operation Mode	Clear Cause		
	TTnCCR0	TTnCCR1	Other
Interval mode	Compare match	-	-
External event count mode	Compare match	-	-
External trigger pulse output mode	Compare match	-	External trigger (TTRGTn pin)
One-shot pulse mode	Compare match	-	-
PWM mode	Compare match	-	-
Free-running mode	-	-	-
Pulse width measurement mode	-	-	External input (TITn0 and TITn1 pins)
Triangular wave PWM mode	Compare match	-	-
Encoder compare mode	Depends on set conditions**Note**	Depends on set conditions**Note**	Pin TECRTn, clear condition level match
Offset trigger generation mode	-	-	External input (TITn0 pin)

Note: Conditions are set with bits TTnECM0 and TTnECM1 of the TTnCTL2 register.

(3) Counter reset and hold operations

In the encoder compare mode, counter value hold is controlled with bit TTnECC of the TTnCTL2 register.
If TTnCE = 0 is set when TTnECC = 0, the counter is reset to 0000H. The setting value of the TTnTCW register is loaded to the counter when TTnCE = 1 is set next.
If TTnCE = 0 is set when TTnECC = 1, the counter value is held as is. Counting resumes from the held value when TTnCE = 1 is set next.

(4) Counter read operation during counter operation

In TMT, the counter value can be read during count operation using the TTnCNT register.

Remark: n = 0, 1

(5) Overflow operation

Counter overflow occurs in the free-running mode, pulse width measurement mode, encoder compare mode and offset trigger generation mode.
Overflow occurs when the counter value changes from FFFFH to 0000H.
In the free-running mode, pulse width measurement mode, offset trigger generation mode, the overflow flag (TTnOVF) is set to 1 and an overflow interrupt (INTTTnOV) is output. At this time, the TTnEOF flag is not set.
In the encoder compare mode, the encoder dedicated overflow flag (TTnEOF) is set to 1 and an overflow interrupt (INTTTnOV) occurs. At this time, the TTnOVF flag is not set.

Under the following conditions, overflow does not occur.

- When the counter value changes from initial setting FFFFH to 0000H immediately after counting start
- When FFFFH is set to the compare register, and the counter is cleared to 0000H upon a match between the counter value and the compare setting value.
- When, in the pulse width measurement mode and offset trigger generation mode, capture operation is performed for counter value FFFFH, and the counter is cleared to 0000H.

(6) Underflow operation

Counter underflow occurs in the triangular wave PWM Mode and encoder compare mode.
Underflow occurs when the counter value changes from 0000H to FFFFH.
When underflow occurs in the triangular wave PWM mode, an overflow interrupt (INTTTnOV) occurs. At this time, the TTnOVF flag is not set.
In the encoder compare mode, the encoder dedicated underflow flag (TTnEUF) is set to 1, and an overflow interrupt (INTTTnOV) occurs.
Underflow does not occur during count down immediately following counter start.

(7) Description of interrupt signal operation

In TMT, the following interrupt signals are output.

Name	Occurrence Cause
INTTTnCC0	• Match between counter and setting value of TTnCCR0 register • Capture to TTnCCR0 register due to TITn0 pin input
INTTTnCC1	• Match between counter and setting value of TTnCCR1 register • Capture to TTnCCR1 register due to TITn1 pin input
INTTTnOV	Overflow and underflow occurrence
INTTTnEC**Note**	Counter clearing through TECRTn pin

Note: In the encoder compare mode, when TTnSCE = 0, an encoder clear interrupt (INTTTnEC) is output.

Remark: n = 0, 1

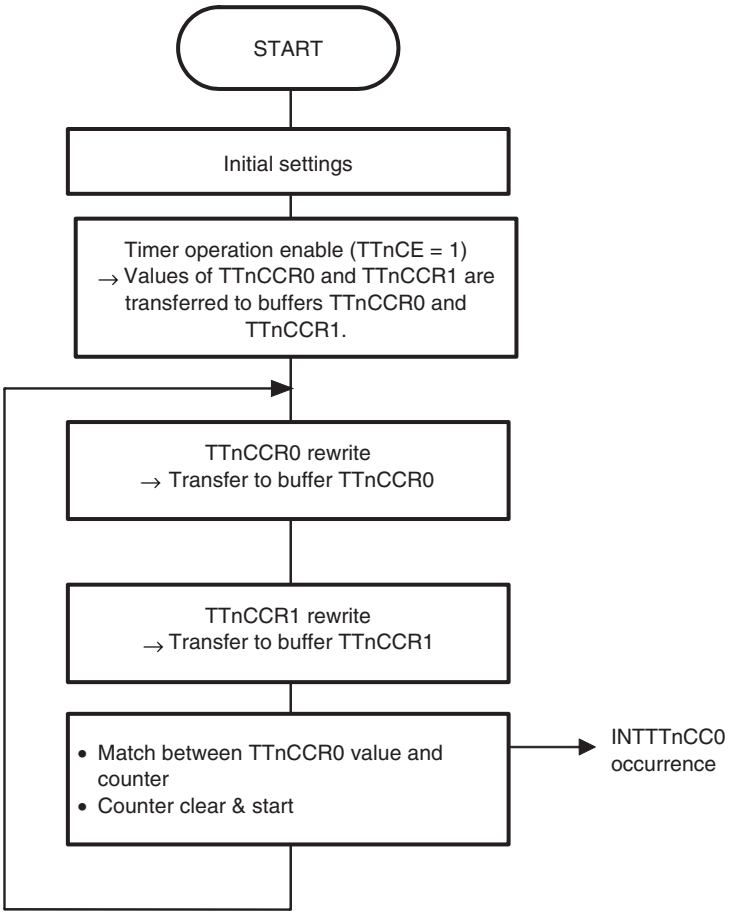
11.5.2 Method for writing to compare register

The TTnCCR0 and TTnCCR1 registers can be rewritten during timer operation (TTnCE = 1). There are two write modes (anytime write, reload), depending on the mode.

(1) Anytime rewrite method

When the TTnCCR0 and TTnCCR1 registers are written during timer operation, the write value is immediately transferred to the TTnCCR0 buffer register and TTnCCR1 buffer register and is used as the value to be compared with the counter.

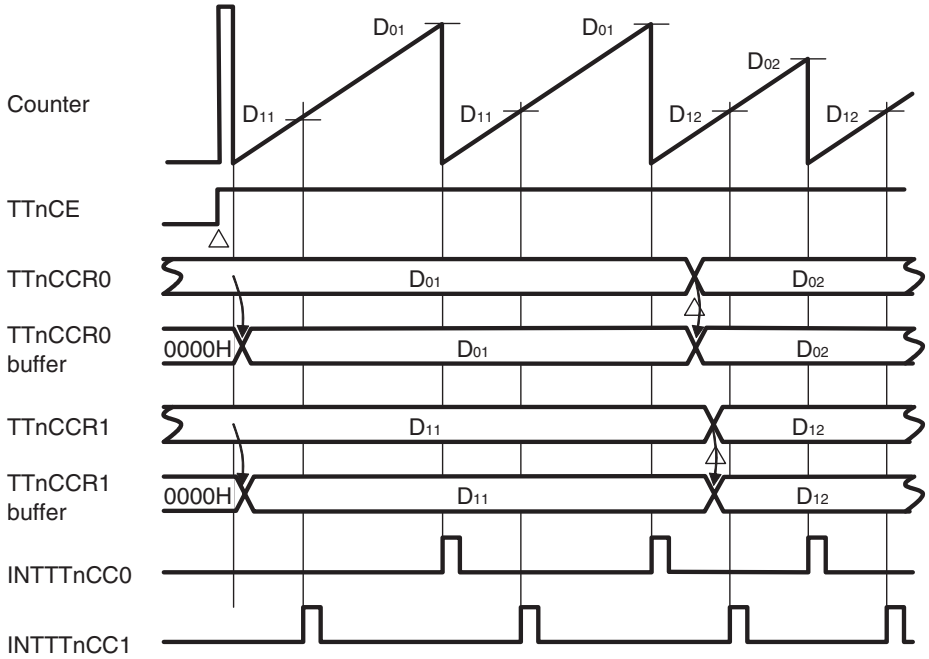
Figure 11-16: Basic Operation Flow for Anytime Rewrite



Remarks: 1. The interval mode is used as an example.

2. n = 0, 1

Figure 11-17: Basic Anytime Rewrite Operation Timing



Remarks: **1.** D_{01}, D_{02}: Setting values of TTnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Setting values of TTnCCR1 register (0000H to FFFFH)

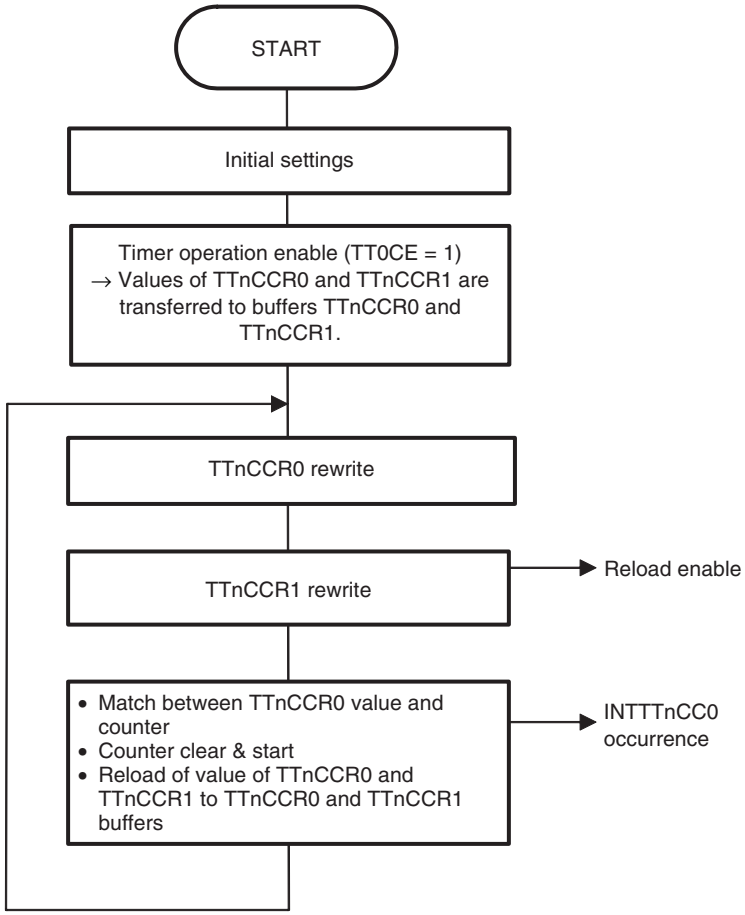
2. The interval mode is used as an example.

3. n = 0, 1

(2) Reload method (Batch rewrite)

When TTnCCR0, TTnCCR1 register write is performed during timer operation, the written value is used as the comparison value for the counter via the TTnCCR0 and TTnCCR1 buffer registers.
Under the reload method, rewrite the TTnCCR0 register before the TTnCCR0 register value is matched, and next, write to the TTnCCR1 register.
Then, when the TTnCCR0 register is matched or the counter is cleared to 0000H through external input, the values of the TTnCCR0 register and TTnCCR1 register are reloaded.
By writing to the TTnCCR1 register, the value becomes valid at the next reload timing.
Therefore, even if wishing to rewrite only the value of the TTnCCR0, rewrite the same value to the TTnCCR1 register to make the next reload valid.

Figure 11-18: Basic Operation Flow for Reload (Batch Rewrite)

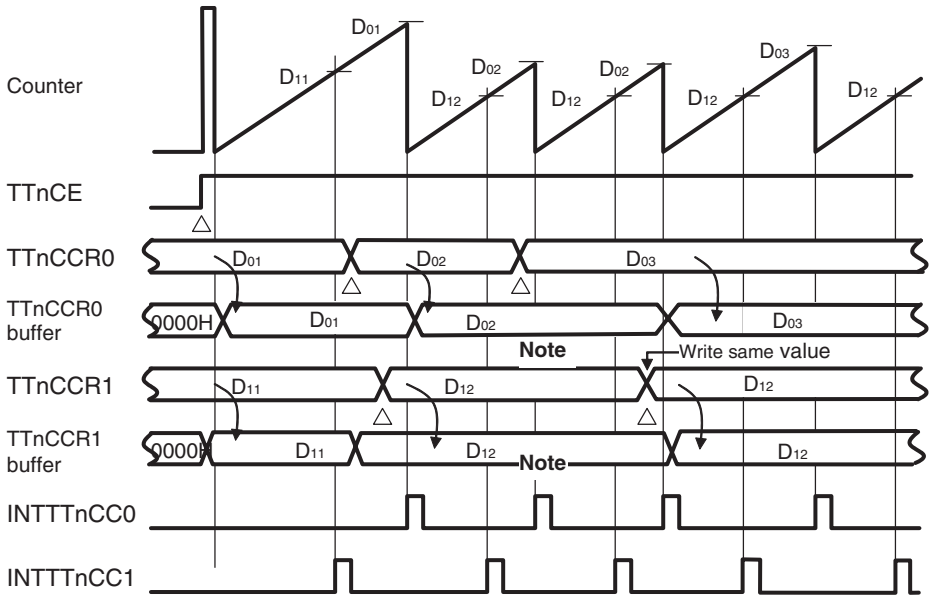


Caution: Rewrite to the TTnCCR1 register includes enabling reload. Therefore, rewrite the TTnCCR1 register after rewriting the TTnCCR0 register.

Remarks: 1. The PWM mode is used as an example.

2. n = 0, 1

Figure 11-19: Basic Reload Operation Timing



Note: Since the TTnCCR1 register is not written to, reloading is not performed even if TTnCCR0 is rewritten.

Remarks: **1.** D_{01}, D_{02}, D_{03}: Setting values of TTnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Setting values of TTnCCR1 register (0000H to FFFFH)

2. The PWM mode is used as an example.

3. n = 0, 1

Table 11-7: Capture/Compare Rewrite Methods in Each Mode

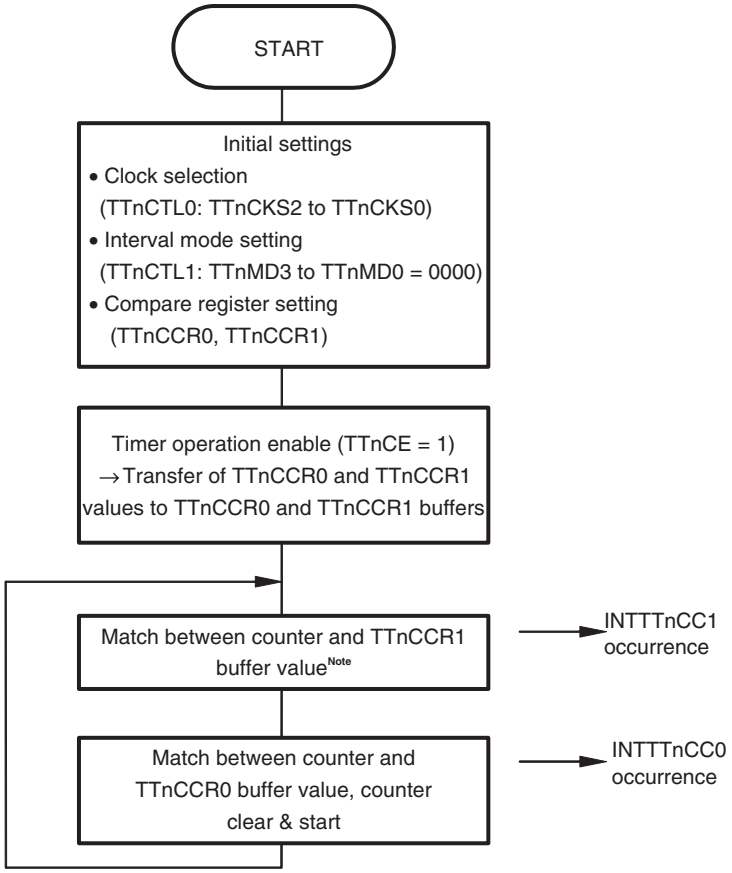
Operation Mode	Capture/Compare Rewrite Method	
	TTnCCR0	TTnCCR1
Interval mode	Compare only (Anytime write type)	
External event count mode		
External trigger pulse output mode	Compare only (Reload type)	
One-shot pulse mode	Compare only (Anytime write type)	
PWM mode	Compare only (Reload type)	
Free-running mode	Capture/compare selectable (When compare is selected, anytime write type)	
Pulse width measurement mode	Capture only	
Triangular wave PWM mode	Compare only (Reload type)	
Encoder compare mode	Compare only (Anytime write type)	
Offset trigger generation mode	Capture only	Compare only (Reload type)

11.6 Operation in Each Mode

11.6.1 Interval timer mode

In the interval timer mode, a compare match interrupt (INTTTnCC0) occurs and the counter is cleared upon a match between the setting value of the TTnCCR0 register and the counter value. The occurrence interval for this counter and TTnCCR0 register match interrupt becomes the interval time.
In the interval timer mode, the counter is cleared only upon a match between the counter and the value of the TTnCCR0 register. Counter clearing using the TTnCCR1 register is not performed.
However, the setting value of the TTnCCR1 is compared to the counter value transferred to the TTnCCR1 buffer register and a compare match interrupt (INTTTnCC1) is output.
The TTnCCR0 and TTnCCR1 registers can be rewritten using the anytime write method, regardless of the value of bit TTnCE.
Pins TOTn0 and TOTn1 are toggle output controlled when bits TTnOE0 and TTnOE1 are set to 1.

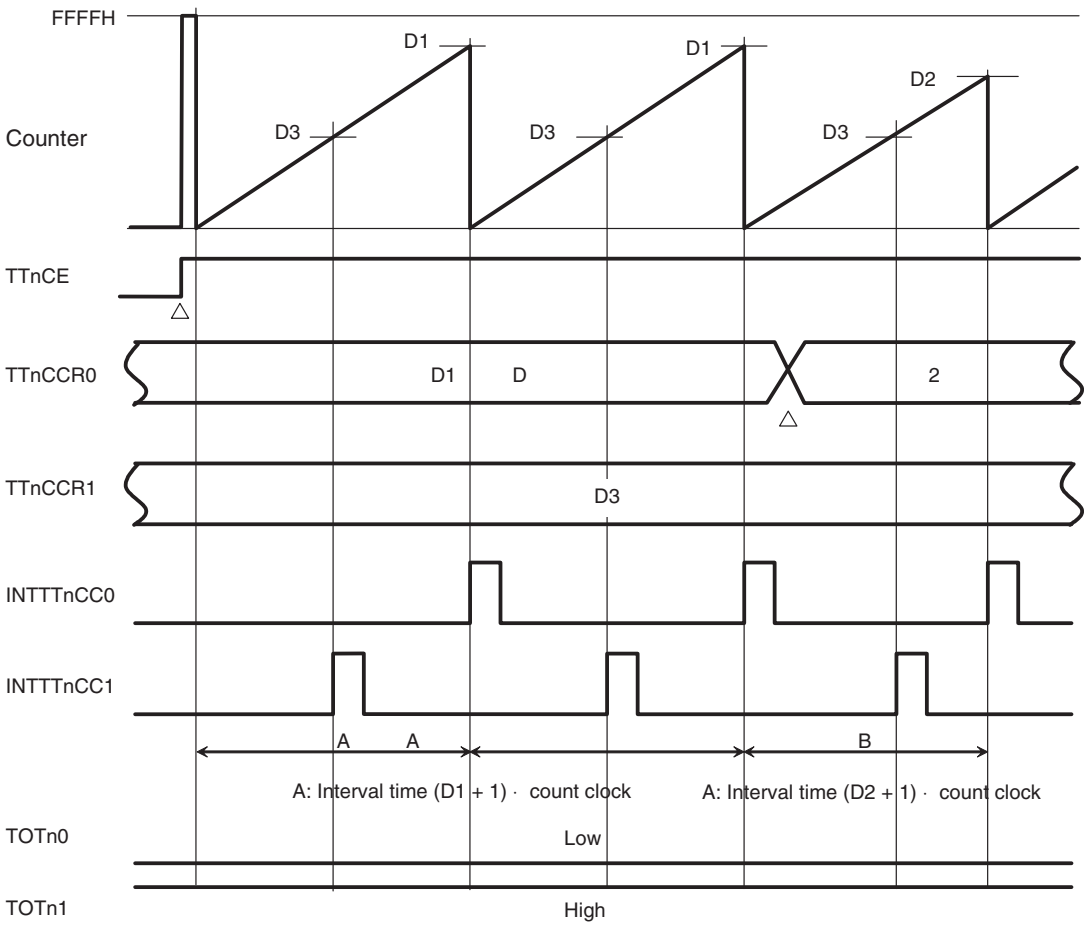
Figure 11-20: Basic Operation Flow in Interval Timer Mode



Note: In the case of a match between the counter and TTnCCR1 register, the counter is not cleared.

Figure 11-21: Basic Timing in Interval Timer Mode (1/2)

(a) When D1>D2>D3, only value of TTnCCR0 register is rewritten, TOTn0 and TOTn1 are not output (TTnOE0, 1 = 0, TTnOL0 = 0, TTnOL1 = 1)



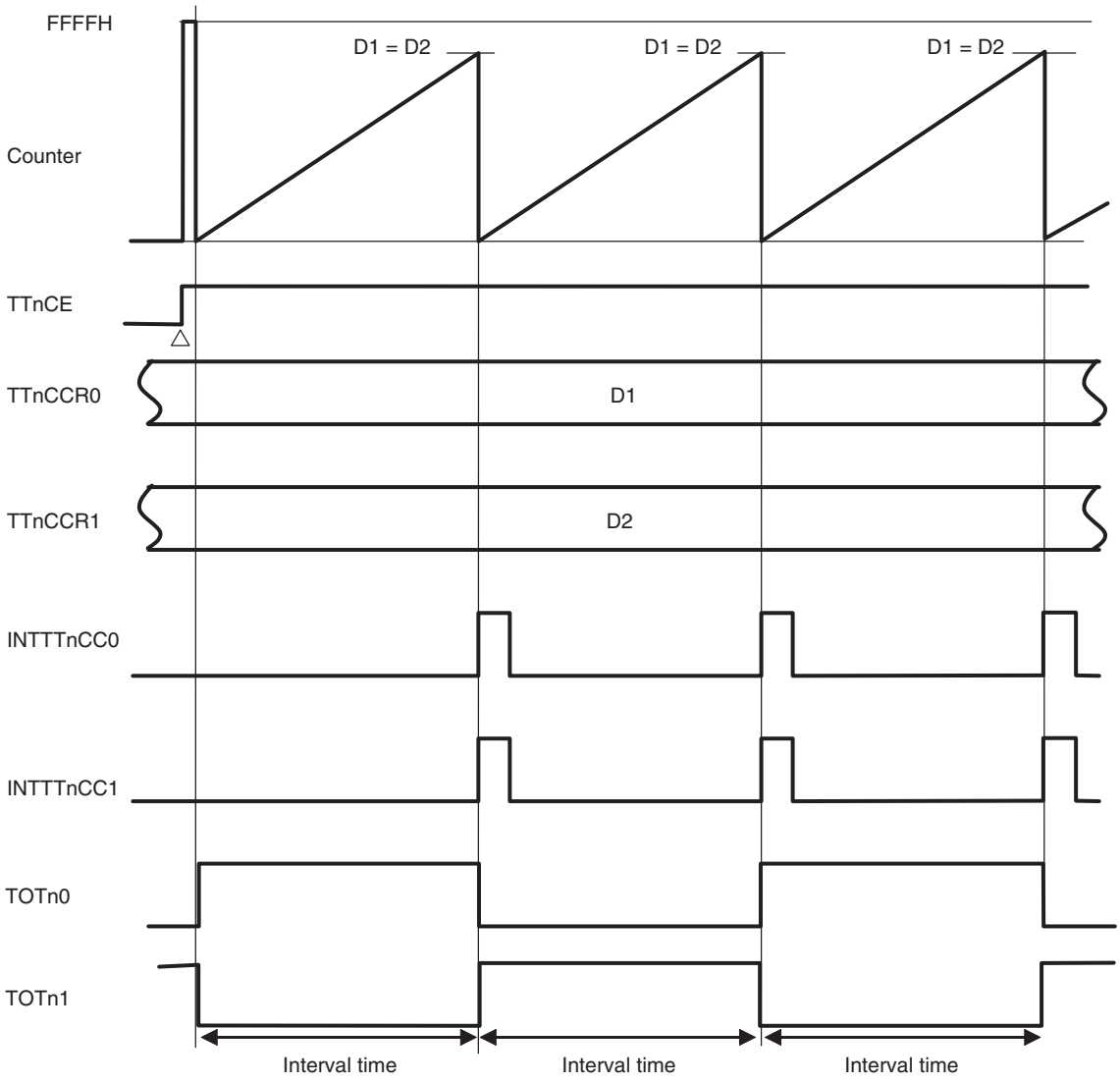
Remarks: 1. D1, D2: Setting values of TTnCCR0 register (0000H to FFFFH)
D3: Setting values of TTnCCR1 register (0000H to FFFFH)

2. Interval time = (Dm + 1) × (count clock cycle)

3. m = 1 to 3, n = 0, 1

Figure 11-21: Basic Timing in Interval Timer Mode (2/2)

(b) When D1 = D2, values of TTnCCR0 and TTnCCR1 registers not rewritten, TOTn1 output performed (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)



Remarks: 1. D1: Setting value of TTnCCR0 register (0000H to FFFFH)
D2: Setting value of TTnCCR1 register (0000H to FFFFH)

2. Interval time = (Dm + 1) × (count clock cycle)

3. TOTn0, TOTn1 toggle time = (Dm + 1) × (count clock cycle)

4. m = 1, 2, n = 0, 1

11.6.2 External event count mode

In the external event count mode, count up starts upon external event input (TEVTTn pin). (The external event input (TEVTTn) is used as the count clock, regardless of bit TTnEEE of the TTnCTL1 register.)
In the external event count mode, the counter is cleared only upon a match between the counter and the value of the TTnCCR0 register. Counter clearing using the TTnCCR1 register does not work.
However, the value of the TTnCCR1 register is transferred to the TTnCCR1 buffer register, compared to the counter value, and a compare match interrupt (INTTTnCC1) is output.
The TTnCCR0 and TTnCCR1 registers can be rewritten with the anytime write method, regardless of the value of bit TTnCE.
Pins TOTn0 and TOTn1 are toggle output controlled when bits TTnOE0 and TTnOE1 are set to 1.
When using only one compare register channel, it is recommended to set the TTnCCR1 register to FFFFH.

[External event count operation flow]

<1> TTnCTL1 register bits TTnMD3 to TTnMD0 = 0001B (mode setting)
Edge detection set with TTnIOC2 register bits TTnEES1 and TTnEES0 (TTnEES1, TTnEES0 = setting other than 01B)
<2> TTnCTL0 register bit TTnCE = 1 (count enable)
<3> TEVTTn pin input edge detection (count-up start)

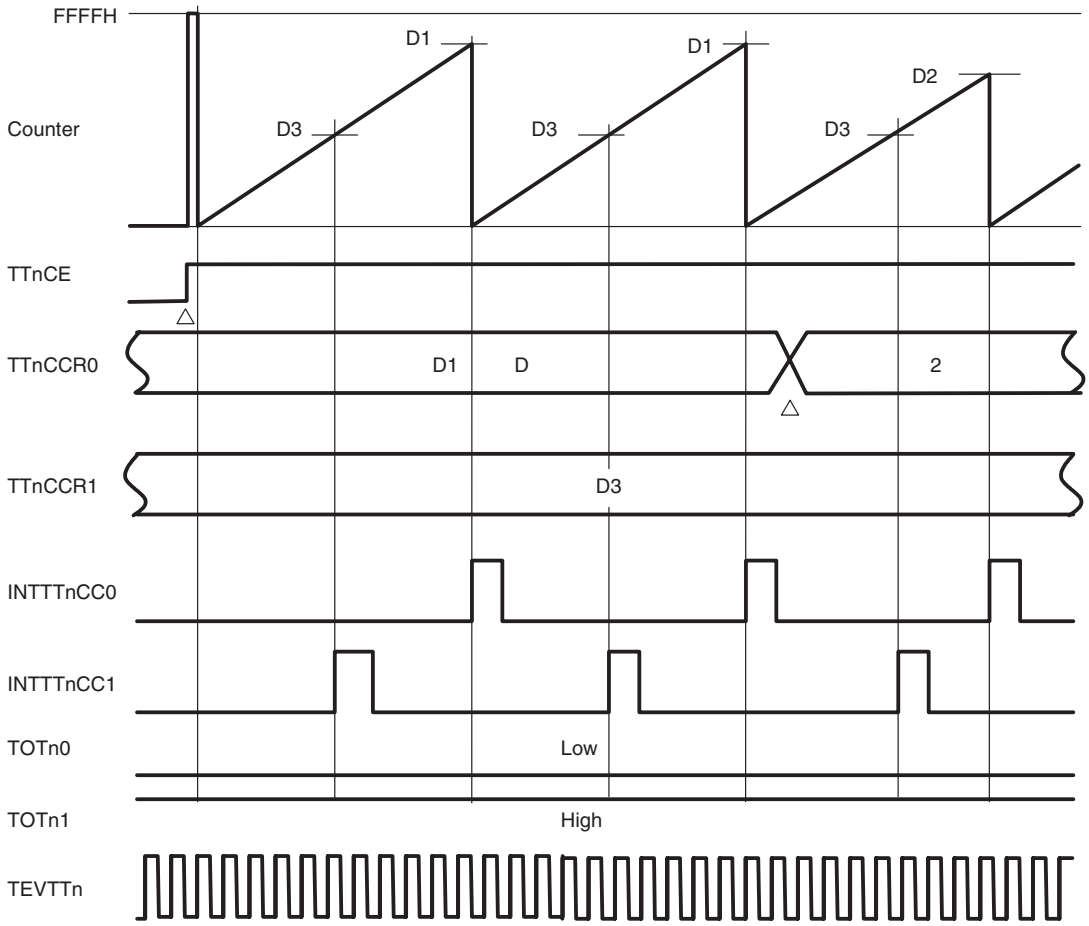
Cautions: 1. In external event count mode, when the setting value of the TTnCCR0 register is set to m, the number of TEVTTn pin input edge detection times is m+1.

2. In external event count mode, do not send the TTnCCR0 register to 0000H.

Remark: n = 0, 1

Figure 11-22: Basic Operation Timing in External Event Count Mode (1/4)

(a) When D1>D2>D3, only value of TTnCCR0 register is rewritten, TOTn0 and TOTn1 are not output. The signal input from TEVTTn and internally synchronized is counted as the count clock (TTnOE0, 1 = 0, TTnOL0 = 0, TTnOL1 = 1)



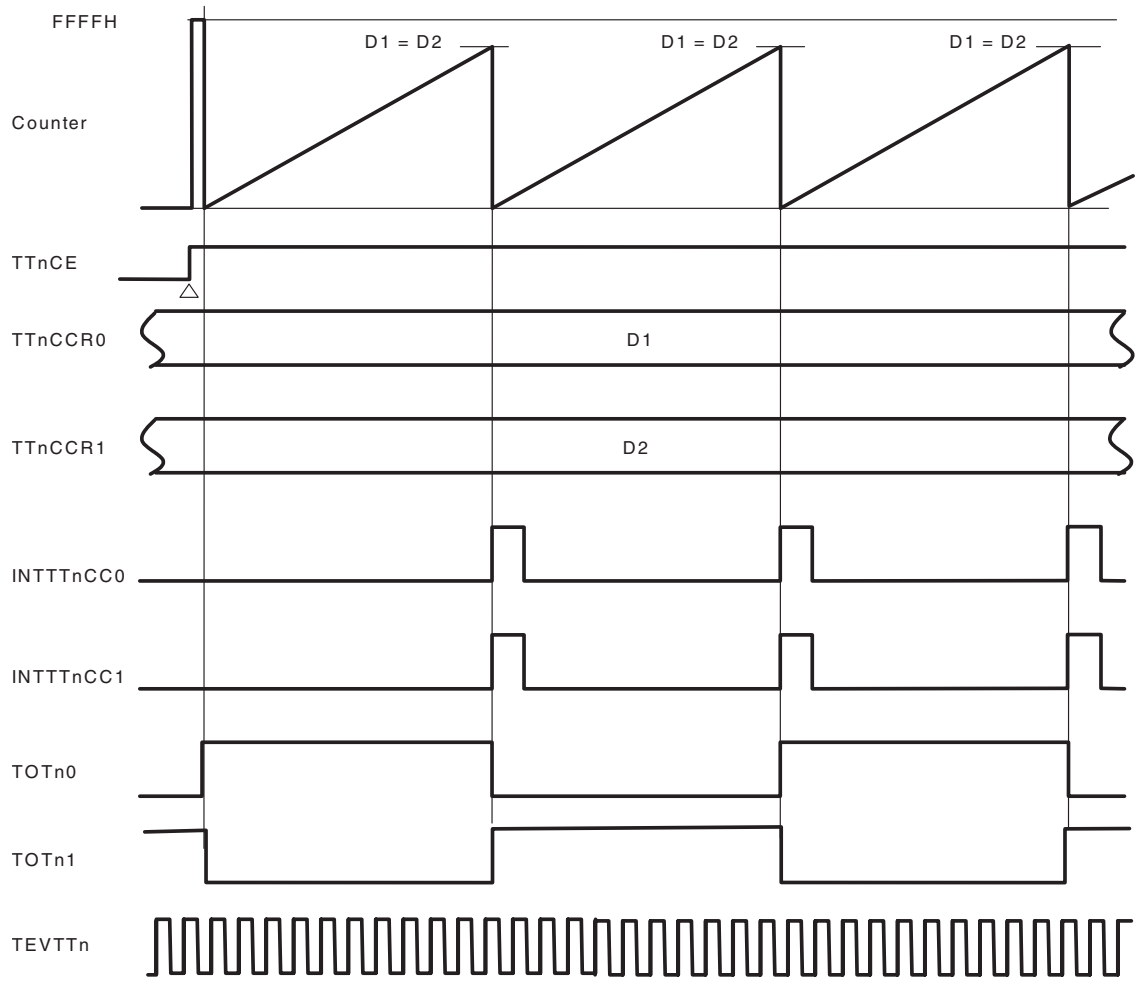
Remarks: 1. D1, D2: Setting values of TTnCCR0 register (0000H to FFFFH)
D3: Setting value of TTnCCR1 register (0000H to FFFFH)

2. Number of event counts = (Dm + 1) (m = 1, 2)

3. n = 0, 1

Figure 11-22: Operation Timing in External Event Count Mode (2/4)

(b) When D1 = D2, TTnCCR0 and TTnCCR1 register values are not rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)



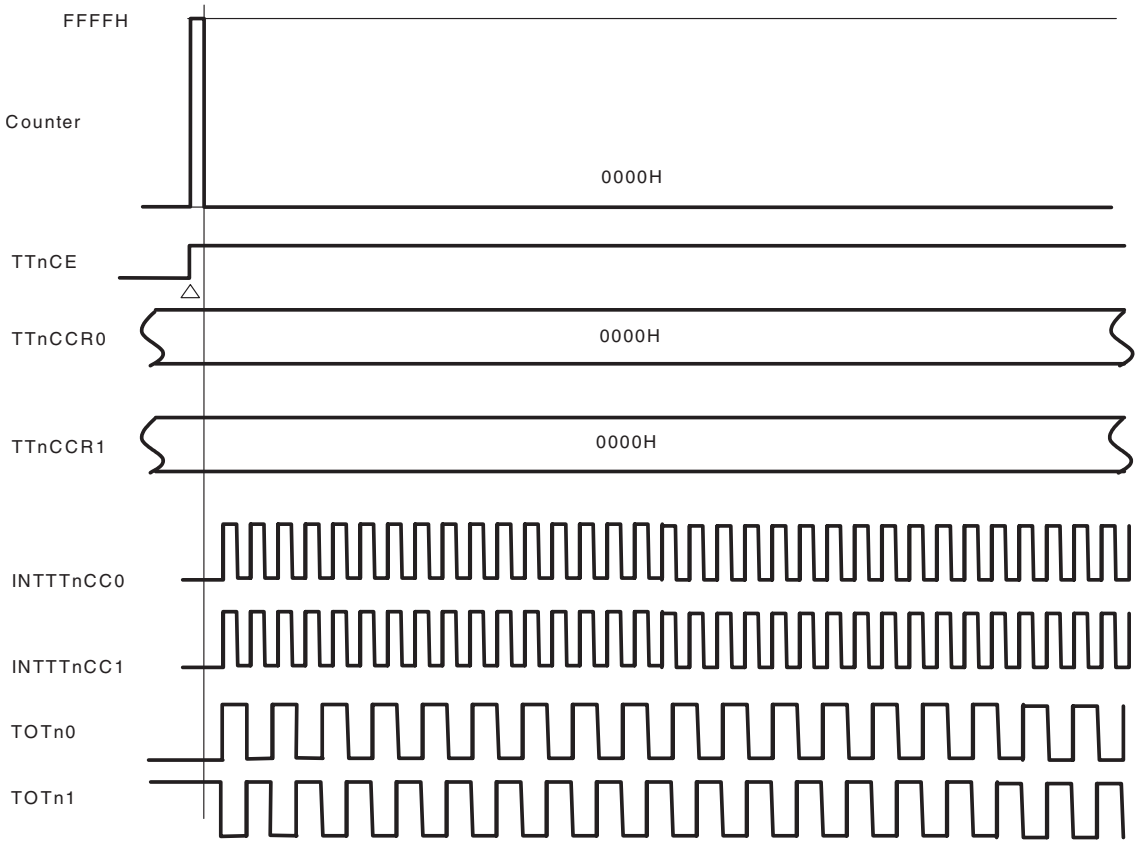
Remarks: 1. D1: Setting value of TTnCCR0 register (0000H to FFFFH)
D2: Setting value of TTnCCR1 register (0000H to FFFFH)

2. Number of event counts = (Dm + 1) (m = 1, 2)

3. n = 0, 1

Figure 11-22: Operation Timing in External Event Count Mode (3/4)

(c) When D1 = D2, TTnCCR0 and TTnCCR1 register values are not rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)



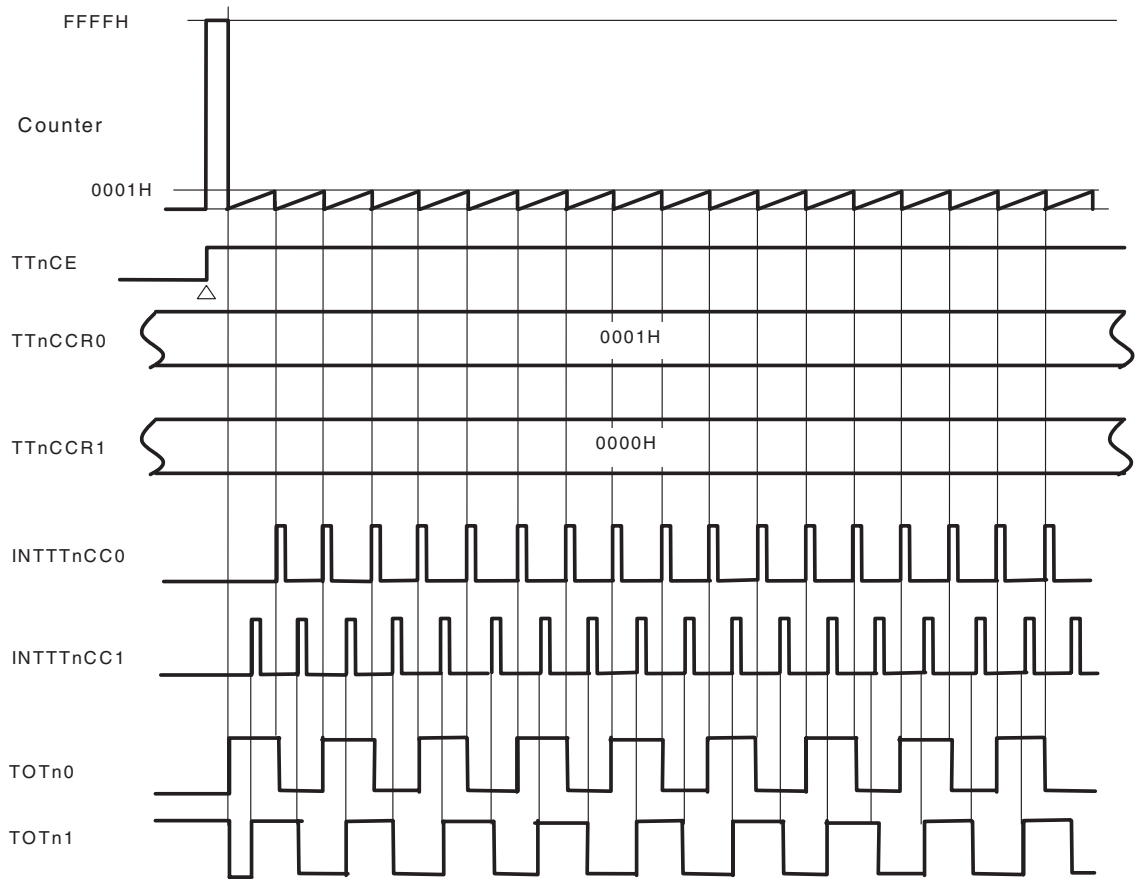
Remarks: 1. D1: Setting value of TTnCCR0 register (0000H)
D2: Setting value of TTnCCR1 register (0000H)

2. Number of event counts = (Dm + 1) (m = 1, 2)

3. n = 0, 1

Figure 11-22: Basic Operation Timing in External Event Count Mode (4/4)

(d) When D1 = D2, TTnCCR0, TTnCCR1 register values are not rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)



Remarks: 1. D1: Setting value of TTnCCR0 register (0001H)
D2: Setting value of TTnCCR1 register (0000H)

2. Number of event counts = (Dm + 1) (m = 1, 2)

3. n = 0, 1

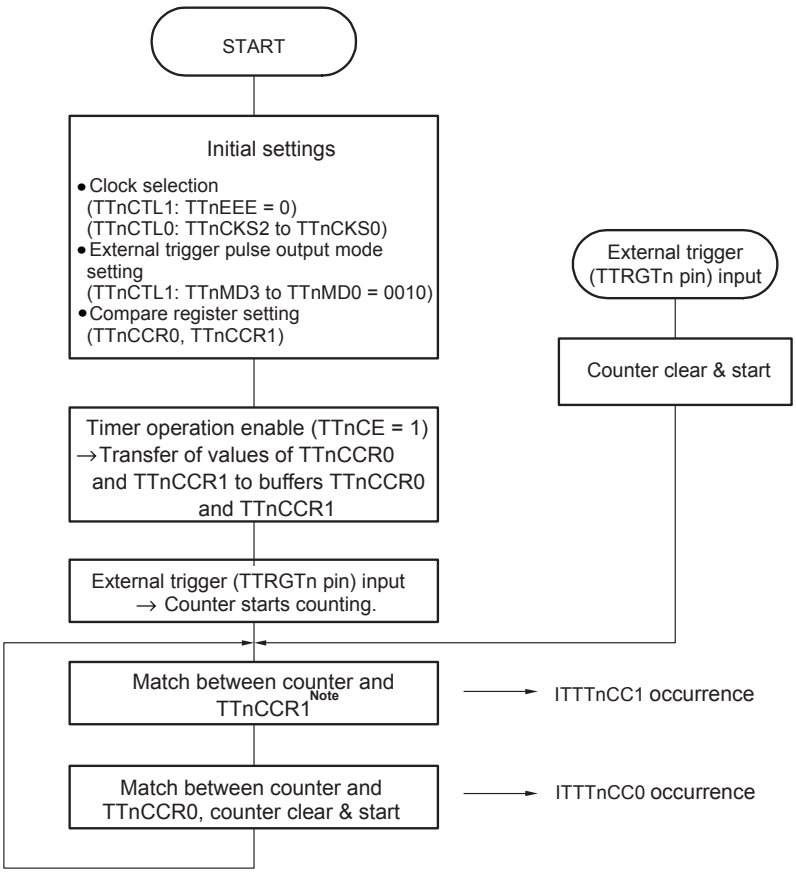
11.6.3 External trigger pulse output mode

When, in the external trigger pulse mode, the duty is set to the TTnCCR1 register, the cycle is set to the TTnCCR0 register, and TTnCE = 1 is set, external trigger input (TTRGTn pin) wait results, with the counter remaining stopped at FFFFH. Upon detection of the valid edge of external trigger input (TTRGTn pin), or when the TTnEST bit of the TTnCTL1 register is set, count up starts. An external trigger pulse is output from pin TOTn1, and toggle output is performed from pin TOTn0 upon a match with the TTnCCR0 register. Moreover, during the count operation, upon a match between the counter and the TTnCCR0 register, a compare match interrupt (INTTTnCC0) is output, and upon a match between the counter and TTnCCR1 register, a compare match interrupt (INTTTnCC1) is output.
The TTnCCR0 and TTnCCR1 registers can be rewritten during count operation. Compare register reload is performed at the timing when the counter value and the TTnCCR0 register match. However, when write access to the TTnCCR1 register is performed, the next reload timing becomes valid, so that even if wishing to rewrite only the value of the TTnCCR0 register, write the same value to the TTnCCR1 register. In this case, reload is not performed even if only the TTnCCR0 register is rewritten.
If, during operation in the external trigger pulse output mode, the external trigger (TTRGTn pin) edge is detected several times, or if the TTnEST bit of the TTnCTL1 register is set (to 1), the counter is cleared and count up is resumed. Moreover, if at this time, the TOTn1 pin is in the low level status, the TOTn1 pin output becomes high level when an external trigger is input. If the TOTn1 pin is in the high level status, it remains high level even if external trigger input occurs.
In the external trigger pulse output mode, the TTnCCR0 and TTnCCR1 registers have their function fixed as compare registers, so the capture function cannot be used.

Caution: In the external trigger pulse mode, set bit TTnEEE of the TTnCTL1 register to 0.

Remark: n = 0, 1

Figure 11-23: Basic Operation Flow in External Trigger Pulse Output Mode

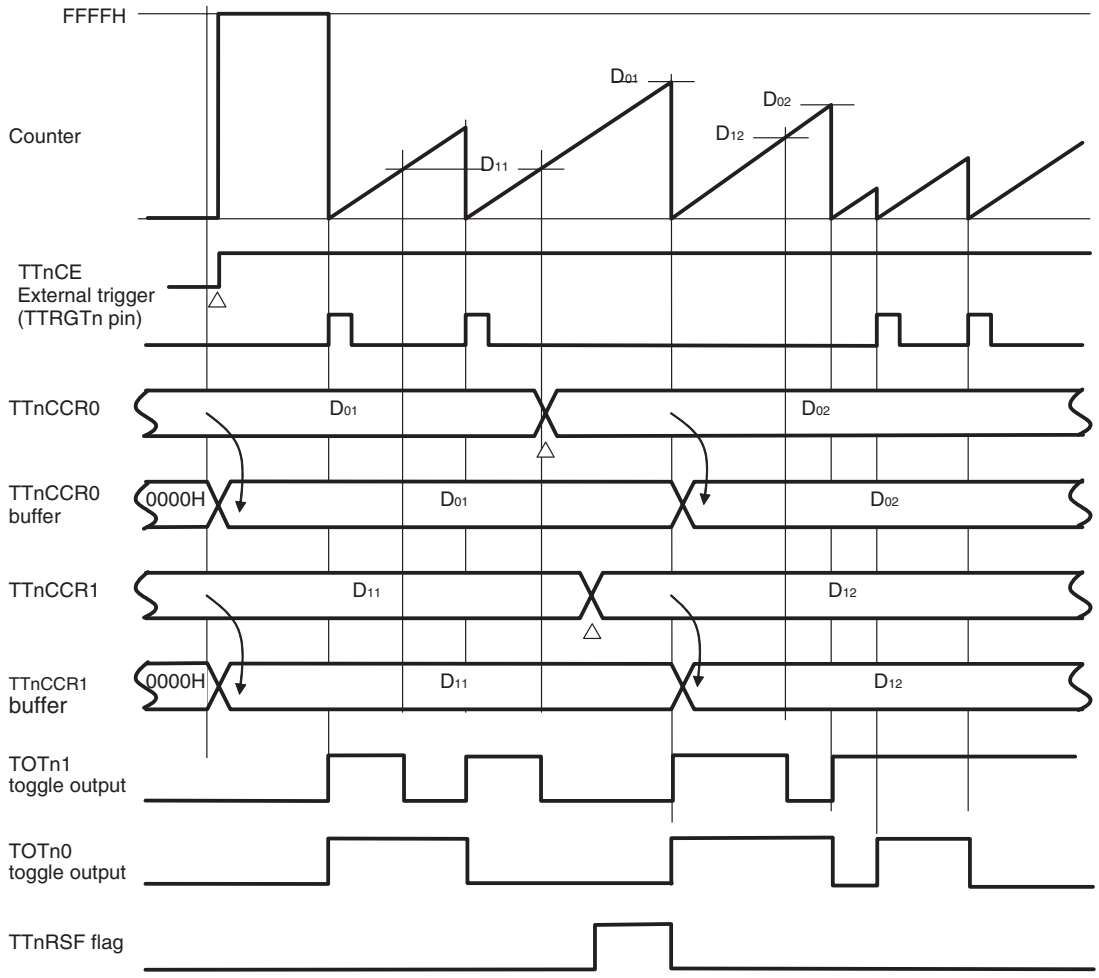


Note: The counter is not cleared upon a match between the counter and the TTnCCR1 buffer register.

Remark: n = 0, 1

Figure 11-24: Basic Operation Timing in External Trigger Pulse Output Mode

(a) When values of TTnCCR0 and TTnCCR1 registers are rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)



Remarks: 1. D_{01}, D_{02}: Setting values of TTnCCR0 register (0000H to FFFFH)
D_{11}, D_{12}: Setting values of TTnCCR1 register (0000H to FFFFH)

2. TOTn1 (PWM) duty = (setting value of TTnCCR1 register) × (count clock cycle)
TOTn1 (PWM) cycle = (setting value of TTnCCR0 register + 1) × (count clock cycle)

3. Pin TOTn0 is toggled when the counter is cleared immediately following count start.

4. n = 0, 1

11.6.4 One-shot pulse mode

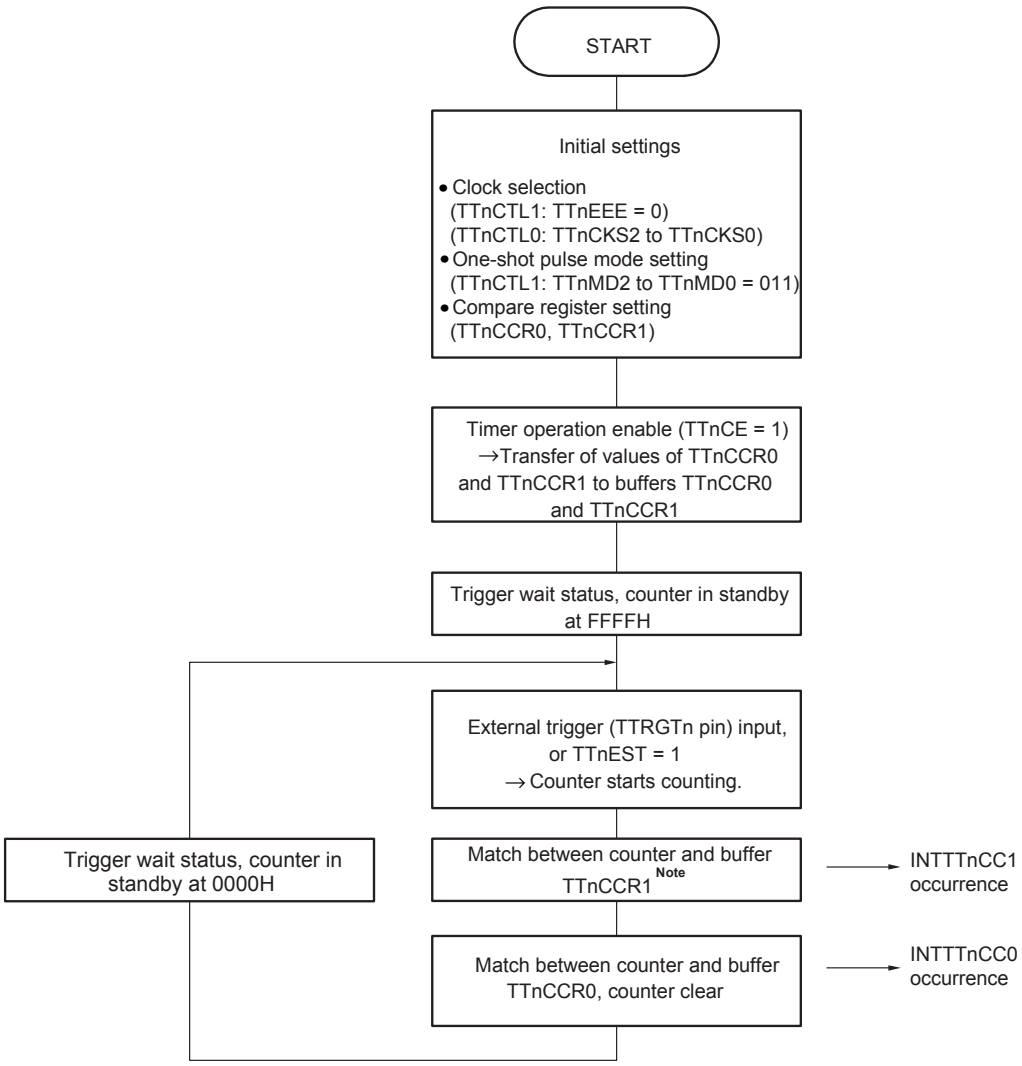
When, in the one-shot pulse mode, the duty is set to the TTnCCR0 register, the output duty delay value is set to the TTnCCR1 register, and bit TTnCE of the TTnCTL0 register is set to 1, external trigger input (TTRGTn pin) wait results, with the counter remaining stopped at FFFFH. Upon detection of the valid edge of external trigger input (TTRGTn pin), or when bit TTnEST of the TTnCTL0 register is set to 1, count up starts. The TOTn1 pin becomes high level upon a match between the counter and TTnCCR1 register. Moreover, upon a match between the counter and TTnCCR0 register, the TOTn1 pin becomes low level, and the counter is cleared to 0000H and then stops. The TOTn0 pin performs toggle output during the count operation upon a match between the counter and the TTnCCR0 buffer register.
Moreover, upon a match between the counter and TTnCCR0 register during count operation, a compare match interrupt (INTTTnCC0) is output, and upon a match between the counter and TTnCCR1 buffer register, a compare match interrupt (INTTTnCC1) is output.
The TTnCCR0 and TTnCCR1 registers can be rewritten using the anytime write method, regardless of the value of bit TTnCE.
Even if a trigger is input during the counter operation, it is ignored. Be sure to input the second trigger when the counter is stopped at 0000H.
In the one-shot pulse mode, registers TTnCCR0 and TTnCCR1 have their function fixed as compare registers, so the capture function cannot be used.

[One-shot pulse operation flow]

<1> TTnCTL1 register bits TTnMD3 to TTnMD0 = 0011B (One-shot pulse mode)
<2> TTnCCR0 register setting (duty setting), TTnIOC0 register bit TTnOE1 = 1 (TOTn1 pin output enable)
<3> TTnCTL0 register bit TTnCE = 1 (counter operation enable): TOTn1 = Low-level output
<4> TTnCTL1 register bit TTnEST = 1 or TTRGTn pin edge detection (count-up start): TOTn1 = Low-level output
<5> Match between counter value and TTnCCR1 buffer register: TOTn1 = High-level output
<6> Match between counter value and TTnCCR0 buffer register: TOTn1 = Low-level output, count clear
<7> Count stop: TOTn1 = Low-level output
<8> TTnCE = 0 (operation reset)
<1> to <2> can be in any order.

Caution: In the one-shot pulse mode, set bit TTnEEE of the TTnCTL1 register to 0.

Figure 11-25: Basic Operation Flow in One-Shot Pulse Mode



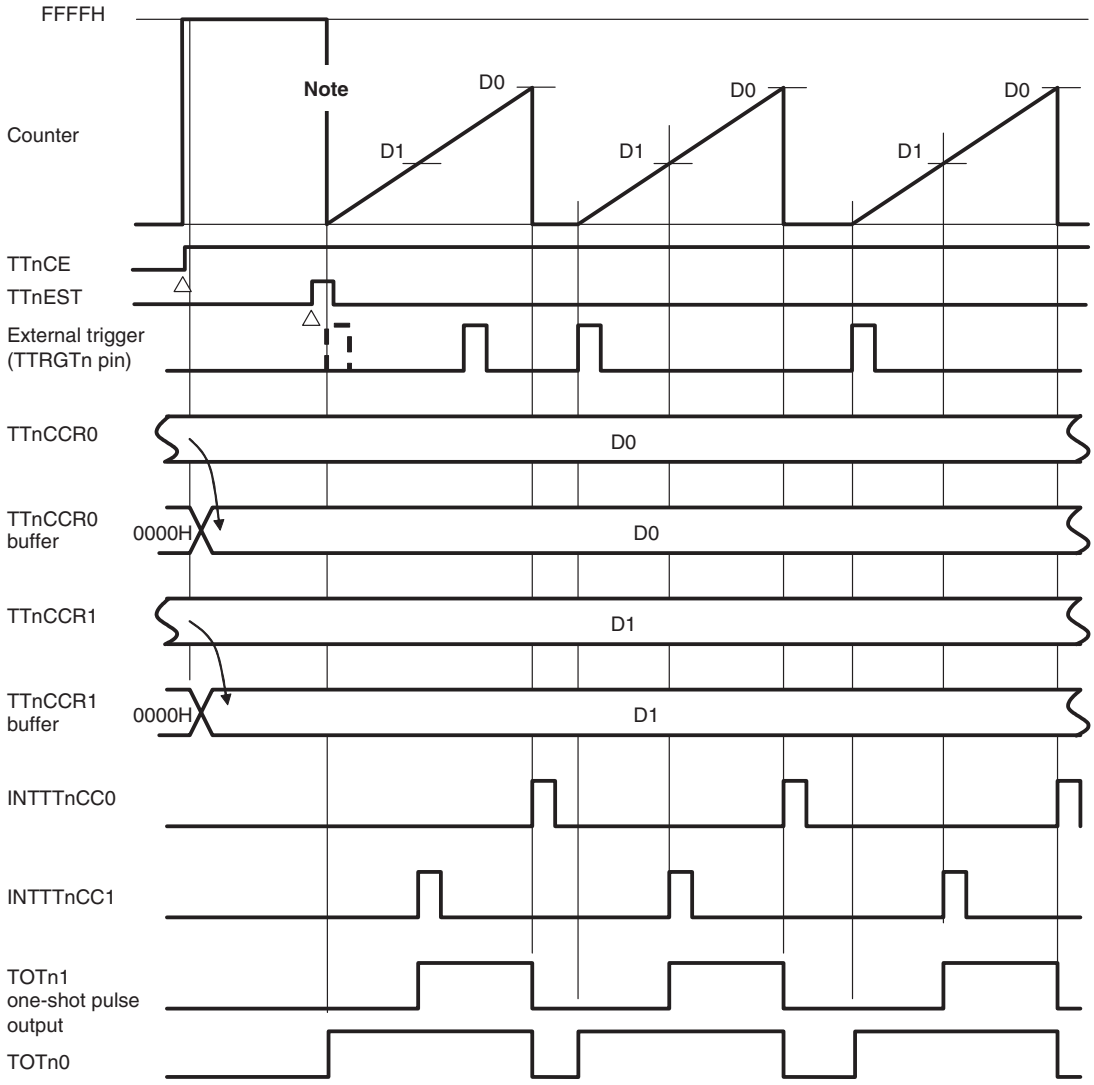
Note: The counter is not cleared upon a match between the counter and the TTnCCR1 buffer register.

Caution: The counter is not cleared even if trigger input is realized while the counter counts up, and the trigger input is ignored.

Remark: n = 0, 1

Figure 11-26: Basic Operation Timing in One-Shot Pulse Mode

(a) (TTnOE0, 1 = 1, TTnOL0, 1 = 0)



Note: Count up starts when the value of TTnEST becomes 1 or TTRGTn is input.

Remarks: 1. D0: Setting value of TTnCCR0 register (0000H to FFFFH)
D1: Setting value of TTnCCR1 register (0000H to FFFFH)

2. TOTn1 (output delay) = (setting value of TTnCCR1 register) × (count clock cycle)
TOTn1 (output pulse width) = {(setting value of TTnCCR0 register +1) -
(setting value of TTnCCR1 register)} × (count clock cycle)

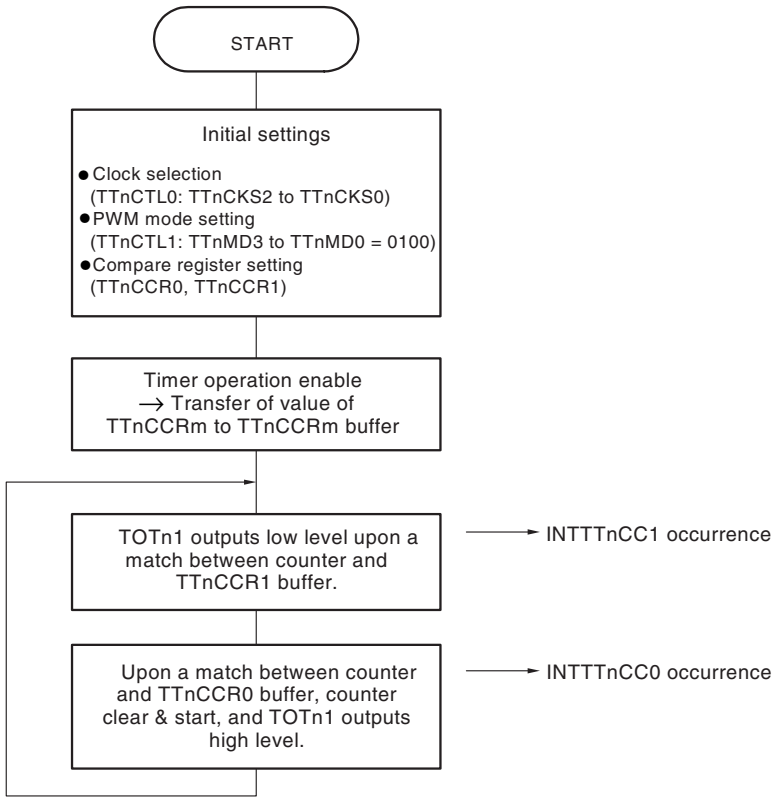
3. n = 0, 1

11.6.5 PWM mode

When, in the PWM mode, the duty is set to the TTnCCR1 register, the cycle is set to the TTnCCR0 register, and TTnCE = 1 is set, variable duty PWM output is performed from pin TOTn1.
Simultaneously with the start of count up operation, pin TOTn1 becomes high level, and upon a match between the counter and the TTnCCR1 register, becomes low level. Next, the TOTn1 pin becomes high level upon a match with the TTnCCR0 register. The TOTn0 pin performs toggle output upon a match with the TTnCCR0 buffer register.
During count operation, a compare match interrupt (INTTTnCC0) is output upon a match between the counter and TTnCCR0 register, and a compare match interrupt (INTTTnCC1) is output upon a match between the counter and TTnCCR1 register.
The TTnCCR0 and TTnCCR1 registers can be rewritten during count operation. Compare register reload occurs upon a match between the counter value and the TTnCCR0 buffer register. However, since the next reload timing becomes valid when the TTnCCR1 register is written to, write the same value to the TTnCCR1 register even when wishing to rewrite only the value of the TTnCCR0 register. Reloading is not performed if only the TTnCCR0 register is rewritten.
In the PWM mode, the TTnCCR0 and TTnCCR1 registers have their function fixed as compare registers, so the capture function cannot be used.

Figure 11-27: Basic Operation Mode in PWM Mode (1/2)

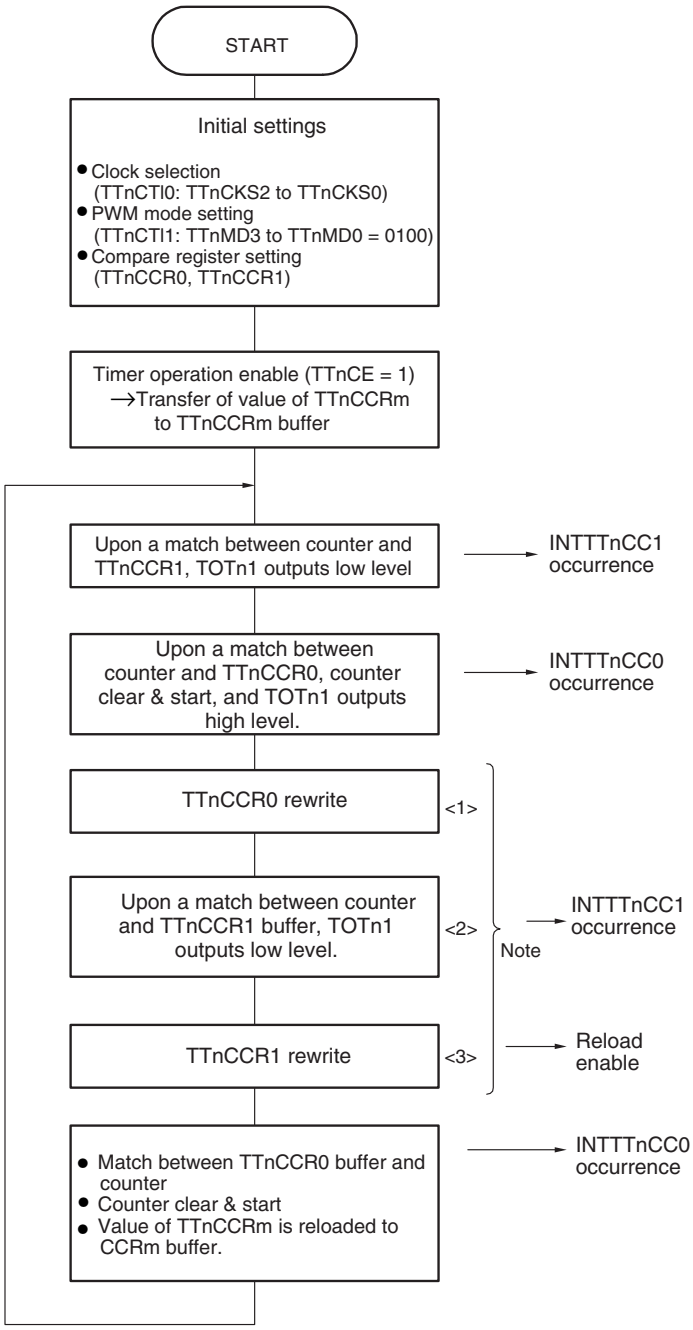
(a) When values of TTnCCR0 and TTnCCR1 registers are rewritten during timer operation



Remark: n = 0, 1
m = 0, 1

Figure 11-27: Basic Operation Flow in PWM Mode (2/2)

(b) When values of TTnCCR0 and TTnCCR1 registers are rewritten during timer operation

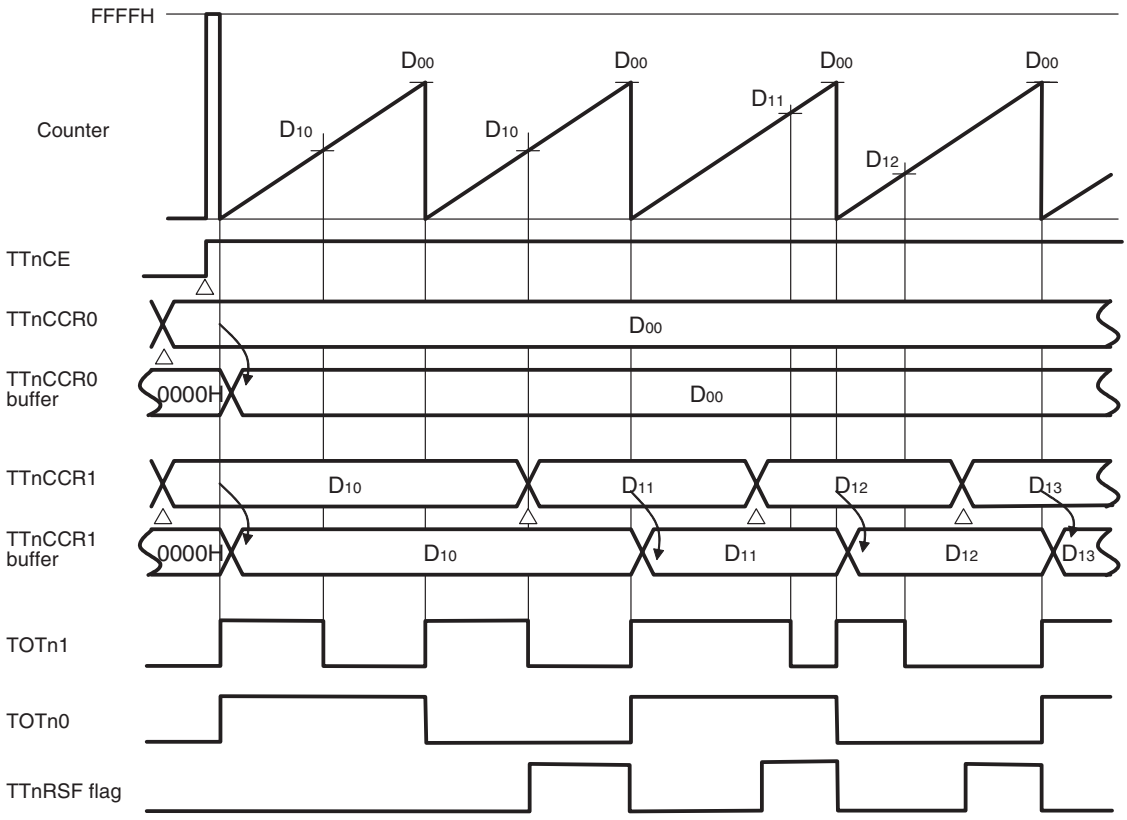


Note: Regarding the sequence, the timing of <2> may differ depending on the <1> or <3> rewrite timing, the value of the TTnCCR1 register, etc., but of <1> and <3>, always make <3> the last.

Remark: n = 0, 1
m = 0, 1

Figure 11-28: Basic Operation Timing in PWM Mode (1/2)

(a) When only value of TTnCCR1 is rewritten, and TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)



Remarks: **1.** D_{00}: Setting value of TTnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}, D_{13}: Setting values of TTnCCR1 register (0000H to FFFFH)

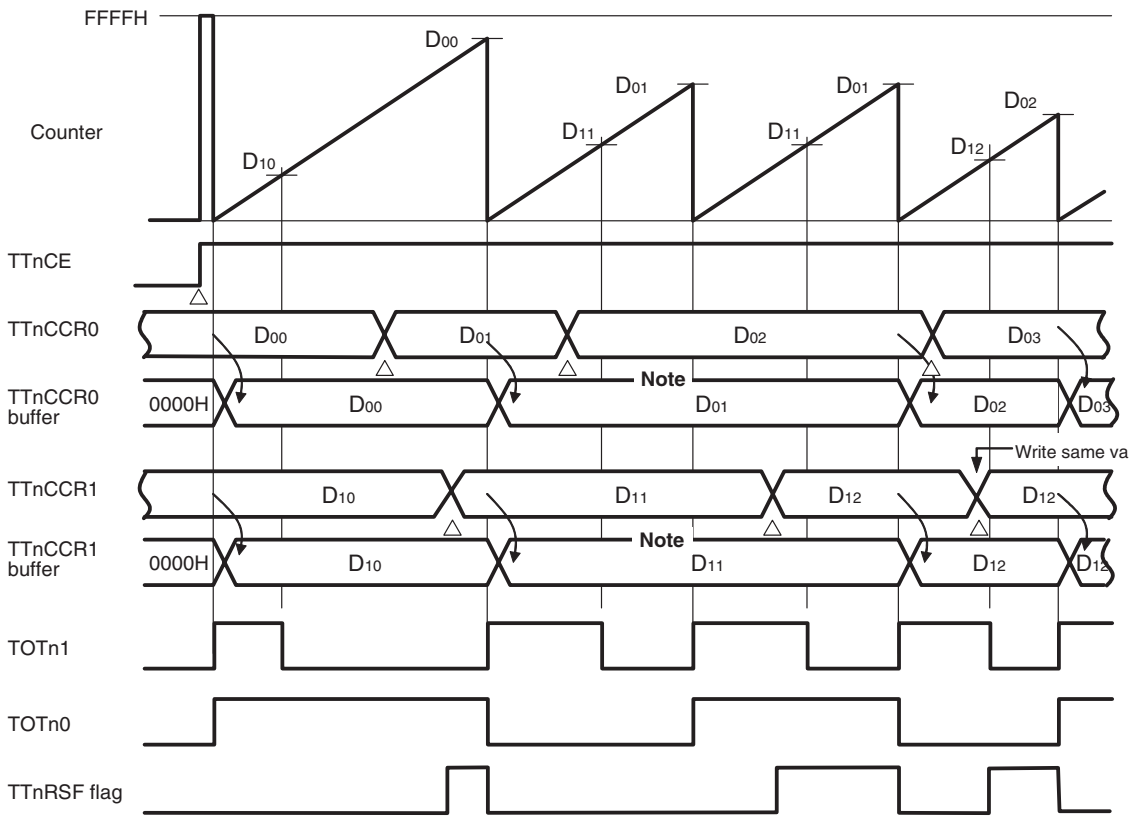
2. TOTn1 (PWM) duty = (setting value of TTnCCR1 register) × (count clock cycle)
TOTn1 (PWM) cycle = (setting value of TTnCCR0 register + 1) × (count clock cycle)

3. TOTn0 is toggled immediately following counter start and at (setting value of TTnCCR0 register + 1) × (count clock cycle)

4. n = 0, 1

Figure 11-28: Basic Operation Timing in PWM Mode (2/2)

(b) When values of TTnCCR0 and TTnCCR1 register are rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)



Note: The TTnCCR1 register was not written to, so transfer to the TTnCCR0 buffer register was not performed. Held until the next reload timing.

Remarks: 1. D_{00}, D_{01}, D_{02}, D_{03}: Setting values of TTnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}, D_{13}: Setting values of TTnCCR1 register (0000H to FFFFH)

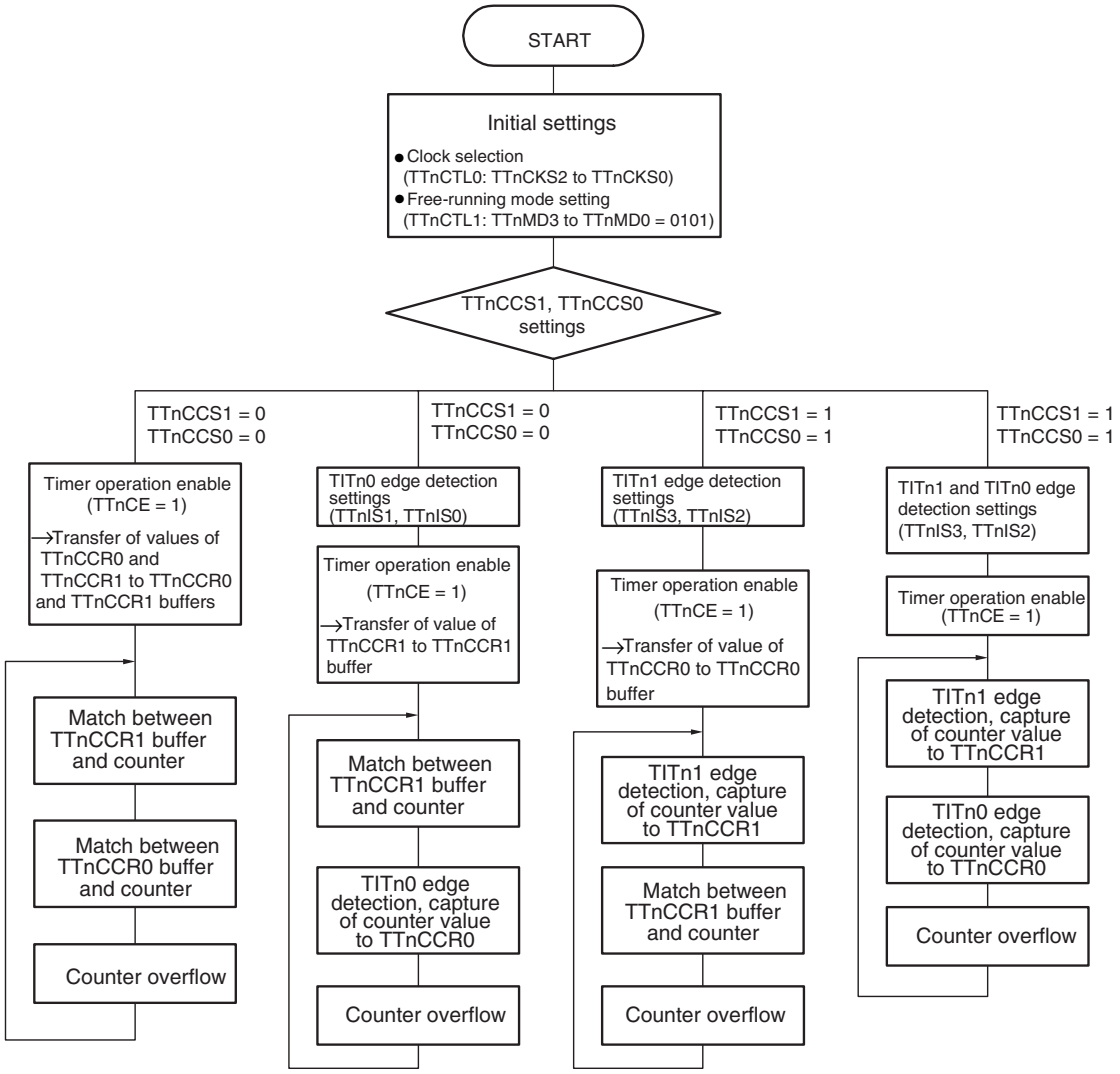
2. The TOTn0 and TOTn1 pins become high level at timer count start.

3. n = 0, 1

11.6.6 Free-running mode

The operation timing of the free-running mode is shown below.
The operation for bits TTnCCS1 and TTnCCS0 of register TTnOPT0 is specified.

Figure 11-29: Basic Operation Flow in Free-Running Mode



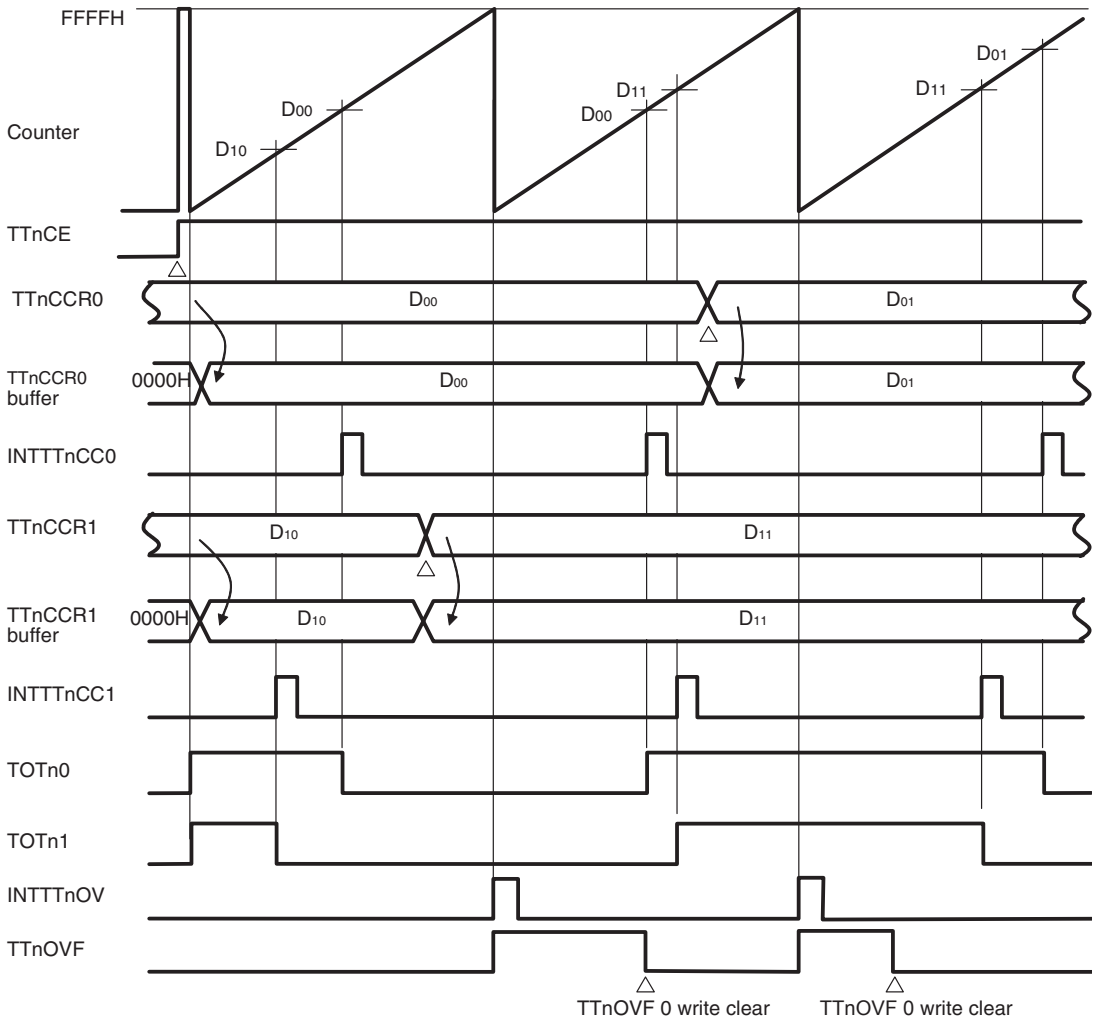
Remark: n = 0, 1

(1) Compare function (TTnCCS1 = 0, TTnCCS0 = 0)

When TTnCTL0 register bit TTnCE is set to 1, the counter counts from 0000H to FFFFH. An overflow interrupt (INTTTnOV) is output when the counter value changes from FFFFH to 0000H, and the counter is cleared. The count operation is performed in the free-running mode until TTnCE = 0 is set. Moreover, during count operation, a compare match interrupt (INTTTnCC0) is output upon a match between the counter and TTnCCR0 buffer register, and a compare match interrupt (INTTTnCC1) is output upon a match between the counter and TTnCCR1 buffer register.
The TTnCCR0 and TTnCCR1 registers can be rewritten using the anytime write method, regardless of the value of the TTnCE bit.
The TOTn0 and TOTn1 pins are toggle output controlled when bits register TTnOE0 and TTnOE1 of the TTnIOC0 register are set to 1.

Figure 11-30: Basic Operation Timing in Free-Running Mode (Compare Function)

(a) When values of TTnCCR0 and TTnCCR1 registers are rewritten, TOTn0, TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)



Remarks: 1. D_{00}, D_{01}: Setting values of TTnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}: Setting values of TTnCCR1 register (0000H to FFFFH)

2. TOTn0 (toggle) width = (setting value of TTnCCR0 register + 1) × (count clock cycle)

3. TOTn1 (toggle) width = (setting value of TTnCCR1 register + 1) × (count clock cycle)

4. Pins TOTn0 and TOTn1 become high level at count start.

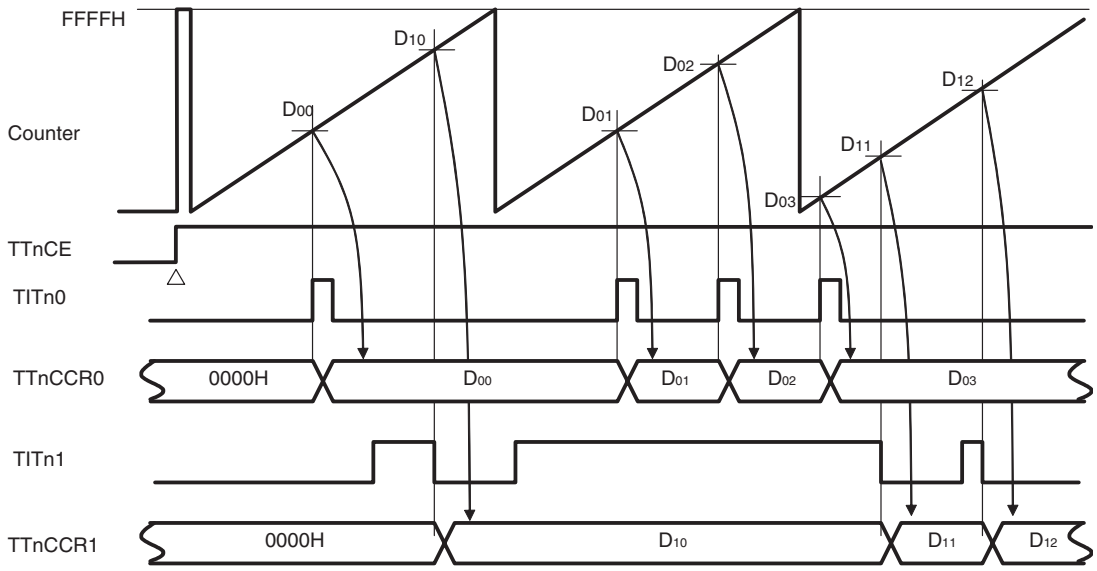
5. n = 0, 1

(2) Capture function (TTnCCS1 = 1, TTnCCS0 = 1)

When TTnCTL0 register bit TTnCE is set to 1, the counter counts from 0000H to FFFFH. An overflow interrupt (INTTTnOV) is output when the value of the counter changes from FFFFH to 0000H, and the counter is cleared. The count operation is performed in the free-running mode until TTnCE = 0 is set. When, during count operation, the counter value is captured to the TTnCCR0 and TTnCCR1 registers through detection of the valid edge of capture input (TITn1, TITn0), a capture interrupt (INTTTnCC0, INTTTnCC1) is output.
Regarding capture in the vicinity of overflow (FFFFH), judgment is possible with the overflow flag (TTnOVF). However, judgment with the TTnOVF flag is not possible when the capture trigger interval is such that it includes two overflow occurrences (2 or more free-running cycles).

Figure 11-31: Basic Operation Timing in Free-Running Mode (Capture Function)

(a) When TOTn0, TOTn1 are not output (TTnOE0, 1 = 0, TTnOL0, 1 = 0)



Remarks: 1. D_{00}, D_{01}: Values captured to TTnCCR0 register (0000H to FFFFH)
D_{10}, D_{11}: Values captured to TTnCCR1 register (0000H to FFFFH)

2. TITn0: Setting to rising edge detection (TTnIOC1 register bits TTnIS1, TTnIS0 = 01)
TITn1: Setting to falling edge detection (TTnIOC1 register bits TTnIS3, TTnIS2 = 10)

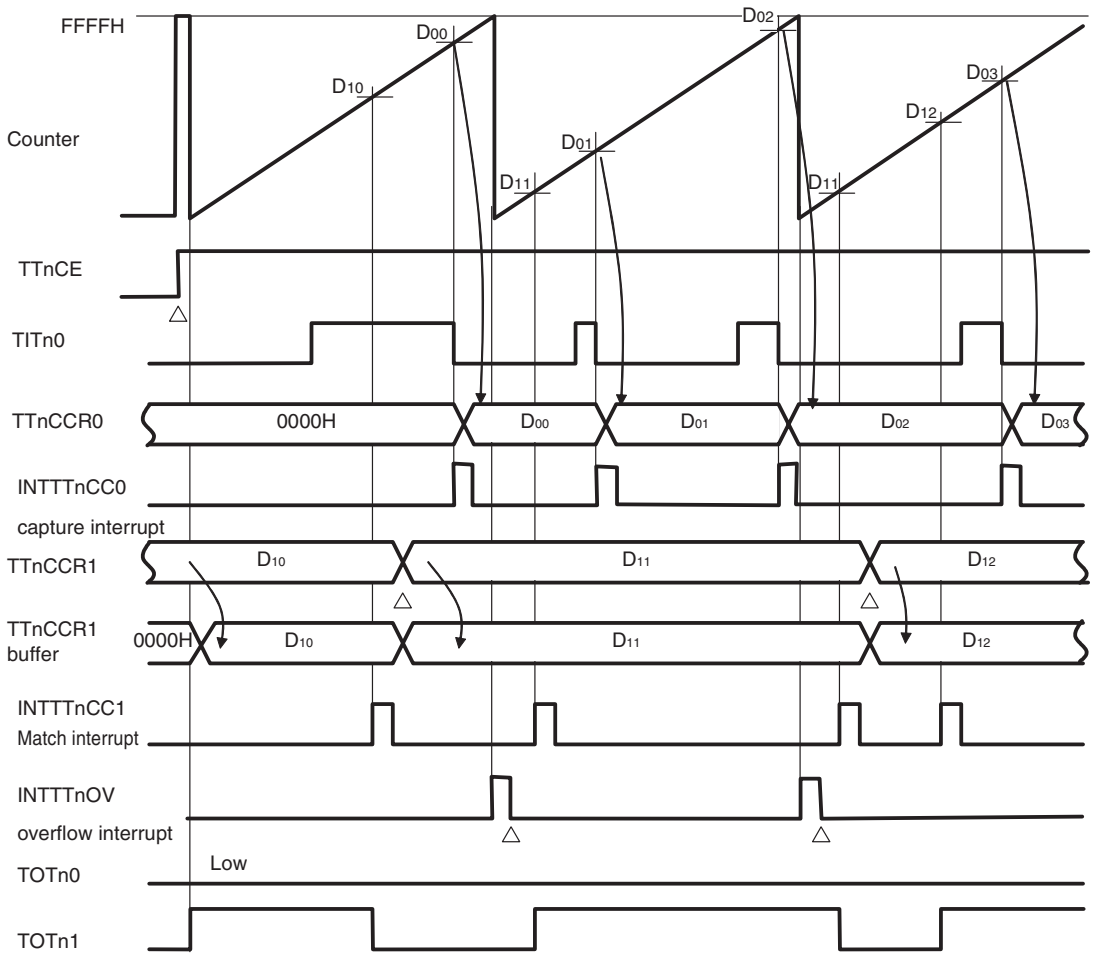
3. n = 0, 1

(3) Compare/capture function (TTnCCS1 = 0, TTnCCS0 = 1)

When TTnCTL0 register bit TTnCE is set to 1, the counter counts from 0000H to FFFFH, an overflow interrupt (INTTTnOV) is output when the value of the counter changes from FFFFH to 0000H, and the counter is cleared. The count operation is performed in the free-running mode until TTnCE = 0 is set. The TTnCCR1 register is used as a compare register, and as the interval function upon a match between the counter and TTnCCR1 register, a compare match interrupt (INTTTnCC1) is output. Since the TTnCCR0 register is set to the capture function, the TOTn0 pin cannot be controlled even when TTnIOC0 register bit TTnOE0 is set to 1.

Figure 11-32: Basic Operation Timing in Free-Running Mode (Compare/Capture Function)

(a) When value of TTnCCR1 is rewritten, TOTn0, TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)



Remarks: 1. D_{00}, D_{01}: Setting values of TTnCCR1 register (0000H to FFFFH)
D_{10}, D_{11}, D_{12}, D_{13}, D_{14}, D_{15}: Values captured to TTnCCR0 register (0000H to FFFFH)

2. TITn0: Setting to rising edge detection (TTnIOC1 register bits TtnIS1, TtnIS0 = 11)

3. n = 0, 1

(4) Overflow flag

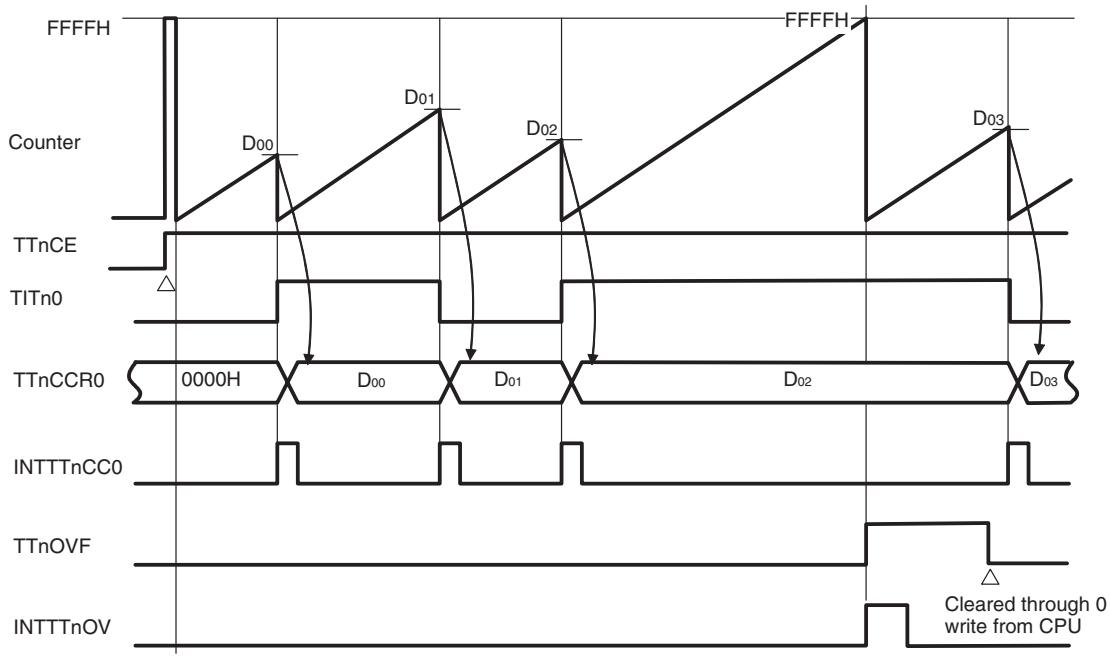
When, in the free-running mode, the counter overflows from FFFFH to 0000H, the overflow flag (TTnOVF) is set to "1", and an overflow interrupt (INTTTnOV) is output.
The overflow flag is cleared through 0 write from the CPU.
(The overflow flag is not cleared by just being read.)

11.6.7 Pulse width measurement mode

In the pulse width measurement mode, counting is performed in the free-running mode. The counter value is saved to the TTnCCR0 register, and the counter is cleared to 0000H. As a result, the external input pulse width can be measured. However, when measuring a long pulse width that exceeds counter overflow, perform judgment with the overflow flag. Measurement of pulses during which overflow occurs twice or more is not possible, so adjust the counter's operating frequency. Even in the case of TITn1 pin edge detection, pulse width measurement can be similarly performed by using the TTnCCR1 register.

Figure 11-33: Basic Operation Timing in Pulse Width Measurement Mode

(a) (TTnOE0, 1 = 0, TTnOL0, 1 = 0)



Remarks: 1. D_{00}, D_{01}, D_{02}, D_{03}: Values captured to TTnCCR0 register (0000H to FFFFH)

2. TITn0: Setting to rising edge/falling edge (both edges) detection (TTnIOC1 register bits TtnIS1, TtnIS0 = 11)

3. n = 0, 1

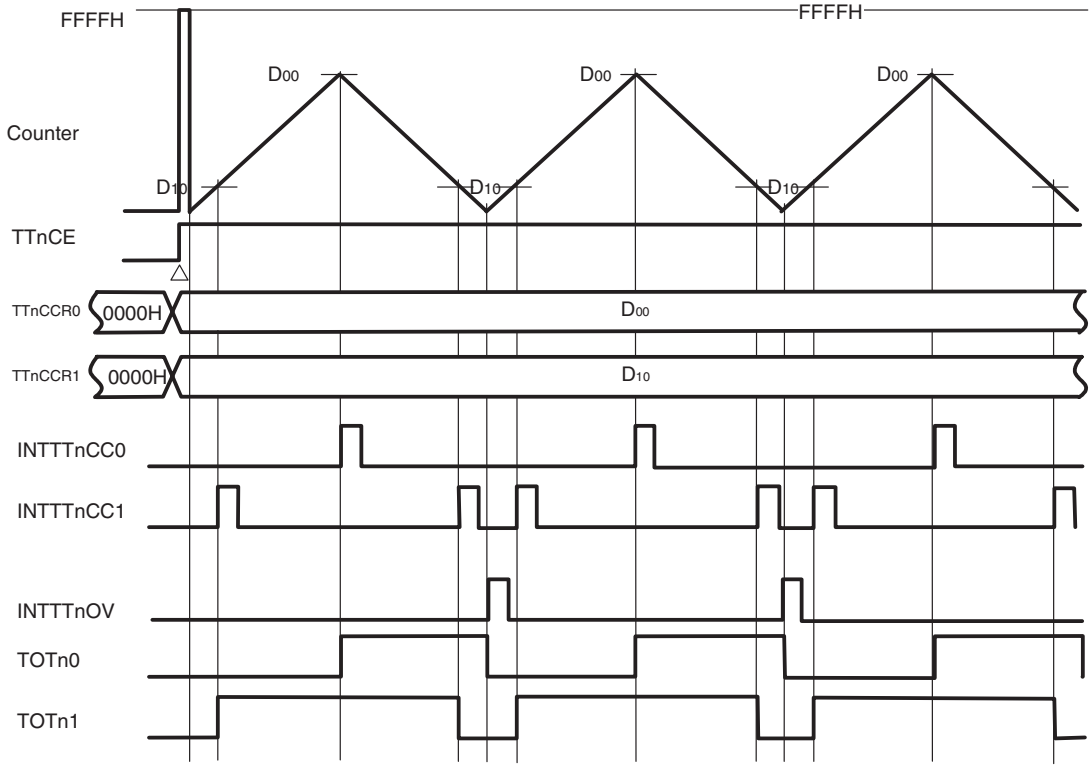
11.6.8 Triangular wave PWM mode

In the triangular wave PWM mode, similarly to in the PWM mode, when the duty is set to the TTnCCR1 register, the cycle is set to the TTnCCR0 register, and TTnCE = 1 is set, variable duty and cycle type triangular wave PWM output is performed from pin TOTn1. The TOTn0 pin is toggle output upon a match with the TTnCCR0 buffer register and upon counter underflow. Upon a match between the counter and TTnCCR0 register during count operation, a compare match interrupt (INTTTnCC0) is output, and upon a match between the counter and TTnCCR1 register, a compare match interrupt (INTTTnCC1) is output. Moreover, upon counter underflow, an overflow interrupt (INTTTnOV) is output.
The TTnCCR0 and TTnCCR1 registers can be rewritten during count operation. Compare register reload occurs upon a match between the counter value and the TTnCCR0 buffer register. However, since the next reload timing becomes valid when the TTnCCR1 register is written to, write the same value to the TTnCCR1 register even when wishing to rewrite only the value of the TTnCCR0 register. Reloading is not performed if only the TTnCCR0 register is rewritten. The reload timing is the underflow timing.
In the triangular wave PWM mode, the TTnCCR0 and TTnCCR1 registers have their function fixed as compare registers, so the capture function cannot be used.

Remark: In the triangular wave PWM mode, set the TTnCCR0 register to a value of 0 ≤TTnCCR0 ≤ FFFEH.

Figure 11-34: Basic Operation Timing in Triangular Wave PWM Mode

(a) When TOTn0, TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)



Remark: n = 0, 1

11.6.9 Encoder count function

The encoder compare mode is provided as follows.

Mode	TTnCCR0 register	TTnCCR1 register
Encoder compare mode	Compare only	Compare only

(1) Counter up/down control

Counter up/down control is performed and the counter is operated according to the phase of signals TENCTn0 and TENCTn1 from the encoder and the set conditions of bits TTnUDS1 and TTnUDS0 of the TTnCTL2 register.

(2) Basic operation

To use the TTnCCR0 and TTnCCR1 registers are compare-only registers, enable rewrite during timer operation.
The rewrite method is anytime write.
A compare match interrupt (INTTTnCC0) is output upon a match between the counter and TTnCCR0 register. A compare match interrupt (INTTTnCC1) is output upon a match between the counter and TTnCCR1 register.

(3) Counter clear operation

Clearing of the counter to 0000H is performed under the following conditions.

Clear Condition	
Method whereby counter is cleared to 0000H upon match with compare register (setting of TTnCTL2 register bits TTnECM1, TTnECM0)	√
Method whereby counter is cleared to 0000H upon detection of edge of pin TECRT0 (setting of bits TTnECS1, TTnECS0 when TTnIOC3 register bit TTnSCE = 0)	√
Method whereby counter is cleared to 0000H by special clear function of encoder (setting of bits TTnZCL, TTnBCL, TTnACL when TTnIOC3 register bit TTnSCE = 1)	√

(4) Control through TTnCTL2 register

The settings of the TTnCTL2 register in the encoder compare mode (TTnMD3 to TTnMD0 = 1000B) are as follows.

TTnMD3 to 0	TTnUDS1 to 0	TTnECM1	TTnECM0	TTnLDE	Clear	Load
1000B	All settings possible 00B 01B 10B 11B	0	0	0	-	-
				1		√
			1	0	TTnCCR0	-
				1		•
		1	0	Invalid	TTnCCR1	-
			1	Invalid	TTnCCR0 TTnCCR1	-

- In the case of bits TTnUDS1 and TTnUDS0, up/down judgment control is performed for the phase input from pins TENCTn0 and TENCTn1.
- In the case of bits TTnECM1 and TTnECM0, counter clear control is performed upon a match between the counter value and the compare setting value.

Bits TTnECM1 and TTnECM0 are valid in modes where the TTnCCR0 or TTnCCR1 register is used as a compare-only register.
These bits are invalid in modes where the TTnCCR0 or TTnCCR1 register is used as a capture-only register.

- The TTnLDE bit controls the function to load to the counter the setting value of the TTnCCR0 register upon occurrence of counter underflow.

Bit TTnLDE is valid only when the TTnECm bit setting is 00B, 01B, in a mode where the TTnCCR0 or TTnCCR1 register is used as a compare-only register.
In the case of all other settings, bit TTnLDE is invalid even if manipulated.
As an example of the use of the encoder count function, counter operation becomes possible between the setting values of registers 0000H to TTnCCR0 by using the counter load functions (TTnLDE = 1) indicated with "•" in the table, and the function for clearing the counter to 0000H in case the count operation following a match with the TTnCCR0 buffer register is up count (TTnECM0 = 1). (Refer to **11.6.9 (4) (c) Counter load function for TTnCCR0 register setting value upon underflow (bit TTnLDE of register TTnCTL2))**).

(a) Up/down count selection specification (TTnCTL2 register bits TTnUDS1, TTnUDS0)

Counter up/down is judged according to the settings of bits TTnUDS1 and TTnUDS0, and the phases input from pins TENCTn0 and TENCTn1.
Bits TTnUDS1 and TTnUDS0 are valid only in the encoder compare mode.

<1> TTnCTL2: TTnUDS1, 0 = 00B (count judgment mode 1)

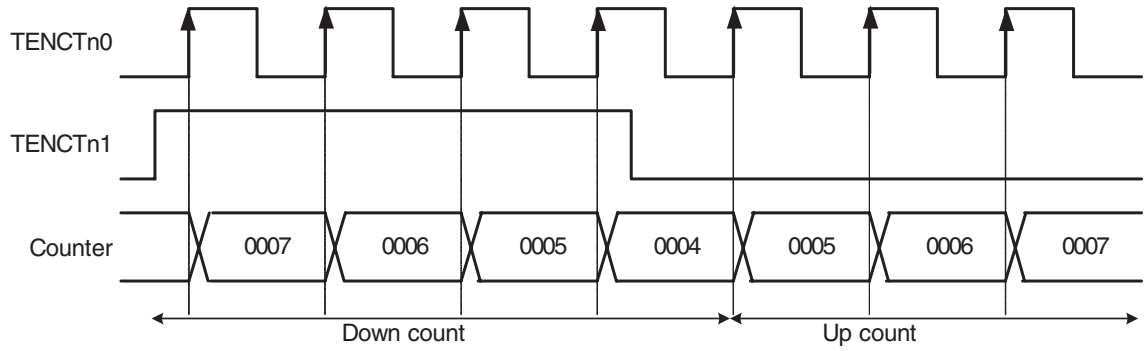
A Phase (Pin TENCTn0)	B Phase (Pin TENCTn1)	Count
Rising edge	High level	Down
Falling edge		
Both edges		
Rising edge	Low level	Up
Falling edge		
Both edges		

Operation example:

TTnIOC3: TTnEIS3 to 2	TENCTn1 pin input	Edge detection specification invalid
TTnIOC3: TTnEIS1 to 0 = 10B	TENCTn0 pin input	Rising edge detection

Figure 11-35: Encoder Count Function Up/Down Count Selection Specification Timings (1/6)

(a) Timing 1



Remarks: 1. Counting is performed when the edges of the TENCTn0/TENCTn1 pin inputs overlap.

2. n = 0, 1

<2> TTnCTL2: TTnUDS1, 0 = 01B (count judgment mode 2)

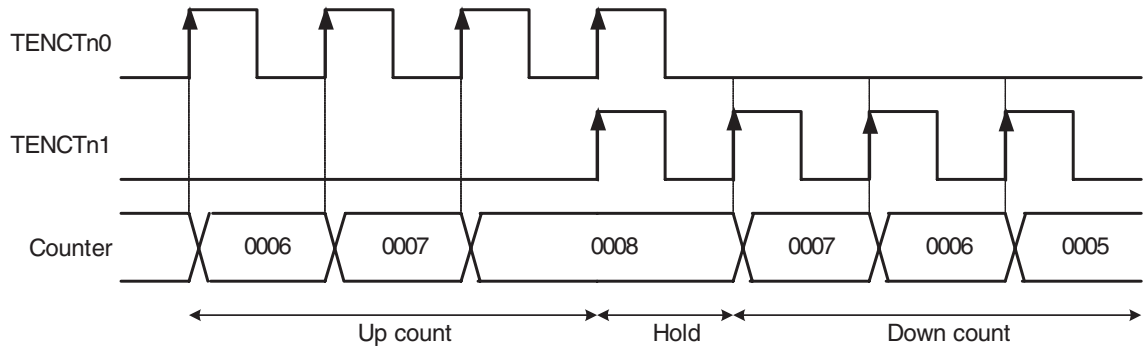
A Phase (Pin TENCTn0)	B Phase (Pin TENCTn1)	Count
Low level	Rising edge	Down
	Falling edge	
	Both edges	
High level	Rising edge	
	Falling edge	
	Both edges	
Rising edge	Low level	Up
Falling edge		
Both edges		
Rising edge	High level	
Falling edge		
Both edges		
Simultaneous pin TENCTn0/TENCTn1 inputs		Hold

Operation example:

TTnIOC3: TTnEIS3, 2 = 10B	TENCTn1 pin input	Rising edge detection
TTnIOC3: TTnEIS1, 0 = 10B	TENCTn0 pin input	Rising edge detection

Figure 11-35: Encoder Count Function Up/Down Count Selection Specification Timings (2/6)

(b) Timing 2



Remarks: 1. The count value is held when the edges of the TENCTn0/TENCTn1 pin inputs overlap.

2. n = 0, 1

<3> TTnCTL2: TTnUDS1, 0 = 10B (count judgment mode 3)

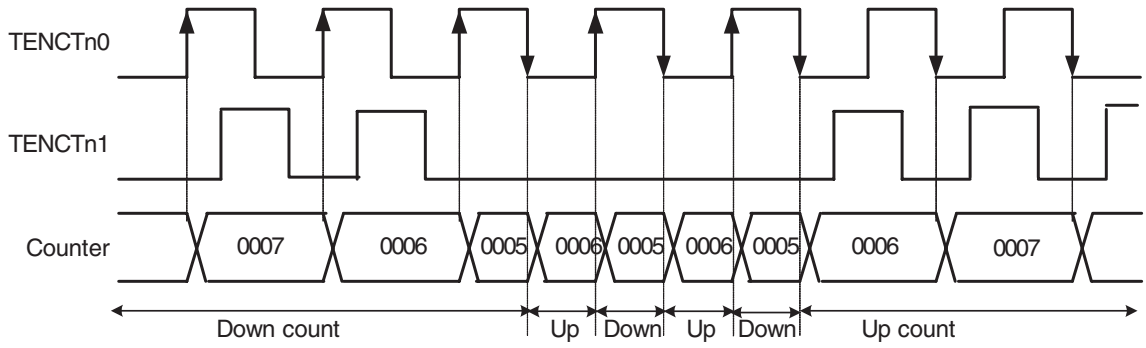
A Phase (Pin TENCTn0)	B Phase (Pin TENCTn1)	Count
Low level	Falling edge	Hold
Rising edge	Low level	Down
High level	Rising edge	Hold
Falling edge	High level	
Rising edge	High level	
High level	Falling edge	
Falling edge	Low level	Up
Low level	Rising edge	Hold
Rising edge	Rising edge	Hold
Falling edge	Rising edge	
Rising edge	Falling edge	Down
Falling edge	Falling edge	Up

Operation example:

TTnIOC3: TTnEIS3 to 0 (Pins TENCTn1, TENCTn0) Edge detection specification invalid

Figure 11-35: Encoder Count Function Up/Down Count Selection Specification Timings (3/6)

(c) Timing 3



Remark: n = 0, 1

<4> TTnCTL2: TTnUDS1, 0 = 11B (count judgment mode 4)

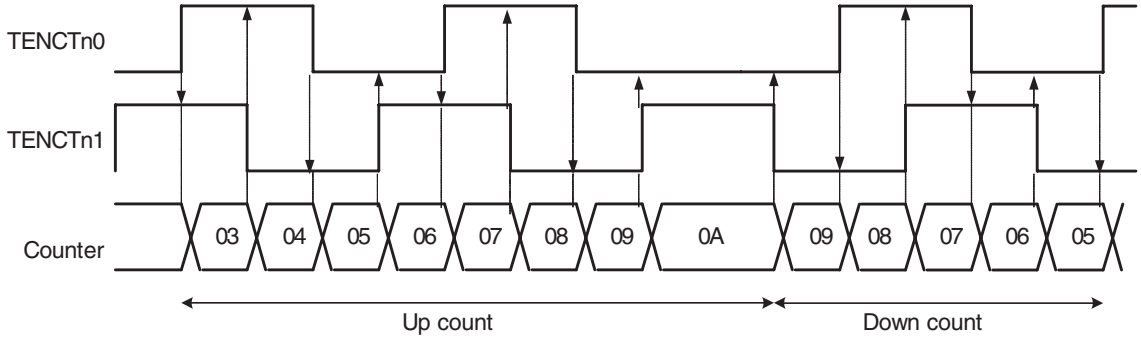
A Phase (Pin TENCTn0)	B Phase (Pin TENCTn1)	Count
Low level	Falling edge	Down
Rising edge	Low level	
High level	Rising edge	
Falling edge	High level	
Rising edge	High level	Up
High level	Falling edge	
Falling edge	Low level	
Low level	Rising edge	
Simultaneous pin TENCTn0/TENCTn1 inputs		Hold

Operation example 1:

TTnIOC2: TTnEIS3 to 0 (pins TENCTn1, TENCTn0) edge detection specification invalid

Figure 11-35: Encoder Count Function Up/Down Count Selection Specification Timings (4/6)

(d) Timing 4



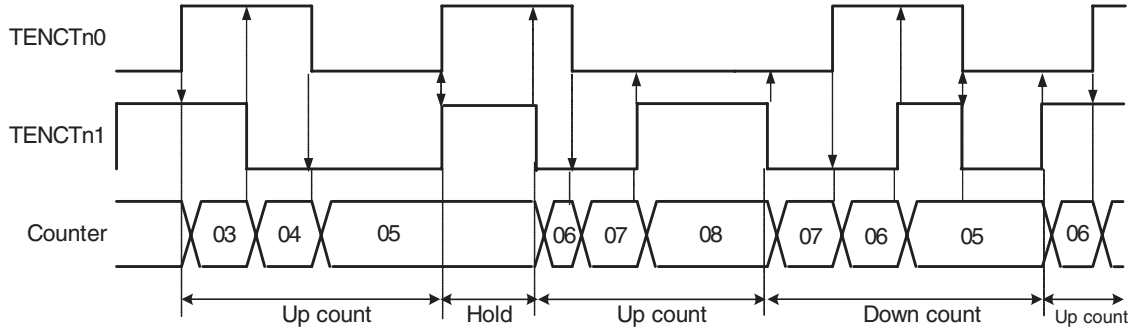
Remark: n = 0, 1

Operation example 2:

TTnIOC2: TTnEIS3 to 0 (pins TENCTn1, TENCTn0) edge detection specification invalid.

Figure 11-35: Encoder Count Function Up/Down Count Selection Specification Timings (5/6)

(e) Timing 5



Remarks: 1. The count value is held when the edges of the TENCTn0/TENCTn1 pin inputs overlap.

2. n = 0, 1

(b) Counter clear condition setting upon match between counter value and compare setting value (TTnCTL2 register bits TTnECM1, TTnECM0)

Counter operation is performed according to the setting values of these bits upon a match between the counter value and the compare setting value.

<1> TTnECM1, 0 = 00B
Counter clear is not performed upon a match between the counter and compare values.

<2> TTnECM1, 0 = 01B
Counter clear is performed upon a match between the counter and the TTnCCR0 register.

Next count Operation	Description
Up count	Clear counter to 0000H.
Down count	Down count the counter value.

<3> TTnECM1, 0 = 10B
Operation is performed under the following conditions upon a match between the counter and TTnCCR1 register.

Next Count Operation	Description
Up count	Up count the counter value.
Down count	Clear counter to 0000H.

<4> TTnECM1, 0 = 11B

- Operation is performed under the following conditions upon a match between the counter and TTnCCR0 register.

Next count Operation	Description
Up count	Clear counter to 0000H.
Down count	Down count the counter value.

- Operation is performed under the following conditions upon a match between the counter and TTnCCR1 register.

Next count Operation	Description
Up count	Up count the counter value.
Down count	Clear counter to 0000H.

Caution: In encoder compare mode (TTnMD3 to TTnMD0 bits = 1000B), if the compare registers (TTnCCR0, TTnCCR1) are set to the same value of TTnTCW register when TTnECC bit = 0, the timer cannot perform the comparison with the compare registers (TTnCCR0, TTnCCR1) and TTnTCW register (which is the start value of TTnCNT). In this case the "encoder clear mode on match of counter and compare register" does not work at the start timing (TTnECM0 = 1, and/or TTnECM1 = 1).

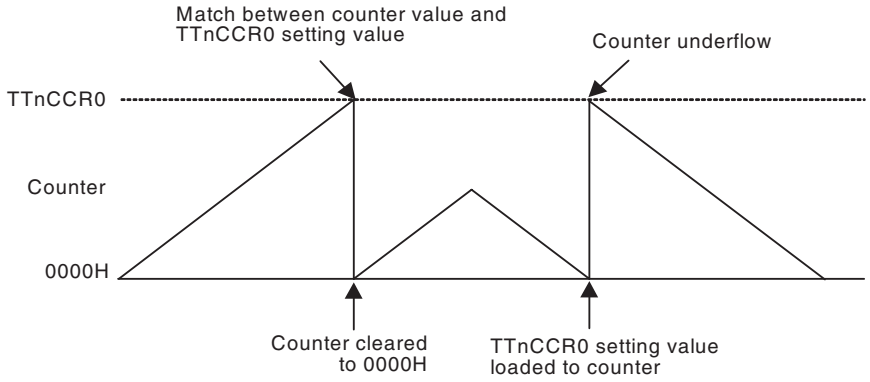
(c) Counter load function for TTnCCR0 register setting value upon underflow (bit TTnLDE of register TTnCTL2))

The setting value of the TTnCCR0 register can be loaded to the counter upon counter underflow, by setting TTnLDE = 1.
Bit TTnLDE is only valid in the encoder compare mode.

Count operation between 0000H and setting value of TTnCCR0 register setting
Set TTnLDE = 1, TTnECM1, 0 = 01B and perform count operation. When TTnECM0 = 1, the counter is cleared to 0000H if the next count following a match between the counter and TTnCCR0 register is up count.
When TTnLDE = 1, the setting value of the TTnCCR0 register is loaded to the counter upon underflow.
Therefore, the setting value of the TTnCCR0 register is used as the maximum count value and count operation can be realized within 0000H-TTnCCR0 register setting values.

Figure 11-35: Encoder Count Function Up/Down Count Selection Specification Timings (6/6)

(f) Timing 6



Remark: n = 0, 1

(5) Counter clearing to 0000H through encoder clear input (pin TECRTn) (TTnIOC3 register bits TTnSCE, TTnECS1, TTnECS0)

There are two methods to clear the counter to 0000H through TECRTn pin input, and encoder clear input is controlled by bit TTnSCE. Bits TTnZCL, TTnBCL, TTnACL, TTnECS1, and TTnECS0 are controlled by the setting of bit TTnSCE.
These clear methods are valid in the encoder compare mode.

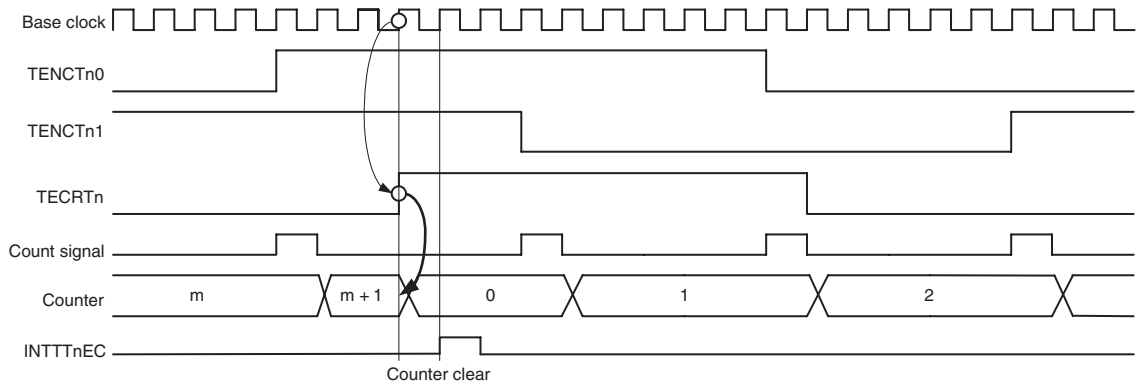
TTnSCE	TTnZCL	TTnBCL	TTnACL	TTnECS1, 0	Method
0	Invalid	Invalid	Invalid	√	<1>
1	√	√	√	Invalid	<2>

<1> Method to clear counter to 0000H through detection of valid edge of TECRTn pin input (TTnSCE = 0)

When TTnSCE = 0, the counter is cleared to 0000H in synchronization with the internal operation clock upon detection of the valid edge set through TECRTn pin input edge detection specification. At this time, an encoder clear interrupt (INTTTnEC) is output. When TTnSCE = 0, the setting of bits TTnZCL, TTnBCL, and TTnACL are invalid.

Figure 11-36: Counter Clearing to 0000H through Encoder Clear Input (pin TECRTn) Timings (1/4)

(a) When TTnSCE = 0, TTnECS1, 0 = 01B, TTnUDS = 11B are Set



Remark: n = 0, 1

<2> Method to clear counter to 0000H through detection of level clear condition (TTnSCE = 1)

When TTnSCE = 1, the counter is cleared to 0000H according to the clear condition level of pins TECRTn, TENCTn1, and TENCTn0 set with bits TTnZCL, TTnBCL, and TTnACL. At this time, no encoder clear interrupt (INTTTnEC) is output. When TTnSCE = 1, the settings of bits TTnECS1 and TTnECS0 are invalid.

Operation example:

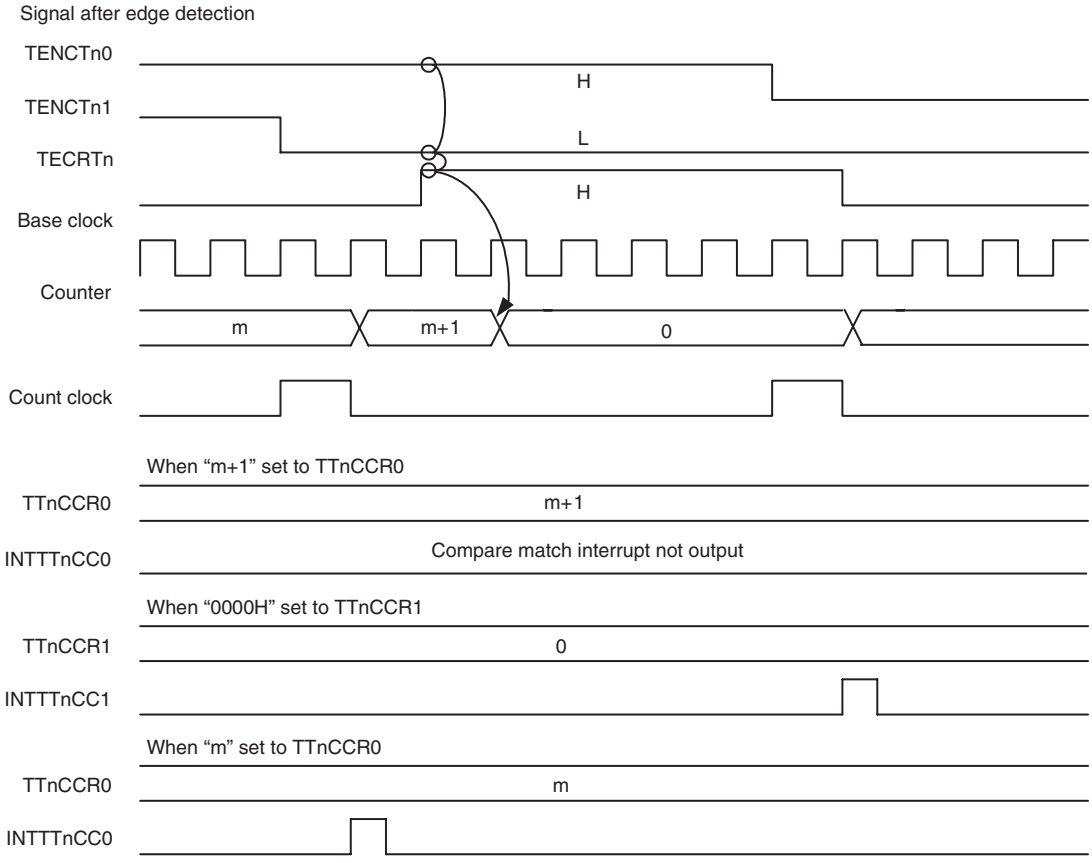
When TTnSCE = 1, TTnCLA = 1, TTnCLB = 0, TTnCLZ = 1, TTnUDS = 11B are set

<Clear condition level>

TECRTn pin: High level, TENCTn1 pin: Low level, TENCTn0 pin: High level

Figure 11-36: Counter Clearing to 0000H through Encoder Clear Input (pin TECRTn) Timings (2/4)

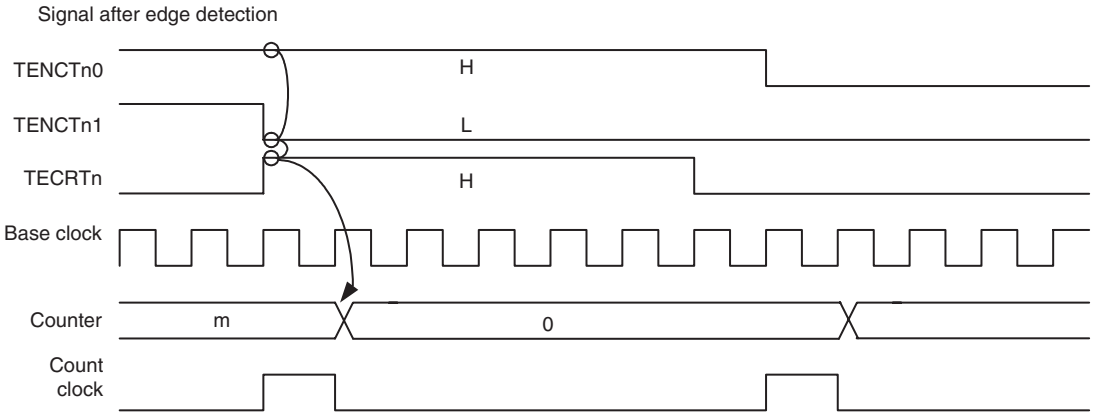
(b) when TECRTn Pin Input Is Delayed from TENCTn1 Pin Input during Up Count



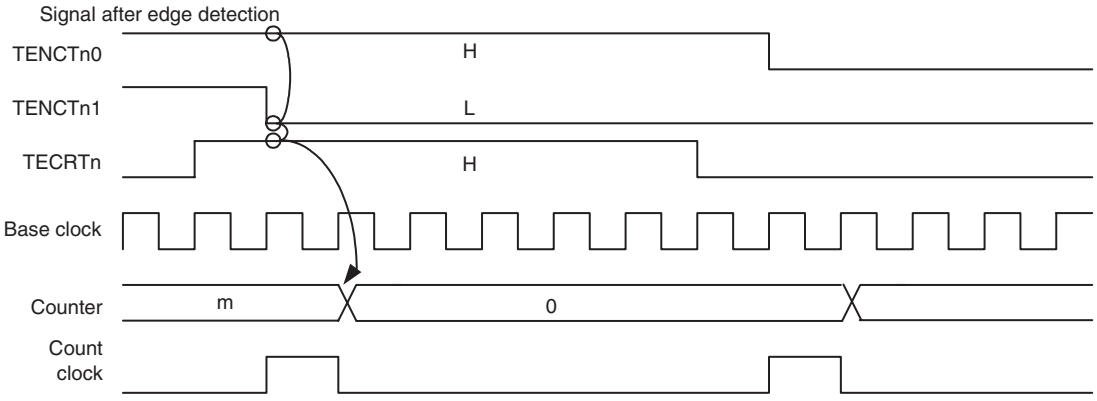
Remark: n = 0, 1

Figure 11-36: Counter Clearing to 0000H through Encoder Clear Input (pin TECRTn) Timings (3/4)

(c) when TECRTn Pin Input and TENCTn1 Pin Input Occur Simultaneously During Up Count



(d) when TECRTn Pin Input Occurs Earlier Than TENCTn1 Pin Input During Up Count

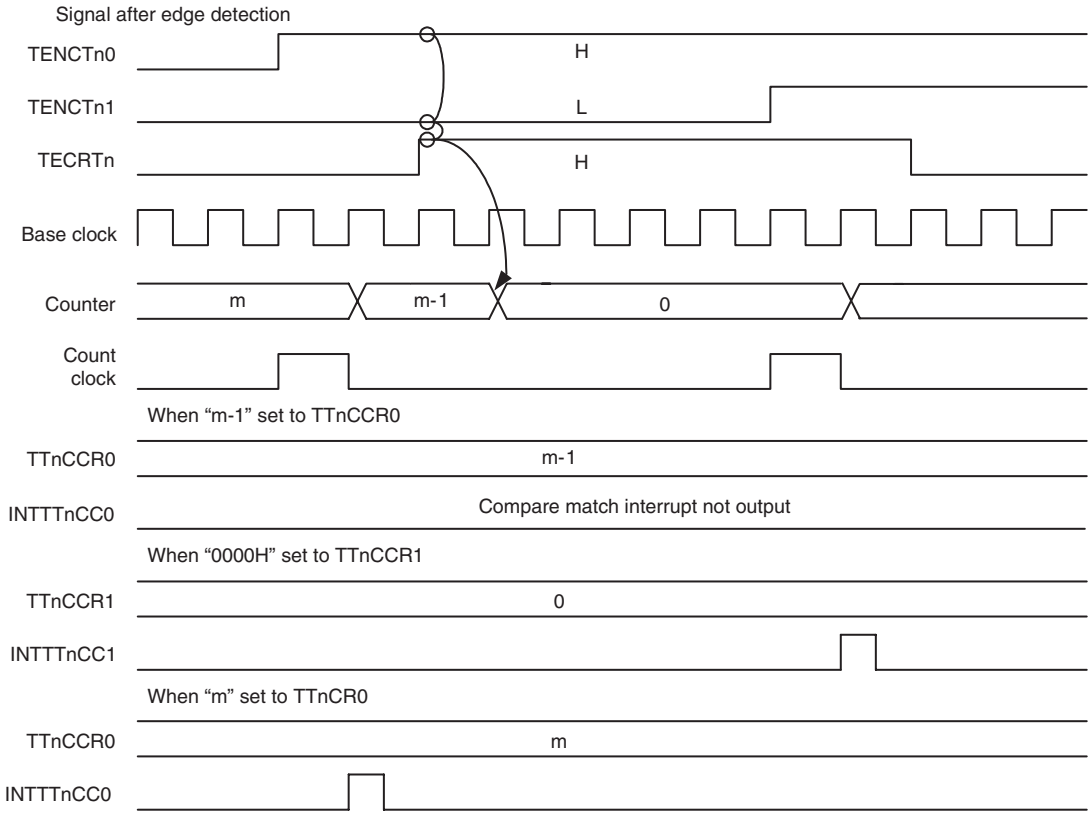


No miscount occurs due to TECRTn pin input delay because the clear condition is set according to the levels of pins TENCTn0, TENCTn1 and TECRTn, and the counter is cleared to 0000H upon clear condition detection.

Remark: n = 0, 1

Figure 11-36: Counter Clearing to 0000H through Encoder Clear Input (pin TECRTn) Timings (4/4)

(e) when TECRTn Pin Input Occurs Later Than TENCTn1 Pin Input During Down Count



No miscount occurs due to the TECRTn pin input delay during down count, similarly to during up count.

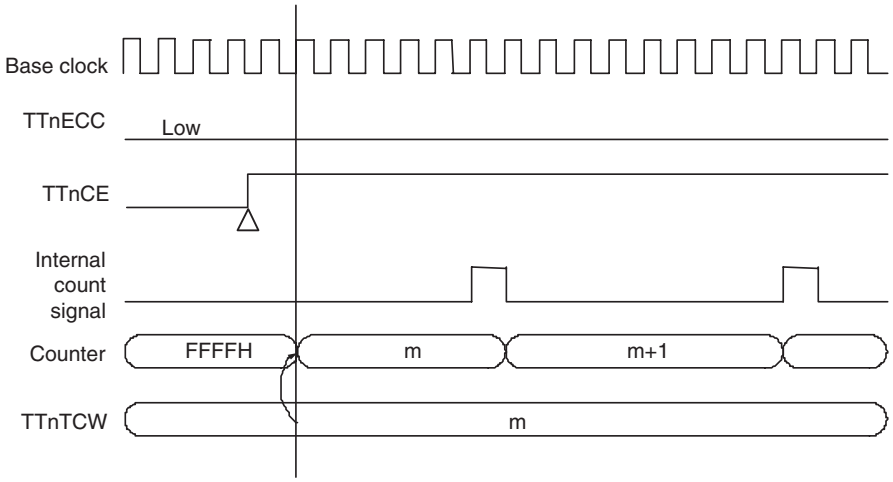
Remark: n = 0, 1

(6) Counter hold through bit TTnECC

(a) Initial counter operation through bit TTnECC setting

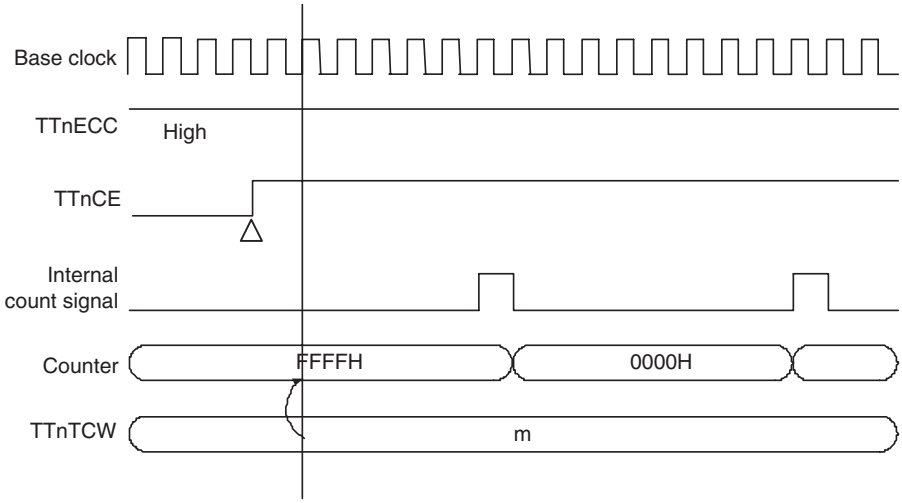
Figure 11-37: Counter Hold through Bit TTnECC Timings (1/5)

(a) Count operation when TTnECC = 0 is set



The setting value of the TTnTCW register is loaded to the counter and count operation is performed from the setting value of the TTnTCW register.
(Initial value 0000H of TTnTCW register)

(b) Count operation when TTnECC = 1 is set



Since the setting value of the TTnTCW register is not loaded to the counter, the count operation is performed from initial value FFFFH.
As the initial operation, it is recommended to set TTnECC = 0 and load to the counter the value set to the TTnTCW register, then start the count operation.

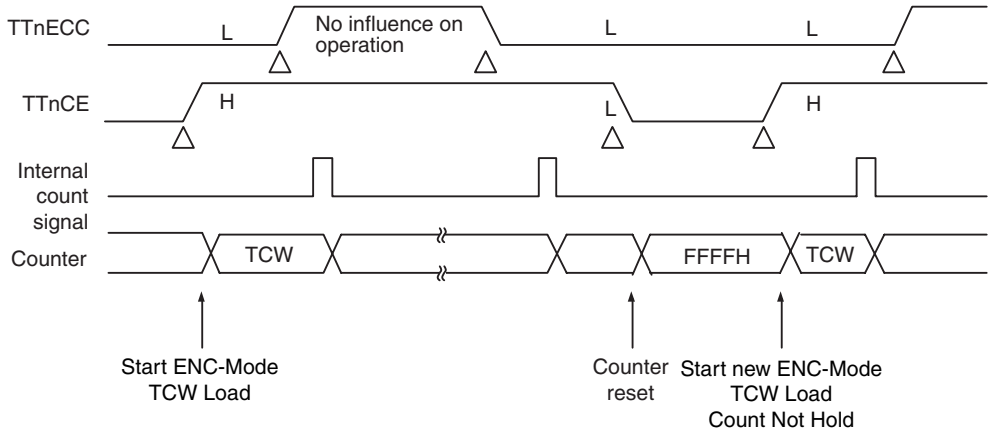
Remark: n = 0, 1

(b) Bit TTnECC rewrite timing and its influence on counter

<1> When setting value of bit TTnECC is rewritten 0 →1 →0 when TTnCE = 1
Even if bit TTnECC rewrite is performed while TTnCE = 1, this has no influence on the counter operation.
Judgment as whether to hold or reset the counter value is performed while TTnCE = 0.
Moreover, judgment as to whether to load the setting value of the TTnTCW register to the counter is performed at the timing when the value of bit TTnCE changes from 0 to 1.

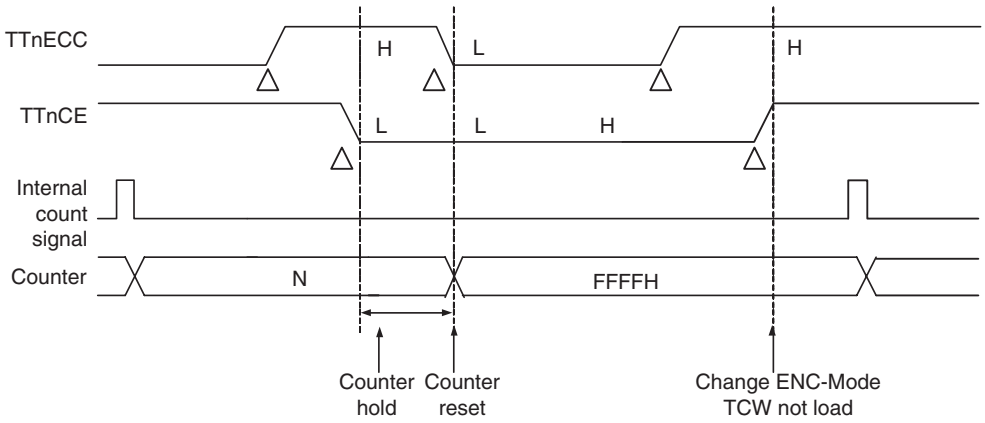
Figure 11-37: Counter Hold through Bit TTnECC Timings (2/5)

(c) when setting value of bit TTnECC is rewritten 0 →1 →0 when TTnCE = 1



<2> When setting value of bit TTnECC is rewritten 1 →0 →1 while TTnCE = 0
The counter is reset when the setting value of bit TTnECC is changed from 1 to 0 while TTnCE = 0. Then, when TTnECC = 1 is set again and the value of bit TTnCE is changed from 0 to 1, counting restarts from the counter's initial value FFFFH, without the setting value of the TTnTCW being loaded to the counter.

(d) when setting value of bit TTnECC is rewritten 1 →0 →1 while TTnCE = 0



Remark: n = 0, 1

(c) Rewrite timing of bit TTnECC

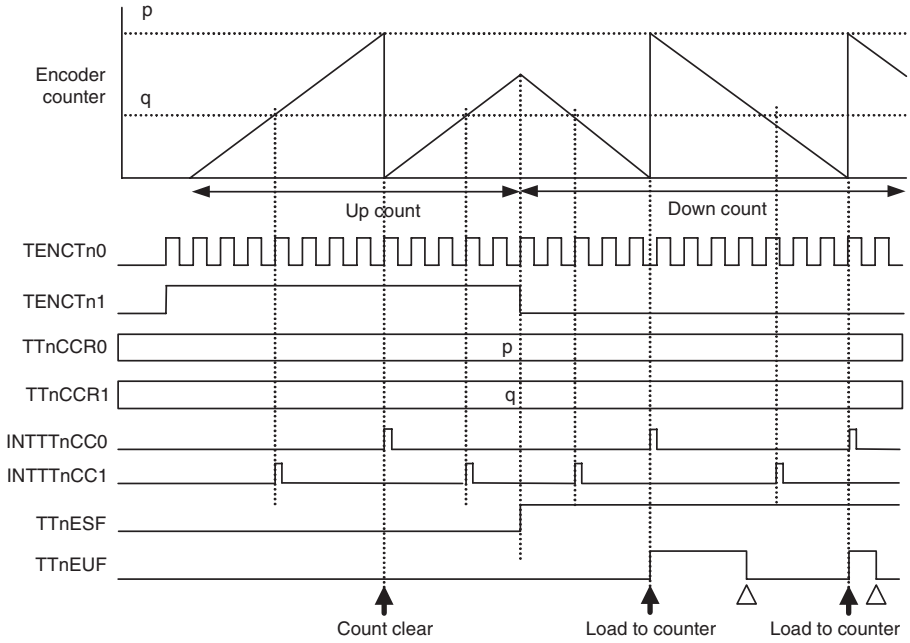
When TTnCE = 0 and TTnECC = 0, setting TTnCE = 1 causes the setting value of the TTnTCW register to be loaded to the counter. Perform rewrite of the TTnECC bit after the operation clock has become valid (after several clocks: TBD), following setting of TTnCE = 1.
If bit TTnECC is rewritten before the operation clock becomes valid, counting starts from FFFFH without loading the setting value of the TTnTCW register to the counter.

Figure 11-37: Counter Hold through Bit TTnECC Timings (3/5)

(e) Basic Timing in Encoder Compare Mode (1)

< Register setting conditions>

- TTnCTL0: TTnMD3 to 0 = 1000B — Encoder compare mode
- TTnCTL1: TTnUDS1, 0 = 00B — Judgment of up/down count with count judgment mode 1
- TTnCTL1: TTnECM1, 0 = 01B — Counter clear upon match between counter value and TTnCCR0 buffer register
- TTnCTL1: TTnLDE = 1 — Loading of setting value of TTnCCR0 register (p) upon underflow occurrence
- TTnIOC3: TTnEIS1, 0 = 01B — Detection of rising edge of TENCTn0 and TENCTn1 pin inputs
- TTnIOC3: TTnSCE = 0, TTnECS1-0 = 00B — Valid edge detection clear (no edge specified)



Since TTnUDS1, 0 and TTnEIS1, 0 that control the count operation are set to 00B and 01B (rising edge detection), respectively, the counter is operated through detection of the phase of pin TENCTn1 upon detection of the rising edge of TENCTn0 pin input. A compare match interrupt (INTTTnCC0) is output upon a match between the counter value and the TTnCCR0 compare register (p). At this time, the counter is cleared to 0000H if the next count operation is up count.
A compare match interrupt (INTTTnCC1) is output upon a match between the counter value and the TTnCCR1 buffer register (q). The counter is not cleared upon a match between the counter value and the TTnCCR1 register. If underflow occurs when TTnLDE = 1 is set, the setting value of the TTnCCR0 buffer register (m) is loaded to the counter. A count operation is possible between 0000H and the setting value of the TTnCCR0 register by setting TTnLDE = 1 and TTnECM0 = 1.

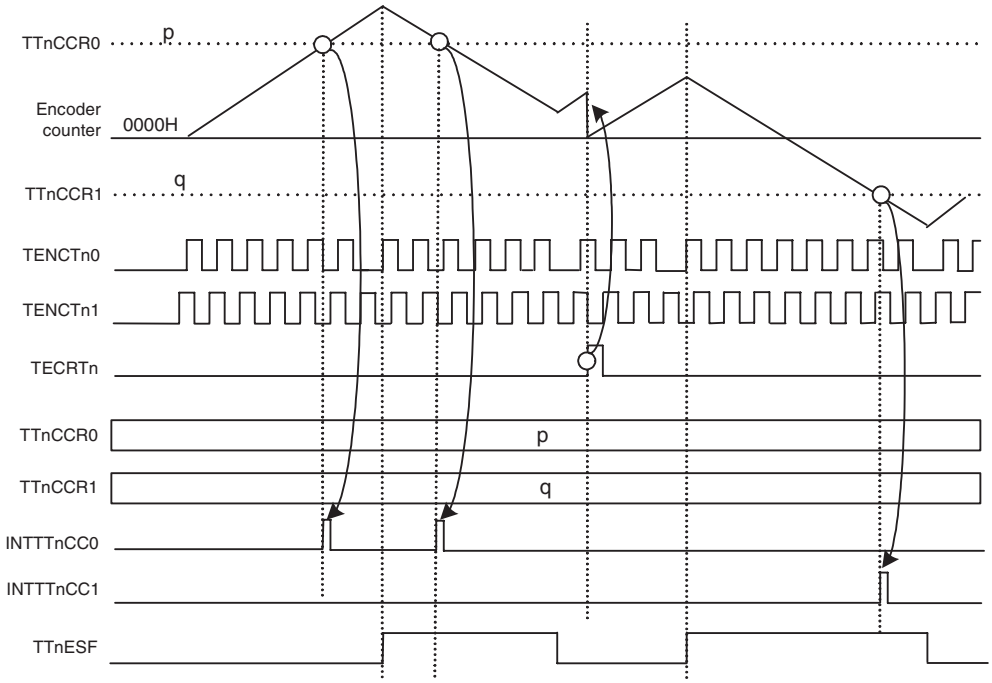
Remark: n = 0, 1

Figure 11-37: Counter Hold through Bit TTnECC Timings (4/5)

(f) Basic Timing in Encoder Compare Mode (2)

<Setting conditions>

- TTnCTL0: TTnMD3 to 0 = 1000B — Encoder compare mode
- TTnCTL1: TTnUDS1, 0 = 11B — Judgment of up/down count with count judgment mode 4
- TTnCTL1: TTnECM1, 0 = 00B — No clear operation upon match between counter value and compare
- TTnCTL1: TTnLDE = 0 — No loading of setting value of TTnCCR0 register (p) to counter
- TTnIOC3: TTnSCE = 0, TTnECS1-0 = 01B — Valid edge detection clear (rising edge specified)



Since TTnUDS1, 0 that control the count operation are set to 11B, the counter is operated through detection of the phase of pins TENCTn0 and TENCTn1.
A compare match interrupt (INTTTnCC0) is output upon a match between the counter value and the TTnCCR0 buffer register (p).
A compare match interrupt (INTTTnCC1) is output upon a match between the counter value and the TTnCCR1 buffer register (q).
The counter is not cleared upon a match with the TTnCCR0 register or the TTnCCR1 register.
Clearing of the counter to 0000H is done upon detection of the valid edge of the encoder clear input (pin TECRTn) when TTnSCE = 0. When TTnECS = 01B is set, the counter is cleared to 0000H in synchronization with the operation clock, following detection of the rising edge of the TECRTn pin input.

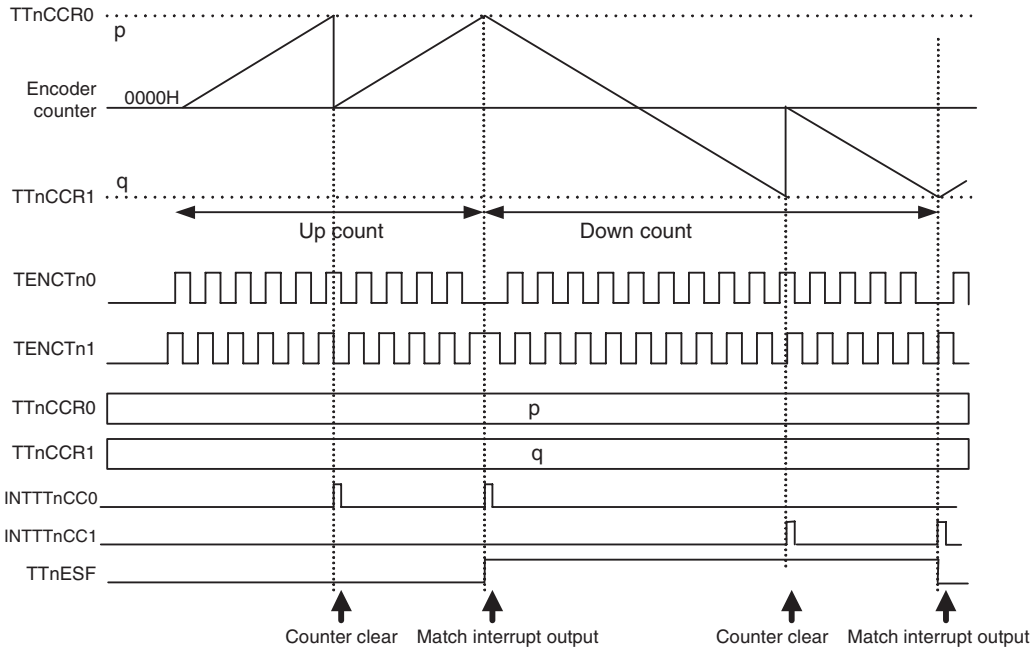
Remark: n = 0, 1

Figure 11-37: Counter Hold through Bit TTnECC Timings (5/5)

(g) Basic Timing in Encoder Compare Mode (3)

<Setting conditions>

- TTnCTL0: TTnMD3 to 0 = 1000B — Encoder compare mode
- TTnCTL1: TTnUDS1 to 0 = 11B — Judgment of up/down count with count judgment mode 4
- TTnCTL1: TTnECM1 to 0 = 11B — Counter clear upon match between counter value and TTnCCR0 buffer register
  Counter clear upon match between counter value and TTnCCR1 buffer register
  (Since TTnCTL1: TTnECM1 to 0 = 11B, the setting of bit TTnLDE is invalid.)
- TTnIOC3: TTnSCE = 0, TTnECS1 to 0 = 00B — Valid edge detection clear (no edge specified)



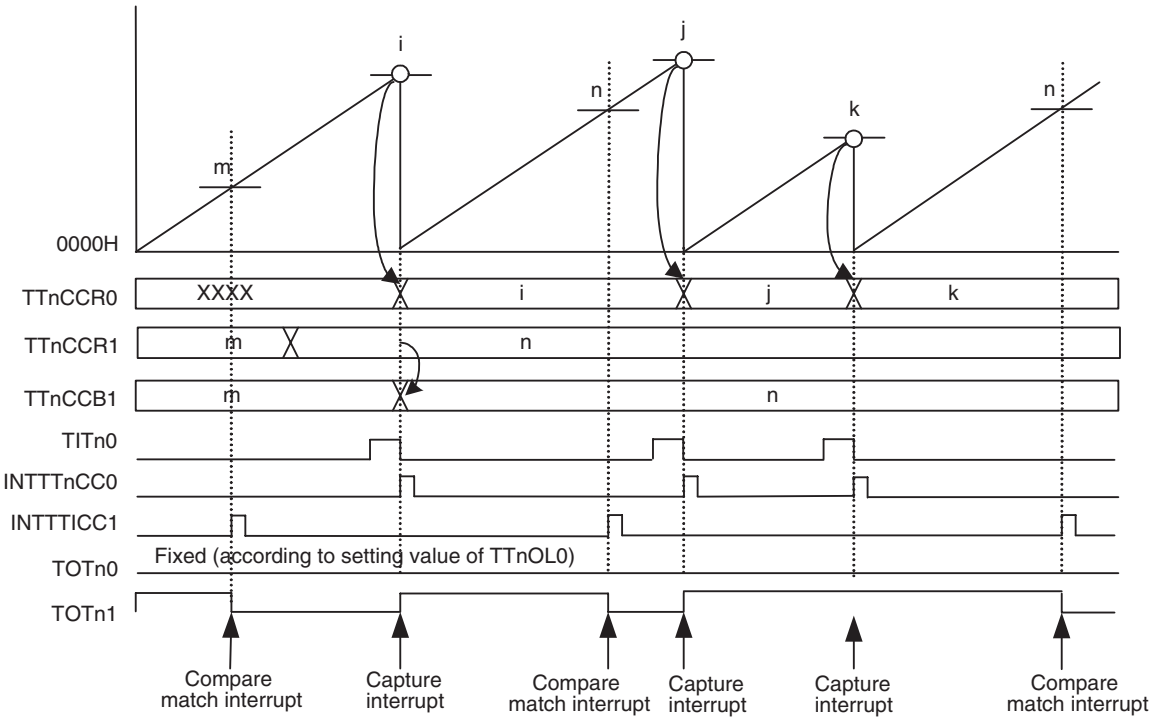
Since TTnUDS1, 0 that control the count operation are set to 11B, the counter is operated through detection of the phase of pins TENCTn0 and TENCTn1.
A compare match interrupt (INTTTnCC0) is output upon a match between the counter value and the TTnCCR0 buffer (p).
At this time, the counter is cleared to 0000H if the next count operation is up count.
A compare match interrupt (INTTTnCC1) is output upon a match between the counter value and the TTnCCR1 buffer (q).
At this time, the counter is cleared to 0000H if the next count operation is down count.

Remark: n = 0, 1

11.6.10 Offset trigger generation mode

In the offset trigger generation mode, the count value is saved to the capture register (TTnCCR0) upon detection of the valid edge of the TITn0 pin, and a capture interrupt (INTTTnCC0) is output. The counter is cleared to 0000H by capture input. (Counter clear operation is not performed using the TTnCCR1 register.)
The TTnCCR0 register and the TTnCCR1 register have their functions fixed as a capture register and a compare register, respectively. The TTnCCR1 register can be rewritten during count operation. Regarding compare register reload, the capture & clear timing upon detection of TITn0 pin input serves as the reload timing.
During count operation, a capture interrupt (INTTTnCC0) is output upon capture to the TTnCCR0 register through TITn0 pin input, and a compare match interrupt (INTTTnCC1) is output upon a match between the counter and the TTnCCR1 register.
The TOTn0 pin becomes the level set with bit TTnOL0. If TTnOL0 = 0, a low level is output a and if TTnOL0 = 1, a high level is output.
The TOTn1 pin is reset upon a match between the counter and the TTnCCR1 register, and is set when the counter is cleared to 0000H.

Figure 11-38: Basic Timing in Offset Trigger Generation Mode



Remark: n = 0, 1

In the offset trigger generation mode, the setting value of the TTnCCR1 register is reloaded to the TTnCCR1 buffer register upon detection of the valid edge of pin TITn0. Until the edge of the TITn0 pin input is detected, the value of the TTnCCR1 register is not reloaded to the TTnCCR1 buffer register, even if this value is changed.
Pin TOTn1 is set when the counter is cleared to 0000H upon detection of the valid edge of pin TITn0, and it is reset upon a match between the counter value and the TTnCCR1 register.
Therefore, pin TOTn1 remains high level if the valid edge of the TITn0 pin input is detected before a match with the TTnCCR1 register occurs.

[MEMO]

Chapter 12 16-bit 2-Phase Encoder Input Up/Down Counter/General Purpose Timer (TMENC10)

12.1 Features

Timer ENC10 (TMENC10) is a 16-bit up/down counter that performs the following operations.

- General-purpose timer mode:
  - Free-running timer
  - PWM output
- Up/down counter mode
  - UDC mode A
  - UDC mode B

12.2 Function Outline

- Compare register × 2
- Capture/compare register × 2
- Interrupt request source
  - Capture/compare match interrupt × 2
  - Compare match interrupt × 2
  - Overflow interrupt × 1
  - Underflow interrupt × 1
- Capture request signal × 2
  - The TMENC10 value can be latched using the valid edge of the TICC10, TICC11 pins corresponding to the capture/compare register as the capture trigger.
- Base clock (f_{CLK}) = f_{XX}/4
  (f_{CLK} = 16 MHz @ f_{XX} = 64 MHz)
- Count clocks selectable through division by prescaler
- 2-phase encoder input
  The 2-phase encoder signal from external is used as the count clock of the timer counter with the external clock input pins (TIUD1, TCUD1). The counter mode can be selected from among the four following modes.
  - Mode 1: Counts the input pulses of the count pulse input pin. Up/down is specified by the level of one more input pin.
  - Mode 2: Counts up/down using the respective input pulses of the up count pulse input pin and down count pulse input pin.
  - Mode 3: Counts up/down using the phase relationship of the pulses input to 2 pins.
  - Mode 4: Counts up/down using the phase relationship of the pulses input to 2 pins. Counting is done using the respective rising edges and the falling edges of the pulses.
- PWM output function
  - In the general-purpose timer mode, 16-bit resolution PWM output can be output from the TO1 pin.

- Timer clear
  - The following timer clear operations are performed according to the mode that is used.

    (a) General-purpose timer mode: Timer clear operation is possible upon occurrence of match with CM100 set value.

    (b) Up/down counter mode: The timer clear operation can be selected from among the following four conditions.

  - Timer clear performed upon occurrence of match with CM100 set value during TMENC10 up count operation, and timer clear performed upon occurrence of match with CM101 set value during TMENC10 down count operation.
  - Timer clear performed only by external input.
  - Timer clear performed upon occurrence of match between TMENC10 count value and CM100 set value.
  - Timer clear performed upon occurrence of external input and match between TMENC10 count value and CM100 set value.

- External pulse output (TO1) × 1

Remark: f_{XX}: Internal system clock

12.3 Basic Configuration

The basic configuration is shown below.

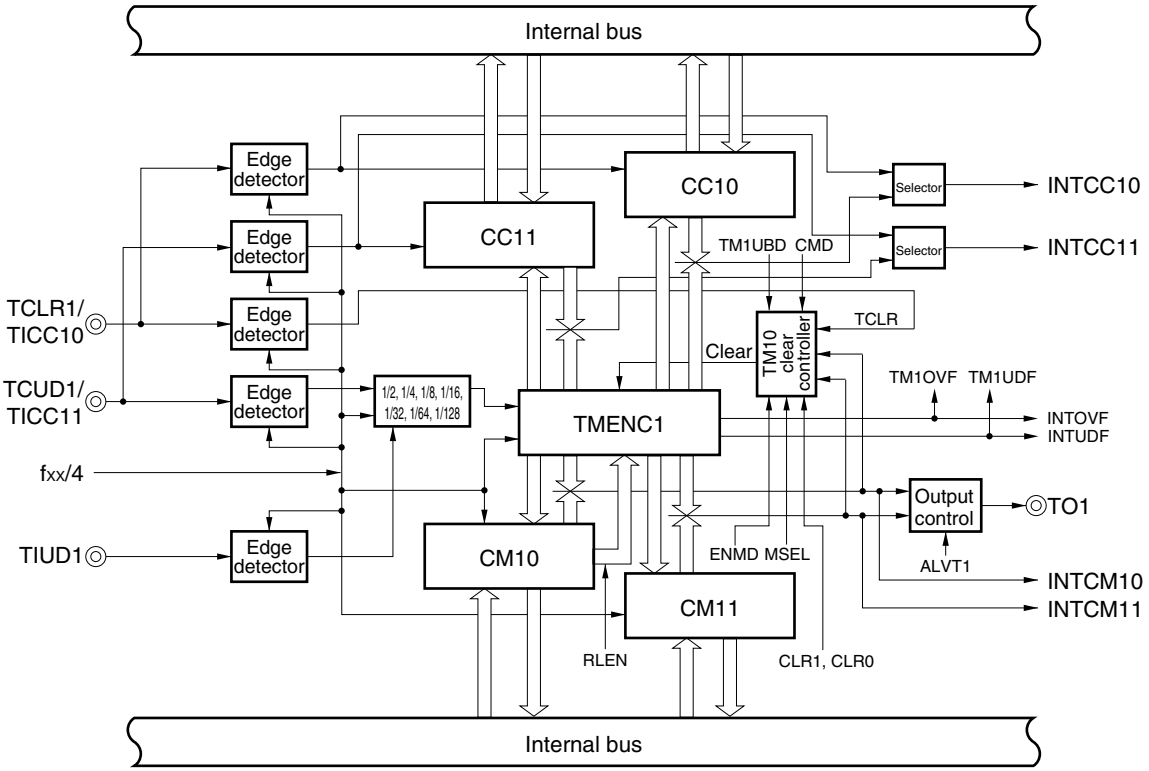
Table 12-1: Timer ENC10 Configuration List

Timer	Count Clock	Register	Read/Write	Generated Interrupt Signal	Capture Trigger
Timer ENC10	$f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, $f_{XX}/512$	TMENC10	Read/write	INTOVF INTUDF	-
		CM100	Read/write	INTCM10	-
		CM101	Read/write	INTCM11	-
		CC100	Read/write	INTCC10	TICC10
		CC101	Read/write	INTCC11	TICC11

Remark: f_{XX}: Internal system clock

Figure 12-1 shows the block diagram of timer ENC10.

Figure 12-1: Block Diagram of Timer ENC10 (TMENC10)



Note: The TICC11 interrupt is the signal of the interrupt from the TICC11 pin or the interrupt from the TICC10 pin, selected by the CSL bit of the CSL1 register.

Remark: f_{XX}: Internal system clock

(1) Timer ENC10 (TMENC10)

TMENC10 is a 2-phase encoder input up/down counter and general-purpose timer.
It can be read/written in 16-bit units.
Reset input clears TMENC10 to 0000H.

Cautions: 1. Write to TMENC10 is enabled only when the TM1CE bit of the TMC10 register is "0" (count operation disabled).

2. It is prohibited to clear the CMD bit (general-purpose timer mode) to 0 and to set the MSEL bit (UDC mode B) of the TUM register to 1.

3. Continuous reading of TMENC10 is prohibited. If TMENC10 is continuously read, the second value read may differ from the actual value. If TMENC1n must be read twice, be sure to read another register between the first and the second read operation.

4. Writing the same value to the TMENC10, CC100, and CC101 registers, and the STATUS10 register is prohibited.
Writing the same value to the CCR10, TUM10, TMC10, SESA10, and PRM10 registers, and CM100 and CM101 registers is permitted (writing the same value is guaranteed even during a count operation).

Figure 12-2: Timer ENC10 (TMENC10)

After reset: 0000H R/W Address: FFFFF6B0H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMENC10																

TMENC10 start and stop is controlled by the TM1CE bit of timer control register 10 (TMC10).
The TMENC10 operation consists of the following two modes.

(a) General-purpose timer mode

In the general-purpose timer mode, TMENC10 operates as a 16-bit interval timer, free-running timer, or for PWM output.
Counting is performed based on the clock selected by software.
Division by the prescaler can be selected for the count clock from among $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, or $f_{XX}/512$ with bits PRM102 to PRM100 of prescaler mode register 10 (PRM10) (f_{XX}: internal system clock).

(b) Up/down counter mode (UDC mode)

In the UDC mode, TMENC10 functions as a 16-bit up/down counter, counting based on the TCUD1 and TIUD1 input signals.
Two operation modes can be set with the MSEL bit of the TUM register for this mode.

- **UDC mode A (when CMD bit = 1, MSEL bit = 0)**

TMENC10 can be cleared by setting the CLR1 and CLR0 bits of the TMC10 register.

- **UDC mode B (when CMD bit = 1, MSEL bit = 1)**

TMENC10 is cleared upon match with CM100 during TMENC10 up count operation.
TMENC10 is cleared upon match with CM101 during TMENC10 down count operation.

When the TM1CE bit of the TMC10 register is “1”, TMENC10 counts up when the operation mode is the general-purpose mode, and counts up/down when the operation mode is the UDC mode.

The conditions for clearing the TMENC10 are classified as follows depending on the operation mode.

Table 12-2: Timer ENC10 (TMENC10) Clear Conditions

Operation Mode	TUM10 Register		TMC10 Register			TMENC10 Clear
	CMD Bit	MSEL Bit	ENMD Bit	CLR1 Bit	CLR0 Bit	
General-purpose timer mode	0	0	0	×	×	Clearing not performed
			1	×	×	Cleared upon match with CM100 set value
UDC mode A	1	0	×	0	0	Cleared only by TCLR1 input
			×	0	1	Cleared upon match with CM1n0 set value during up count operation
			×	1	0	Cleared by TCLR1 input or upon match with CM100 set value during up count operation
			×	1	1	Clearing not performed
UDC mode B	1	1	×	×	×	Cleared upon match with CM100 set value during up count operation or upon match with CM101 set value during down count operation
Settings other than the above						Setting prohibited

Remark: ×: Indicates that the set value of that bit is ignored.

(2) Compare register 100 (CM100)

CM100 is a 16-bit register that always compares its value with the value of TMENC10. When the value of a compare register matches the value of TMENC10, an interrupt signal is generated. The interrupt generation timing in the various modes is described below.

- In the general-purpose timer mode (CMD bit of TUM10 register = 0) and UDC mode A (MSEL bit of TUM10 register = 0), an interrupt signal (INTCM10) is always generated upon occurrence of a match.
- In UDC mode B (MSEL bit of TUM10 register = 1), an interrupt signal (INTCM10) is generated only upon occurrence of a match during up count operation.

This register can be read/written in 16-bit units.
Reset input clears this register to 0000H.

Caution: When the TM1CE bit of the TMC10 register is 1, it is prohibited to overwrite the value of the CM100 register.

Figure 12-3: Compare Register 100 (CM100)

After reset: 0000H R/W Address: FFFFF6B2H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CM100																

(3) Compare register 101 (CM101)

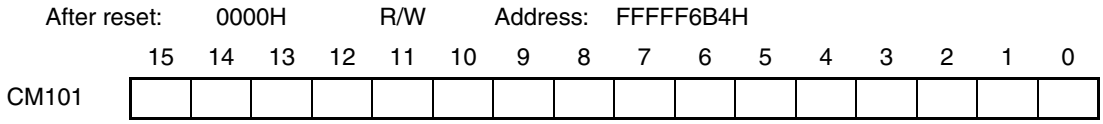
CM101 is a 16-bit register that always compares its value with the value of TMENC10. When the value of a compare register matches the value of TMENC10, an interrupt signal is generated. The interrupt generation timing in the various modes is described below.

- In the general-purpose timer mode (CMD bit of TUM10 register = 0) and UDC mode A (MSEL bit of TUM10 register = 0), an interrupt signal (INTCM11) is always generated upon occurrence of a match.
- In UDC mode B (MSEL bit of TUMn register = 1), an interrupt signal (INTCM11) is generated only upon occurrence of a match during down count operation.

This register can be read/written in 16-bit units.
Reset input clears this register to 0000H.

Caution: When the TM1CE bit of the TMC10 register is 1, it is prohibited to overwrite the value of the CM101 register.

Figure 12-4: Compare Register 101 (CM101)



(4) Capture/compare register 100 (CC100)

CC100 is a 16-bit register. It can be used as a capture register or as a compare register through specification with capture/compare control register n (CCR).
This register can be read/written in 16-bit units.
Reset input clears this register to 0000H.

Cautions: 1. When used as a capture register (CMS0 bit of CCR register = 0), write access is prohibited.

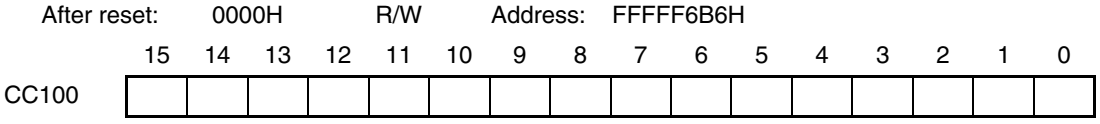
2. When used as a compare register (CMS0 bit of CCR register = 1) and the TM1CE bit of the TMC10 register is 1, overwriting the CC100 register values is prohibited.

3. When the TM1CE bit of the TMC10 register is 0, the capture trigger is disabled.

4. When the operation mode is changed from capture register to compare register, set a new compare value.

5. Continuous reading of CC100 is prohibited. If CC100 is continuously read, the second read value may differ from the actual value. If CC100 must be read twice, be sure to read another register between the first and the second read operation.

Figure 12-5: Capture/Compare Register 100 (CC100)



(a) When set as a capture register

When CC100 is set as a capture register, the valid edge of the corresponding external TICC10 signal is detected as the capture trigger. TMENC10 latches the count value in synchronization with the capture trigger (capture operation). The latched value is held in the capture register until the next capture operation.
The valid edge of external interrupts (rising edge, falling edge, both edges) is selected with signal edge selection register 10 (SESA10).
When the CC100 register is specified as a capture register, an INTCC10 interrupt is generated upon detection of the valid edge of the external TICC10 signal.

(b) When set as a compare register

When CC100 is set as a compare register, it always compares its own value with the value of TMENC10. If the value of CC100 matches the value of the TMENC10, CC100 generates an interrupt signal (INTCC10).

(5) Capture/compare register 101 (CC101)

CC101 is a 16-bit register. It can be used as a capture register or as a compare register through specification with capture/compare control register (CCR).
This register can be read/written in 16-bit units.
Reset input clears this register to 0000H.

Cautions: 1. When used as a capture register (CMS1 bit of CCR register = 0), write access is prohibited.

2. When used as a compare register (CMS1 bit of CCRn register = 1) and the TM1CE bit of the TMC10 register is 1, overwriting the CC101 register values is prohibited.

3. When the TM1CE bit of the TMC10 register is 0, the capture trigger is disabled.

4. When the operation mode is changed from capture register to compare register, set a new compare value.

5. Continuous reading of CC101 is prohibited. If CC101 is continuously read, the second read value may differ from the actual value. If CC101 must be read twice, be sure to read another register between the first and the second read operation.

Figure 12-6: Capture/Compare Register 101 (CC101)

After reset: 0000H R/W Address: FFFFF6B8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC101																

(a) When set as a capture register

When CC101 is set as a capture register, the valid edge of the corresponding external TICC11 signal is detected as the capture trigger. TMENC10 latches the count value in synchronization with the capture trigger (capture operation). The latched value is held in the capture register until the next capture operation.
The valid edge of external interrupts (rising edge, falling edge, both edges) is selected with signal edge selection register 10 (SESA10).
When the CC101 register is specified as a capture register, an INTCC11 interrupt is generated upon detection of the valid edge of the external TICC11 signal.

(b) When set as a compare register

When CC101 is set as a compare register, it always compares its own value with the value of TMENC10. If the value of CC101 matches the value of the TMENC10, CC101 generates an interrupt signal (INTCC11).

12.4 Control Registers

(1) Timer unit mode register 10 (TUM10)

The TUM10 register is an 8-bit register used to specify the TMENC10 operation mode or to control the operation of the PWM output pin.
This register can be read/written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Cautions: 1. Changing the value of the TUM10 register during TMENC10 operation (TM1CE bit of TMC register = 1) is prohibited.

2. When the CMD bit = 0 (general-purpose timer mode), setting MSEL bit = 1 (UDC mode B) is prohibited.

Figure 12-7: Timer Unit Mode Register 10 (TUM10)

After reset: 00H R/W Address: FFFFF6BBH

	7	6	5	4	3	2	1	0
TUM10	CMD	0	0	0	TOE	ALVT1	0	MSEL

CMD	TMENC10 Operation Mode Specification
0	General-purpose timer mode (up count)
1	UDC mode (up/down count)

TOE	Timer Output (TO1) Control
0	Timer output disabled
1	Timer output enabled
When CMD bit = 1 (UDC mode), timer output is not performed regardless of the setting of the TOE bit. At this time, timer output consists of the inverted phase level of the level set by the ALVT1 bit.	

ALVT1	Active Level Specification for Timer Output (TO1)
0	Active level is high level
1	Active level is low level
When CMD bit = 1 (UDC mode), timer output is not performed regardless of the setting of the TOE bit. At this time, timer output consists of the inverted phase level of the level set by the ALVT1 bit.	

MSEL	Mode Selection in UDC Mode (Up/Down Count)
0	UDC mode A. TMENC10 can be cleared by setting the CLR1, CLR0 bits of the TMC10 register.
1	UDC mode B. TMENC10 is cleared in the following cases. • Upon match with CM100 during TMENC10 up count operation • Upon match with CM101 during TMENC10 down count operation
When UDC mode B is set, the ENMD, CLR1, and CLR0 bits of the TMC10 register become invalid.	

(2) Timer control register 10 (TMC10)

The TMC10 register is used to enable/disable TMENC10 operation and to set transfer and timer clear operations.

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

Reset input clears this register to 00H.

Caution: Changing the values of the TMC10 register bits other than the TM1CE bit during TMENC10 operation (TM1CE bit = 1) is prohibited.

Figure 12-8: Timer Control Register 10 (TMC10) (1/2)

After reset: 00H R/W Address: FFFFF6BCH

	7	6	5	4	3	2	1	0
TMC10	0	TM1CE	0	0	RLEN	ENMD	CLR1	CLR0

TM1CE	TMENC10 Operation Control
0	Count operation disabled
1	Count operation enabled

RLEN	Transfer Operation Control in UDC Mode A
0	Transfer operation from CM100 register to TMENC10 disabled
1	Transfer operation from CM100 register to TMENC10 enabled

- When RLEN = 1, the value set to CM100 is transferred to TMENC10 upon occurrence of TMENC10 underflow.
- When the CMD bit of the TUM10 register = 0 (general-purpose timer mode), the RLEN bit settings are invalid, and a transfer operation is not executed even if the RLEN bit is set to 1.
- When the MSEL bit of the TUM10 register = 1 (UDC mode B), the RLEN bit settings are invalid, and a transfer operation is not executed even if the RLEN bit is set to 1.

ENMD	Clear Operation Control in General Purpose Mode
0	Clear disabled (free-running mode) Clearing is not performed even when TMENC10 and CM100 values match.
1	Clear enabled Clearing is performed upon match of TMENC10 and CM100 values.

When the CMD bit of the TUM10 register = 1 (UDC mode), the ENMD bit setting becomes invalid.

Figure 12-8: Timer Control Register 10 (TMC10) (2/2)

CLR1	CLR0	Clear Operation Control in UDC Mode A
0	0	Clear only by external input (TCLR1)
0	1	Clear upon match of TMENC10 count value and CM100 set value
1	0	Clear by TCLR1 input or upon match of TMENC10 count value and CM100 set value
1	1	No clearing

- Clearing by match of the TMENC10 count value and CM100 set value is valid only during TMENC10 up count operation (TMENC10 is not cleared during TMENC10 down count operation).
- When the CMD bit of the TUM10 register = 0 (general-purpose timer mode), the CLR1 and CLR0 bit settings are invalid.
- When the MSEL bit of the TUM10 register = 1 (UDC mode B), the CLR1 and CLR0 bit settings are invalid.
- When clearing by TCLR1n has been enabled with bits CLR1 and CLR0, clearing is performed whether the value of the TM1CEn bit is 1 or 0.

(3) Capture/compare control register 10 (CCR10)

The CCR10 register specifies the operation mode of the capture/compare registers (CC100, CC101).

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

Reset input clears this register to 00H.

Cautions: 1. Overwriting the CCR10 register during TMENC10 operation (TM1CE bit = 1) is prohibited.

2. The TCUD1 pin is used for the UDC mode and shared with the external capture input pin TICC11. Therefore, in the UDC mode, the external capture function cannot be used.

3. The TCLR1 pin is used for the UDC mode and alternately shared with the external capture input pin TICC10. Therefore, when the TCLR1 input is used in UDC mode A, the external capture function cannot be used.

Figure 12-9: Capture/Compare Control Register 10(CCR10)

After reset: 00H R/W Address: FFFFF6BAH

	7	6	5	4	3	2	1	0
CCR10	0	0	0	0	0	0	CMS1	CMS0

CMS1	CC101 Operation Mode Specification
0	CC101 operates as capture register
1	CC101 operates as compare register

CMS0	CC100 Operation Mode Specification
0	CC100 operates as capture register
1	CC100 operates as compare register

(4) Signal edge selection register 10 (SESA10)

The SESA10 register specifies the valid edge of external interrupt requests from external pins (TICC10, TICC11, TCLR1).

The valid edge (rising edge, falling edge, or both edges) can be specified independently for each pin.

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

Reset input clears this register to 00H.

Cautions: 1. Changing the values of the SESA10 register bits during TMENC10 operation (TM1CE bit = 1) is prohibited.

2. Be sure to set (1) the TM1CE bit of timer control register 1 (TMC10) even when TMENC10 is not used and the TICC10 and TICC11 pins are used as external interrupts INTCC10 and INTCC11 respectively.

3. Before setting the trigger mode of the TICC10, TICC11, and TCLR1n pins, set the PM10 and PMC10 registers. If the PM10 and PMC10 registers are set after the SESA10 register has been set, an illegal interrupt, incorrect counting, and incorrect clearing may occur, depending on the timing of setting the PM10 and PMC10 registers.

Figure 12-10: Signal Edge Selection Register 10 (SESA10) (1/2)

After reset: 00H R/W Address: FFFFF6BDH

	7	6	5	4	3	2	1	0
SESA10	TESUD1	TESUD0	CESUD1	CESUD0	IES111	IES110	IES101	IES100
	TIUD, TCUD1		TCLR1		TICC11 capture trigger		TICC10 capture trigger	

TESUD1	TESUD0	Valid Edge Specification of TIUD1 and TCUD1 Pins
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

- The set values of the TESUD1 and TESUD0 bits are only valid in UDC mode A and UDC mode B.
- If mode 4 is specified as the operation mode of TMENC10 (specified with PRM102 to PRM100 bits of PRM10 register), the valid edge specifications for the TIUD1 and TCUD1 pins (TESUD1 and TESUD0 bits) are not valid.

CESUD1	CESUD0	Valid Edge and Level Specification of TCLR1 Pins
0	0	Falling edge (TMENC10 cleared after edge detection)
0	1	Rising edge (TMENC10 cleared after edge detection)
1	0	Low level (TMENC10 clear status held)
1	1	High level (TMENC10 clear status held)

The set values of the CESUD1 and CESUD0 bits are valid only in UDC mode A.

Figure 12-10: Signal Edge Selection Register 10 (SESA10) (2/2)

IES111	IES110	Valid Edge Specification of TICC11 Capture Trigger Input Pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges
A valid edge on the TICC11 pin triggers the capture register CC101. Simultaneously an interrupt (INTCC11) is generated.		

IES101	IES100	Valid Edge Specification of TICC10 Capture Trigger Input Pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges
A valid edge on the TICC10 pin triggers the capture register CC100. Simultaneously an interrupt (INTCC10) is generated.		

(5) Prescaler mode register 10 (PRM10)

The PRM register is used to perform the following selections.

- Selection of count clock in the general-purpose timer mode (CMD bit of TUM10 register = 0)
- Selection of count operation mode in the UDC mode (CMD bit = 1)

This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 07H.

Cautions: 1. Overwriting the PRM10 register during TMENC10 operation (TM1CE bit = 1) is prohibited.

2. When the CMD bit of the TUM10 register = 1 (UDC mode), setting the values of the PRM2 to PRM0 bits to 000B, 001B, 010B, and 011B is prohibited.

3. When TMENC10 is in mode 4, specification of the valid edge for the TIUD1 and TCUD1 pins is invalid.

Figure 12-11: Prescaler Mode Register 10 (PRM10)

After reset: 07H R/W Address: FFFFF6BEH

	7	6	5	4	3	2	1	0
PRM10	0	0	0	0	0	PRM102	PRM101	PRM100

PRM102	PRM101	PRM100	CMD = 0	CMD = 1	
			Count Clock	Count Clock	UDC Mode
0	0	0	Setting prohibited	Setting prohibited	
0	0	1	$f_{XX}/8$		
0	1	0	$f_{XX}/16$		
0	1	1	$f_{XX}/32$		
1	0	0	$f_{XX}/64$	TIUD1	Mode 1
1	0	1	$f_{XX}/128$		Mode 2
1	1	0	$f_{XX}/256$		Mode 3
1	1	1	$f_{XX}/512$		Mode 4

Remark: f_{XX}: Internal system clock

(a) In general-purpose timer mode (CMD bit = 0)

The count clock is fixed to the internal clock. The clock rate of TMENC10 is specified by the PRM102 to PRM100 bits.

(b) UDC mode (CMD bit = 1)

The TMENC10 count triggers in the UDC mode are as follows.

Operation Mode	TMENC10 Operation
Mode 1	Down count when TCUD1 = high level Up count when TCUD1 = low level
Mode 2	Up count upon detection of valid edge of TIUD1 input Down count upon detection of valid edge of TCUD1 input
Mode 3	Automatic judgment by TCUD1 input level upon detection of valid edge of TIUD1 input
Mode 4	Automatic judgment upon detection of both edges of TIUD1 input and both edges of TCUD1 input

(6) Status register 10 (STATUS10)

The STATUS10 register indicates the operating status of TMENC10.
This register is read-only in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Caution: Overwriting the STATUS10 register during TMENC10 operation (TM1CE bit = 1) is prohibited.

Figure 12-12: Status Register 10 (STATUS10)

After reset: 00H R/W Address: FFFFF6BFH

	7	6	5	4	3	2	1	0
STATUS10	0	0	0	0	0	TM1UDF	TM1OVF	TM1UBD

TM1UDF	TMENC10 Underflow Flag
0	No TMENC10 count underflow
1	TMENC10 count underflow
The TM1UDF bit is cleared (0) upon completion of read access to the STATUS10 register from the CPU.	

TM1OVF	TMENC10 Overflow Flag
0	No TMENC10 count overflow
1	TMENC10 count overflow
The TM1OVF bit is cleared (0) upon completion of read access to the STATUS10 register from the CPU.	

TM1UBD	TMENC10 Up/Down Counter Operation Status
0	TMENC10 up count in progress
1	TMENC10 down count in progress
The state of the TM1UBD bit differs according to the mode as follows. • The TM1UBD bit is fixed to 0 when the CMD bit of the TUM10 register = 0 (general-purpose timer mode). • The TM1UBD bit indicates the TMENC10 up/down count status when the CMD bit of the TUM register = 1 (UDC mode)	

12.5 Operation

12.5.1 Basic operation

The following two operation modes can be selected for TMENC10.

(1) General-purpose timer mode (CMD bit of TUM10 register = 0)

In the general-purpose timer mode, the TMENC10 operates either as a 16-bit interval timer or as a PWM output timer (count operation is up count only).
The count clock to TMENC10 is selected by prescaler mode register 10 (PRM10).

(2) Up/down counter mode (UDC mode) (CMD bit of TUM10 register = 1)

In the UDC mode, TMENC10 operates as a 16-bit up/down counter.
External clock input (TIUD1, TCUD1 pins) set by PRM10 register setting is used as the TMENC10 count clock.
The UDC mode is further divided into two modes according to the TMENC10 clear conditions.

- **UDC mode A (TUM10 register's CMD bit = 1, MSEL bit = 0)**

The TMENC10 clear source can be selected as external clear input (TCLR1), the internal signal indicating a match between the TMENC10 count value and the CM100 set value during an up count operation, or the logical sum (OR) of the two signals, using the CLR1 and CLR0 bits of the TMC10 register.
TMENC10 can transfer (reload) the value of CM100 upon occurrence of TMENC10 underflow, when the RLEN bit of the TMC10 register is set (1).

- **UDC mode B (TUM10 register's CMD bit = 1, MSEL bit = 1)**

The status of TMENC10 after a match of the TMENC10 count value and CM100 set value is as follows.

<1> In case of an up count operation, TMENC10 is cleared (0000H), and the INTCM10 interrupt is generated.
<2> In case of a down count operation, the TMENC10 count value is decremented (-1).

The status of TMENC10 after a match of the TMENC10 count value and CM101 set value is as follows.

<1> In case of an up count operation, the TMENC10 count value is incremented (+1).
<2> In case of a down count operation, TMENC10 is cleared (0000H), and the INTCM11 interrupt is generated.

12.5.2 Operation in general-purpose timer mode

TMENC10 can perform the following operations in the general-purpose timer mode.

(1) Interval operation

TMENC10 and CM100 always compare their values and the INTCM10 interrupt is generated upon occurrence of a match. TMENC10 is cleared (0000H) at the count clock following the match. Furthermore, when one more count clock is input, TMENC10 counts up to 0001H.
The interval time can be calculated by the following formula.

Interval time = (CM100 value + 1) × TMENC10 count clock rate

Caution: Interval operation can be achieved by setting the ENMD bit of the TMC register to 1.

(2) Free-running operation

TMENC10 performs full count operation from 0000H to FFFFH, and after the TM1OVF bit of the STATUS10 register is set (1), TMENC10 is cleared and resumes counting.
The free-running cycle can be calculated by the following formula.

Free-running cycle = 65,536 × TMENC10 count clock rate

Caution: The free-running operation can be achieved by setting the ENMD bit of the TMC register to 0.

(3) Compare function

TMENC10 connects two compare register (CM100, CM101) channels and two capture/compare register (CC100, CC101) channels.
When the TMENC10 count value and the set value of one of the compare registers match, a match interrupt (INTCM10, INTCM11, INTCC10**Note**, INTCC11**Note**) is output. Particularly in the case of an interval operation, TMENC10 is cleared upon generation of the INTCM10 interrupt.

Note: This match interrupt is generated when CC100 and CC101 are set to the compare register mode.

(4) Capture function

TMENC10 connects two capture/compare register (CC100, CC101) channels.
When CC100 and CC101 are set to the capture register mode, the value of TMENC10 is captured in synchronization with the corresponding capture trigger signal.
Furthermore, an interrupt request (INTCC10, INTCC11) is generated by the TICC10, TICC11 input signals.

Table 12-3: Capture Trigger Signal to 16-Bit Capture Register

Capture Register	Capture Trigger Signal
CC100	TICC10
CC101	TICC11

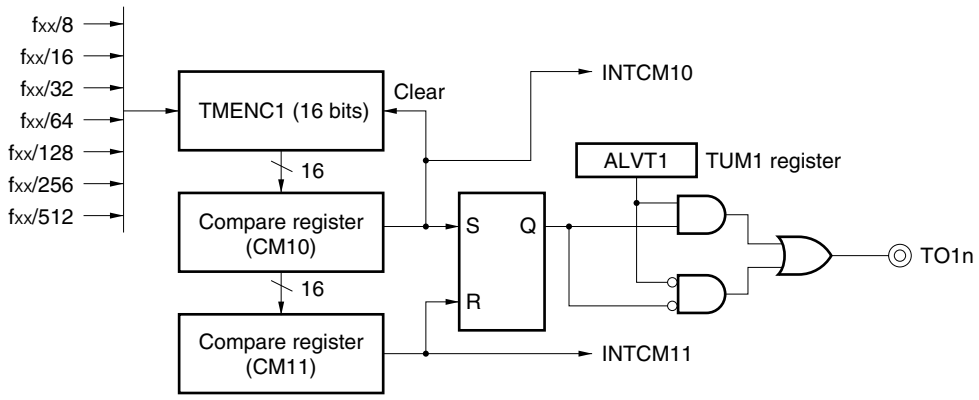
Remark: CC100 and CC101 are capture/compare registers. Which of these registers is used is specified with capture/compare control register 1 (CCR10).

The valid edge of the capture trigger is specified by signal edge selection register 10 (SESA10). If both the rising edge and the falling edge are selected as the capture triggers, it is possible to measure the input pulse width from external. If a single edge is selected as the capture trigger, the input pulse cycle can be measured.

(5) PWM output operation

PWM output operation is performed from the TO1 pin by setting TMENC10 to the general-purpose timer mode (CMD bit of the TUM10 register = 0).
The resolution is 16 bits, and the count clock can be selected from among seven internal clocks ($f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, $f_{XX}/512$).

Figure 12-13: TMENC10 Block Diagram (During PWM Output Operation)

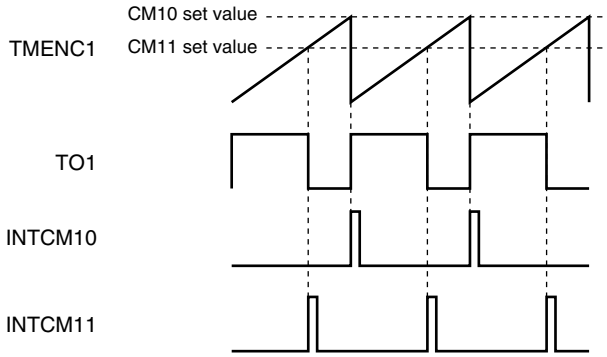


Remark: f_{XX}: Internal system clock

- **Description of operation**

The PWM output cycle is specified by using the compare register CM100. When the value of this register matches the value of TMENC10, the INTCM10 interrupt is generated, and TMENC10 is cleared at the next count clock after the match.
The required PWM output duty is set by using the compare register CM101.

Figure 12-14: PWM Signal Output Example (When ALVT10 Bit = 0 Is Set)



Cautions: 1. Changing the values of the CM100 and CM101 registers is prohibited during TMENC10 operation (TM1CE bit of TMC10 register = 1).

2. Changing the value of the ALVT1 bit of the TUM register is prohibited during TMENC10 operation.

3. PWM signal output is performed from the second PWM cycle after the TM1CE bit is set (1).

12.5.3 Operation in UDC mode

(1) Overview of operation in UDC mode

The count clock input to TMENC10 in the UDC mode (CMD bit of TUM10 register = 1) can only be externally input from the TIUD1 and TCUD1 pins. Up/down count judgment in the UDC mode is determined based on the phase difference of the TIUD1 and TCUD1 pin inputs according to the PRM10 register setting (there is a total of four choices).

Table 12-4: List of Count Operations in UDC Mode

PRM10 Register			Operation Mode	TM1n Operation
PRM102	PRM101	PRM100		
1	0	0	Mode 1	Down count when TCUD1 = high level Up count when TCUD1 = low level
1	0	1	Mode 2	Up count upon detection of valid edge of TIUD1 input Down count upon detection of valid edge of TCUD1 input
1	1	0	Mode 3	Automatic judgment in TCUD1 input level upon detection of valid edge of TIUD1 input
1	1	1	Mode 4	Automatic judgment upon detection of both edges of TIUD1 input and both edges of TCUD1 input

The UDC mode is further divided into two modes according to the TMENC10 clear conditions (count operation is performed only with TIUD1, TCUD1 input in both modes).

- **UDC mode A (TUM register's CMD bit = 1, MSEL bit = 0)**

  The TMENC10 clear source can be selected as only external clear input (TCLR1), a match signal between the TMENC10 count value and the CM100 set value during up count operation, or logical sum (OR) of the two signals, using bits CLR1 and CLR0 of the TMC10 register.
  TMENC10 can transfer (reload) the value of CM100 upon occurrence of TMENC10 underflow, when the RLEN bit of the TMC10 register is set (1).

- **UDC mode B (TUMn register's CMD bit = 1, MSEL bit = 1)**

  The status of TMENC10 after match of the TMENC10 count value and CM100 set value is as follows.
  <1> In case of an up count operation, TMENC10 is cleared (0000H), and the INTCM10 interrupt is generated.
  <2> In case of a down count operation, the TMENC10 count value is decremented (-1).

  The status of TMENC10 after match of the TMENC10 count value and CM101 set value is as follows.
  <1> In case of an up count operation, the TMENC10 count value is incremented (+1).
  <2> In case of a down count operation, TMENC10 is cleared (0000H), and the INTCM11 interrupt is generated.

(2) Up/down count operation in UDC mode

TMENC10 up/down count judgment in the UDC mode is determined based on the phase difference of the TIUD1 and TCUD1 pin inputs according to the PRM register setting.

(a) Mode 1 (PRM12 to PRM10 bits = 100B)

In mode 1, the following count operations are performed based on the level of the TCUD1 pin upon detection of the valid edge of the TIUD1 pin.

- TMENC10 down count operation when TCUD1 pin = high level
- TMENC10 up count operation when TCUD1 pin = low level

Figure 12-15: Mode 1 (When Rising Edge Is Specified as Valid Edge of TIUD1 Pin)

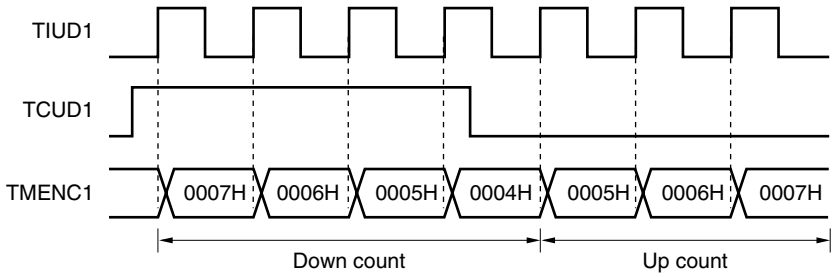
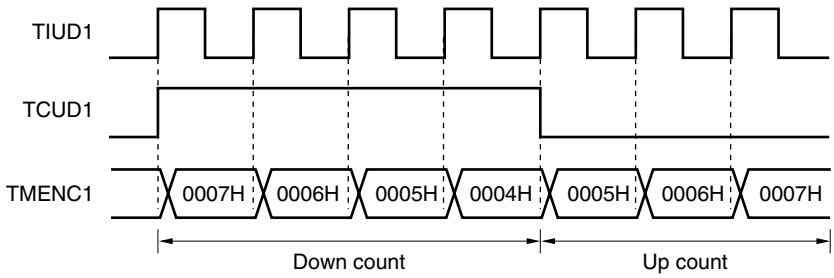


Figure 12-16: Mode 1 (When Rising Edge Is Specified as Valid Edge of TIUD1 Pin): In Case of Simultaneous TIUD1, TCUD1 Pin Edge Timing



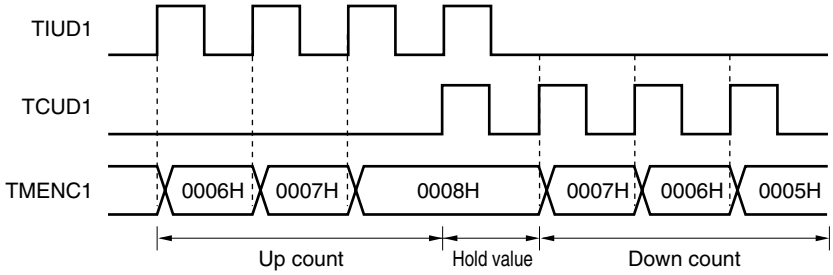
(b) Mode 2 (PRM102 to PRM100 bits = 101B)

The count conditions in mode 2 are as follows.

- TMENC10 up count upon detection of valid edge of TIUD1 pin
- TMENC10 down count upon detection of valid edge of TCUD1 pin

Caution: If the count clock is simultaneously input to the TIUD1 pin and the TCUD1 pin, count operation is not performed and the immediately preceding value is held.

Figure 12-17: Mode 2 (When Rising Edge Is Specified as Valid Edge of TIUD1, TCUD1 Pins)



(c) Mode 3 (PRM102 to PRM100 bits = 110B)

In mode 3, when two signals 90 degrees out of phase are input to the TIUD1 and TCUD1 pins, the level of the TCUD1 pin is sampled at the timing of the valid edge of the TIUD1 pin (refer to **Figure 12-18**).
If the TCUD1 pin level sampled at the valid edge timing of the TIUD1 pin is low, TMENC10 counts down.
If the TCUD1 pin level sampled at the valid edge timing of the TIUD1 pin is high, TMENC10 counts up.

Figure 12-18: Mode 3 (When Rising Edge Is Specified as Valid Edge of TIUD1 Pin)

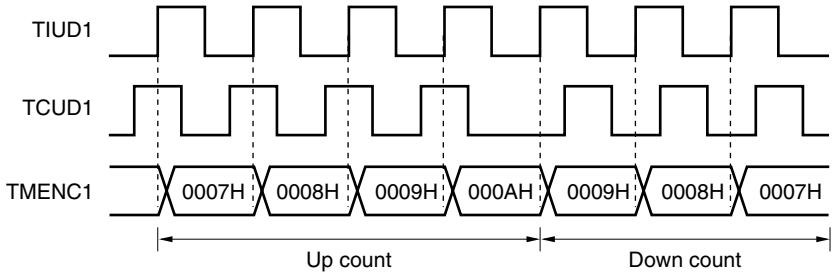
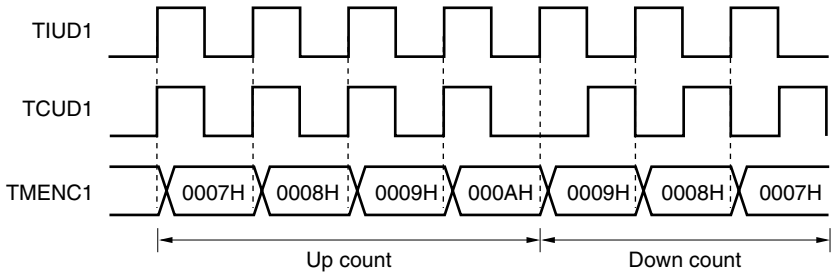


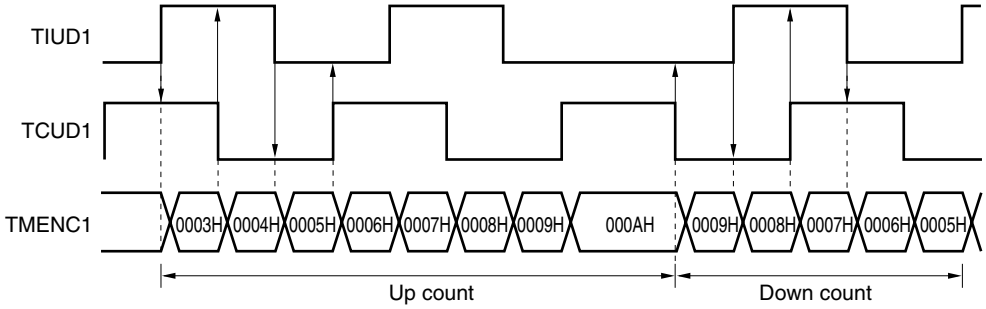
Figure 12-19: Mode 3 (When Rising Edge Is Specified as Valid Edge of TIUD1 Pin): In Case of Simultaneous TIUD1, TCUD1 Pin Edge Timing



(d) Mode 4 (PRM102 to PRM100 bits = 111B)

In mode 4, when two signals out of phase are input to the TIUD1 and TCUD1 pins, up/down operation is automatically judged and counting is performed according to the timing shown in **Figure 12-20**.
In mode 4, counting is executed at both the rising and falling edges of the two signals input to the TIUD1 and TCUD1 pins. Therefore, TMENC10 counts four times per cycle of an input signal (× 4 count).

Figure 12-20: Mode 4



Cautions: 1. When mode 4 is specified as the operation mode of TMENC10, the valid edge specifications for pins TIUD1 and TCUD1 are not valid.

2. If the TIUD1 pin edge and TCUD1 pin edge are input simultaneously in mode 4, TMENC10 continues the same count operation (up or down) it was performing immediately before the input.

(3) Operation in UDC mode A

(a) Interval operation

The operations at the count clock following a match of the TMENC10 count value and the CM100 set value are as follows.

- In case of up count operation: TMENC10 is cleared (0000H) and the INTCM10 interrupt is generated.
- In case of down count operation: The TMENC10 count value is decremented (-1) and the INTCM10 interrupt is generated.

Remark: The interval operation can be combined with the transfer operation.

(b) Transfer operation

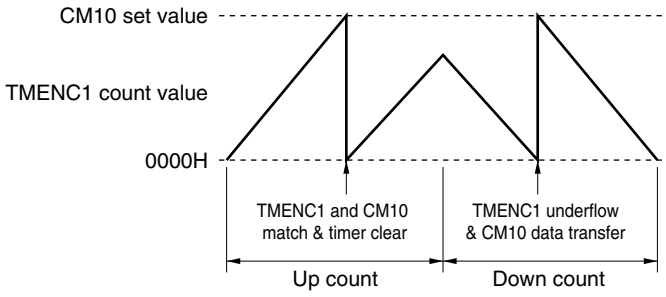
The operations at the next count clock after the count value of TMENC10 becomes 0000H during TMENC10 count down operation are as follows.

- In case of down count operation: The data held in CM100 is transferred.
- In case of up count operation: The TMENC10 count value is incremented (+1).

Remarks: 1. Transfer enable/disable can be set with the RLEN bit of the TMC10 register.

2. The transfer operation can be combined with the interval operation.

Figure 12-21: Example of TMENC10 Operation When Interval Operation and Transfer Operation are Combined



(c) Compare function

TM1n connects two compare register (CM100, CM101) channels and two capture/compare register (CC100, CC101) channels.
When the TMENC10 count value and the set value of one of the compare registers match, a match interrupt (INTCM10, INTCM11, INTCC10**Note**, INTCC11**Note**) is output.

Note: This match interrupt is generated when CC100 and CC101 are set to the compare register mode.

(d) Capture function

TMENC10 connects two capture/compare register (CC100, CC101) channels.
When CC100 and CC101 are set to the capture register mode, the value of TMENC10 is captured in synchronization with the corresponding capture trigger signal.
When the TMENC10 is set to the capture register mode, a capture interrupt (INTCC10, INTCC11) is generated upon detection of the valid edge.

(4) Operation in UDC mode B

(a) Basic operation

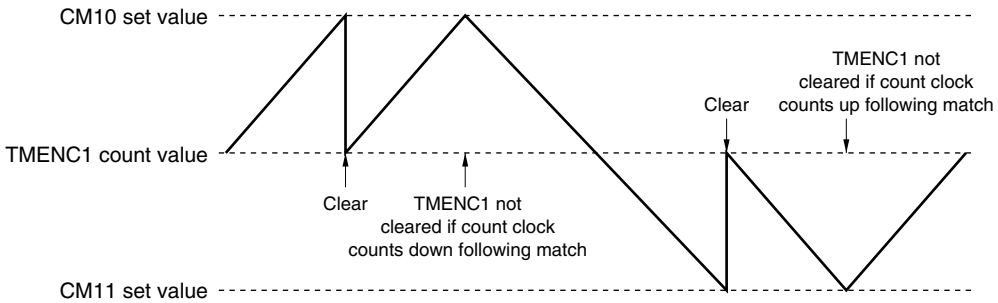
The operations at the next count clock after the count value of TMENC10 and the CM100 set value match when TMENC10 is in UDC mode B are as follows.

- In case of up count operation: TMENC10 is cleared (0000H) and the INTCM10 interrupt is generated.
- In case of down count operation: The TMENC10 count value is decremented (-1).

The operations at the next count clock after the count value of TMENC10 and the CM101 set value match when TMENC10 is in UDC mode B are as follows.

- In case of up count operation: The TMENC10 count value is incremented (+1).
- In case of down count operation: TMENC10 is cleared (0000H) and the INTCM11 interrupt is generated.

Figure 12-22: Example of TM1Operation in UDC Mode



(b) Compare function

TMENC10 connects two compare register (CM100, CM101) channels and two capture/compare register (CC100, CC101) channels.
When the TMENC10 count value and the set value of one of the compare registers match, a match interrupt (INTCM10 (only during up count operation), INTCM11 (only during down count operation), INTCC10[Note], INTCC11[Note]) is output.

Note: This match interrupt is generated when CC100 and CC101 are set to the compare register mode.

(c) Capture function

TMENC10 connects two capture/compare register (CC100, CC101) channels.
When CC100 and CC101 are set to the capture register mode, the value of TMENC10 is captured in synchronization with the corresponding capture trigger signal.
When the TMENC10 is set to the capture register mode, a capture interrupt (INTCC10, INTCC11) is generated upon detection of the valid edge.

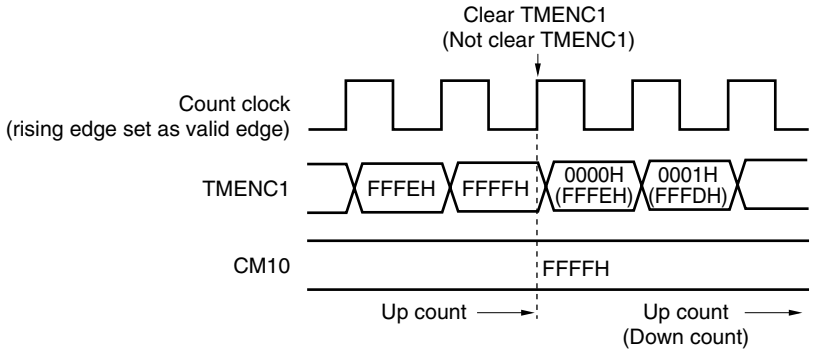
12.6 Supplementary Description of Internal Operation

12.6.1 Clearing of count value in UDC mode B

When TMENC10 is in UDC mode B, the count value clear operation is as follows.

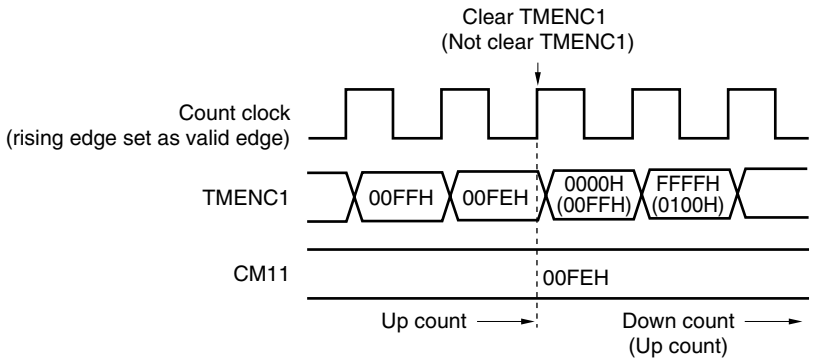
- In case of TMENC10 up count operation: TMENC10 is cleared upon match with CM100
- In case of TMENC10 down count operation: TMENC10 is cleared upon match with CM101

Figure 12-23: Clear Operation upon Match with CM100 During TMENC10 Up Count Operation



Remark: Items between parentheses in the above figure apply to down count operation.

Figure 12-24: Clear Operation upon Match with CM101 during TMENC10 Down Count Operation

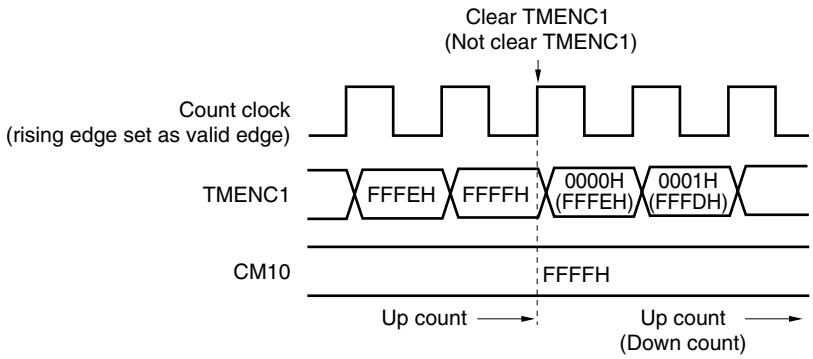


Remark: Items between parentheses in the above figure apply to up count operation.

12.6.2 Clearing of count value upon occurrence of compare match

The internal operation during TMENC10 clear operation upon occurrence of a compare match is as follows.

Figure 12-25: Count Value Clear Operation upon Compare Match



Caution: The operations at the next count clock after the count value of TMENC10 and the CM100 set value match are as follows.

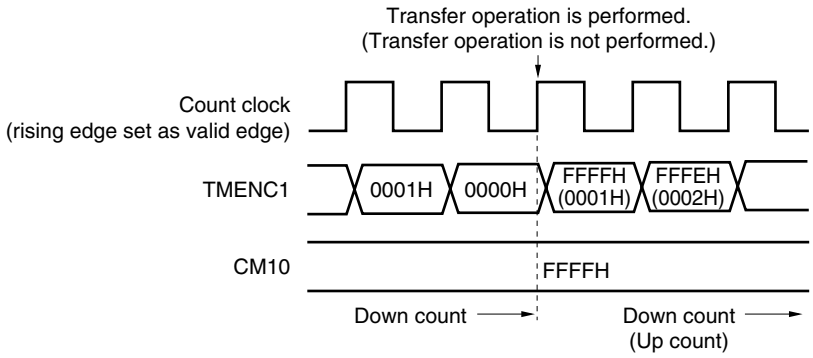
- **In case of up count: Clear operation is performed.**
- **In case of down count: Clear operation is not performed.**

Remark: Items between parentheses in the above figure apply to down count operation.

12.6.3 Transfer operation

The internal operation during TMENC10 transfer operation is as follows.

Figure 12-26: Internal Operation During Transfer Operation



Caution: The count operations after the TMENC10 count value becomes 0000H are as follows.

- **In case of down count: Transfer operation is performed.**
- **In case of up count: Transfer operation is not performed.**

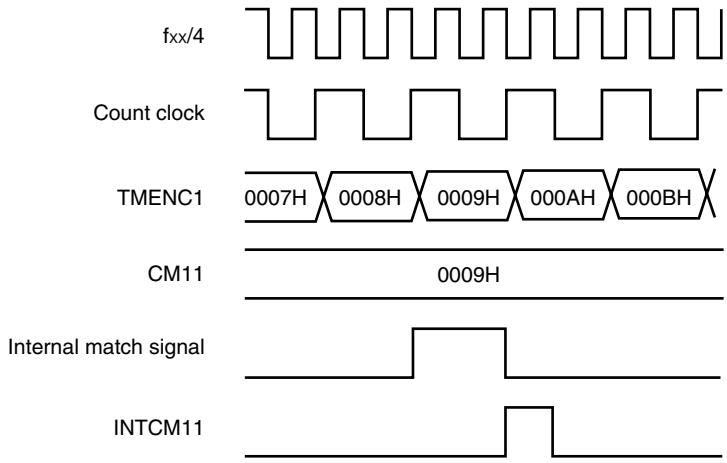
Remark: Items between parentheses in the above figure apply to up count operation.

12.6.4 Interrupt signal output upon compare match

An interrupt signal is output when the count value of TMENC10 matches the set value of the CM100, CM101, CC10[Note], or CC11[Note] register. The interrupt generation timing is as follows.

Note: When CC100 and CC101 are set to the compare register mode.

Figure 12-27: Interrupt Output upon Compare Match (CM101 with Operation Mode set to General-Purpose Timer Mode and Count Clock Set to $f_{XX}/8$)



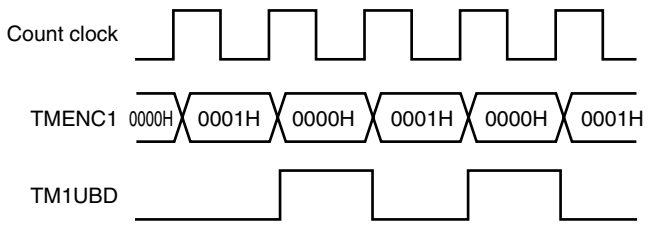
Remark: f_{CLK}: Base clock

An interrupt signal such as illustrated in Figure 12-27 is output at the next count following match of the TMENC10 count value and the set value of a corresponding compare register.

12.6.5 TM1UBD flag (bit 0 of STATUS register) operation

In the UDC mode (CMD bit of TUM register = 1), the TM1UBD flag changes as follows during TMENC10 up/down count operation at every internal operation clock.

Figure 12-28: TM1UBDn Flag Operation



Chapter 13 Auxiliary Frequency Output Function (AFO)

13.1 Features

- Frequency up to 8 Mbps
- Programmable frequency output
- Interval timer function
- Interrupt request signal (INTBRG2)

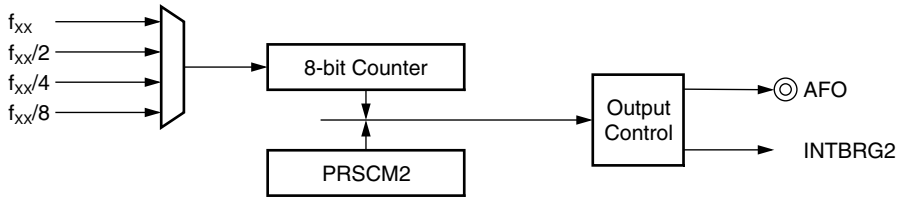
13.2 Configuration

The AFO function includes the following hardware.

Table 13-1: AFO Configuration

Item	Configuration
Control registers	Prescaler mode registers 2 (PRSM2)
	Prescaler compare registers 2 (PRSCM2)

Figure 13-1: Block Diagram of Auxiliary Frequency Output Function



13.3 Control Registers

(1) Prescaler mode register 2 (PRSM2)

The PRSM2 register controls generation of a baud rate signal for the AFO function.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Figure 13-2: Prescaler Mode Register 2 (PRSM2)

After reset: 00H R/W Address: FFFFFDE0H

	7	6	5	4	3	2	1	0
PRSM2		0	0	BGCE2	0	0	BGCS21	BGCS20

BGCE2	Baud Rate Generator Output Control
0	Disabled
1	Enabled

BGCS21	BGCS20	Baud Rate Generator Clock Selection (f_{BGCS2})	Setting Value (k)
0	0	f_{XX}	0
0	1	$f_{XX}/2$	1
1	0	$f_{XX}/4$	2
1	1	$f_{XX}/8$	3

Cautions: 1. Do not rewrite the PRSM2 register during operation.

2. Set the BGCS21, BGCS20 bits before setting the BGCE2 bit to 1.

(2) Prescaler compare registers 2 (PRSCM2)

The PRSCM2 register is an 8-bit compare register.
This register can be read or written in 8-bit units.
Reset input clears this register to 00H.

Figure 13-3: Prescaler Compare Register 2 (PRSCM2)

After reset: 00H R/W Address: FFFFFDE1H

	7	6	5	4	3	2	1	0
PRSCM2	PRSCM27	PRSCM26	PRSCM25	PRSCM24	PRSCM23	PRSCM22	PRSCM21	PRSCM20

PRSCM27	PRSCM26	PRSCM25	PRSCM24	PRSCM23	PRSCM22	PRSCM21	PRSCM20	AFO Clock	N
0	0	0	0	0	0	0	0	$f_{BGSC2}/256$	256
0	0	0	0	0	0	0	1	f_{BGSC2}	1
0	0	0	0	0	0	1	0	$f_{BGSC2}/2$	2
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	$f_{BGSC2}/252$	252
1	1	1	1	1	1	0	1	$f_{BGSC2}/253$	253
1	1	1	1	1	1	1	0	$f_{BGSC2}/254$	254
1	1	1	1	1	1	1	1	$f_{BGSC2}/255$	255

Cautions: 1. Do not rewrite the PRSCM2 register during operation.

2. Set the PRSCM2 register before setting the BGCE2 bit of the PRSM2 register to 1.

3. Do not set the AFO clock to a higher frequency than 8 MHz.

Remark: f_{BGCS2}: Clock frequency selected by the BGCS21, BGCS20 bits of the PRSM2 register.

13.4 Operation

13.4.1 Auxiliary frequency output

The auxiliary frequency output (AFO) is enabled as soon as the shared port (P75) is set into control output mode by setting bit 5 of the PM7 register to 0 and bit 5 of the PMC7 register to 1.

13.4.2 Auxiliary frequency generation

The auxiliary frequency output clock is generated by dividing the main clock. The baud rate generated from the main clock is obtained by the following equation.

$$f_{AFO} = \frac{f_{BGCS2}}{N \times 2} = \frac{f_{XX}}{2^k \times N \times 2}$$

Remarks: **1.** f_{AFO}: AFO clock

2. f_{BGCS2}: Clock frequency selected by the BGCS21, BGCS20 bits of the PRSM2 register.

3. f_{XX}: Main clock oscillation frequency

4. k: PRSM2 register setting value ($2 \leq k \leq 5$)

5. N: PRSCMm register setting value (1 to 255), when PRSCM2 = 01H to FFH, or N = 256, when PRSCM2 = 00H.

13.4.3 Interval timer function

The AFO function can be used as interval timer regardless whether the auxiliary frequency output is used or not. For this purpose an interrupt request signal (INTBRG2) is assigned, which can be handled like any maskable interrupt.

Chapter 14 A/D Converter

14.1 Features

- Analog input: 2 × 10 channels (ANI00 to ANI09, ANI10 to ANI19)
- 10-bit resolution
- On-chip A/D conversion result register (ADCRn0 to ADCRn9): 10 bits × 10
- A/D conversion trigger mode
  - A/D trigger mode
  - Timer trigger mode
  - External trigger mode
- Successive approximation method
- DMA transfer support of A/D conversion result to internal RAM

Remark: n = 0, 1

14.2 Configuration

The A/D converter of the V850E/PH2 adopts the successive approximation method, and uses A/D converter n mode registers 0, 1, 2 (ADMn0, ADMn1, ADMn2), and the A/D conversion result register (ADCRn0 to ADCRn9) to perform A/D conversion operations (n = 0, 1).

(1) Input circuit

The input circuit selects the analog input (ANIn0 to ANIn9) according to the mode set by the ADMn0, ADMn1, and ADMn2 registers.

(2) C-Array

Holds the charge of the differential voltage between the voltage input from the analog input pins (ANIn0 to ANIn9) and the reference voltage (1/2 AV_{DD}), and redistributes the sampled charges.

(3) C-Dummy

This block holds the reference voltage (1/2 AV_{DD}) and assigns the reference of the comparator input.

(4) Voltage comparator

The voltage comparator compares the C-Array comparison potential with the C-Dummy reference potential.

(5) A/D conversion result register (ADCRnm), A/D conversion result register nH (ADCRnmH) (n = 0, 1)(m = 0 to 9)

ADCRnm is a 10-bit register that holds A/D conversion results. Each time A/D conversion is completed, the conversion results are loaded from the successive approximation register (SAR). RESET input makes this register undefined.

(6) A/D conversion result register for DMA transfer (ADDMAn) (n = 0, 1)

ADDMAn is a 16-bit register that holds the last 10-bit A/D conversion result and an over rung flag for indicating a DMA transfer failure.

(7) ANIn0 to ANIn9 pins (n = 0, 1)

These are 10-channel analog input pins for the A/D converter n. They input the analog signals to be A/D converted.

Caution: Make sure that the voltages input to ANIn0 to ANIn9 do not exceed the rated values. If a voltage higher than AV_{DD} or lower than AV_{SSn} (even within the range of the absolute maximum ratings) is input to a channel, the conversion value of the channel is undefined, and the conversion values of the other channels may also be affected.

(8) AV_{REFn} pins (n = 0, 1)

This is the pin for inputting the reference voltage of the A/D converter. It converts signals input to the ANIn0 to ANIn9 pins to digital signals based on the voltage applied between AV_{SSn} and AV_{REFn}.

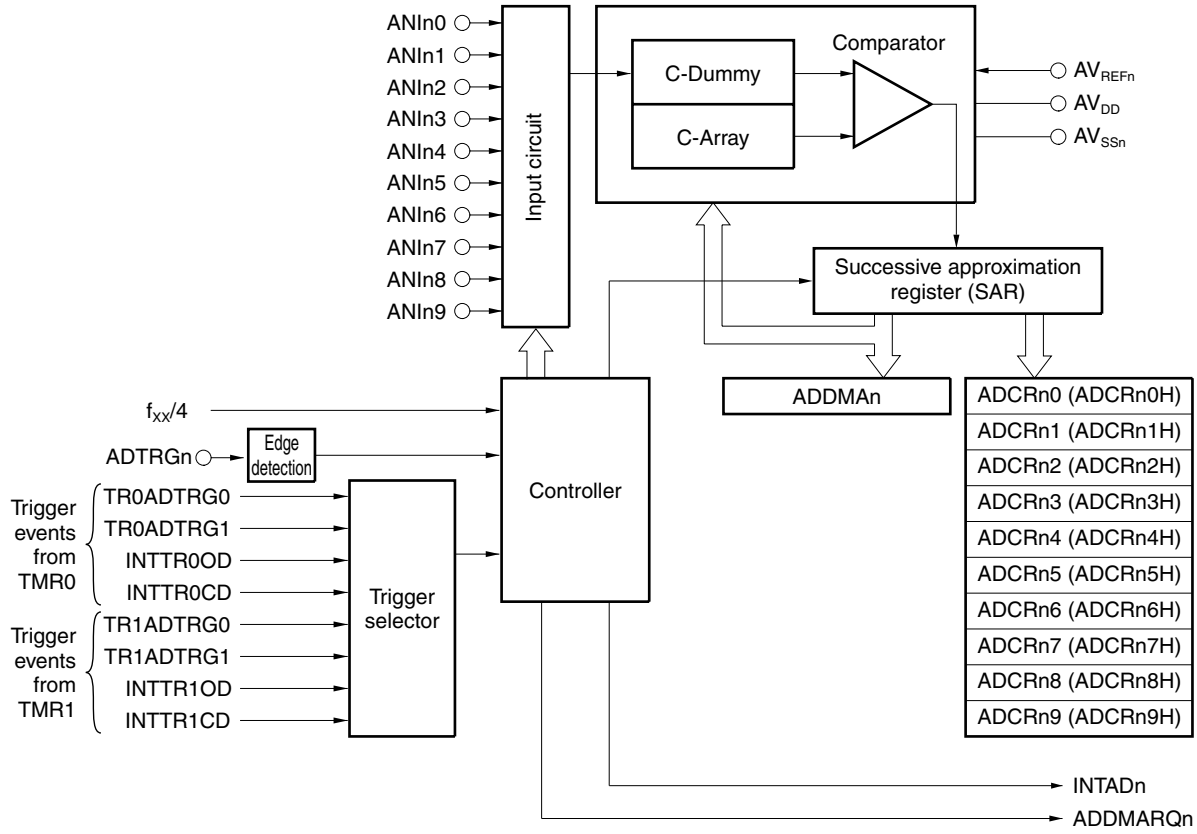
(9) AV_{SSn} pin (n = 0, 1)

This is the ground pin of the A/D converter. Always use this pin at the same potential as that of the EV_{SS} pin even when the A/D converter is not used.

(10) AV_{DD} pin

This is the analog power supply pin of both A/D converters (ADC0, ADC1).

Figure 14-1: Block Diagram of A/D Converter (ADCn)



Remarks: 1. f_{XX}: Main clock

2. n = 0, 1

Cautions: 1. If there is noise at the analog input pins (ANIn0 to ANIn9) or at the reference voltage input pin (AV_{REFn}), that noise may generate an illegal conversion result. Software processing will be needed to avoid a negative effect on the system from this illegal conversion result.
An example of this software processing is shown below.

- **Take the average result of a number of A/D conversions and use that as the A/D conversion result.**
- **Execute a number of A/D conversions consecutively and use those results, omitting any exceptional results that may have been obtained.**

2. Do not apply a voltage outside the AV_{SSn} to AV_{REFn} range to the pins that are used as A/D converter input pins.

14.3 Control Registers

(1) A/D converter n mode register 0 (ADMn0)

The ADMn0 register is an 8-bit register that specifies the operation mode, and executes conversion operations.

This register can be read or written in 8-bit or 1-bit units. However, bit 6 can only be read. Writing this bit is ignored.

Reset input sets this register to 00H.

Cautions: 1. When the ADCEn bit is 1 in the timer trigger mode and external trigger mode, the trigger signal standby state is set. To clear the ADCEn bit, write 0 or reset.
In the A/D trigger mode, the conversion trigger is set by writing 1 to the ADCEn bit. After the operation, when the mode is changed to the timer trigger mode or external trigger mode without clearing the ADCEn bit, the trigger input standby state is set immediately after changing the register.

2. Changing the setting of the BSn and MSn bits is prohibited while A/D conversion is enabled (ADCEn bit = 1).

3. When data is written to the ADMn0 register during an A/D conversion operation, the conversion operation is initialized and conversion is executed from the beginning.

Figure 14-2: A/D Converter n Mode Register 0 (ADMn0)

After reset: 00H R/W Address: ADM00 FFFFF200H, ADM10 FFFFF240H

	7	6	5	4	3	2	1	0
ADMn0 (n = 0, 1)	ADCEn	ADCSn	BSn	MSn	0	0	0	0

ADCEn	A/D Conversion Operation Control of ADCn
0	Disables A/D conversion operation of ADCn
1	Enables A/D conversion operation ADCn

ADCSn	A/D Conversion Status Flag of ADCn
0	A/D conversion of ADCn is stopped
1	A/D conversion of ADCn is operating

BSn	ADCn Buffer Mode Specification
0	1-buffer mode
1	4-buffer mode

MSn	ADCn Operation Mode Specification
0	Scan mode
1	Select mode

Remark: n = 0, 1

(2) A/D converter n mode register 1 (ADMn1)

The ADMn1 register is an 8-bit register that specifies the conversion operation time and trigger mode.

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

Reset input sets this register to 00H.

Cautions: 1. Changing the setting of the EGAn1, EGAn0, and FRn3 to FRn0 bits is prohibited while A/D conversion is enabled (ADCEn bit of the ADMn0 register = 1).

2. When data is written to the ADMn1 register during an A/D conversion operation, the conversion operation is initialized and conversion is executed from the beginning.

Figure 14-3: A/D Converter n Mode Register 1 (ADMn1) (1/2)

After reset: 00H R/W Address: ADM01 FFFFF201H, ADM11 FFFFF241H

	7	6	5	4	3	2	1	0
ADMn1	EGAn1	EGAn0	TRGn1	TRGn0	FRn3	FRn2	FRn1	FRn0

(n = 0, 1)

EGAn1	EGAn0	Valid Edge Specification of External Trigger Input (ADTRGn)
0	0	No edge detected (does not operate as external trigger)
0	1	Falling edge detected
1	0	Rising edge detected
1	1	Both edges, falling and rising edge detected

TRGn1	TRGn0	ADCn Trigger Mode Specification
0	0	A/D trigger mode
0	1	Timer trigger mode
1	0	External trigger mode
1	1	Setting prohibited

Remark: n = 0, 1

Figure 14-3: A/D Converter n Mode Register 1 (ADMn1) (2/2)

FRn3	FRn2	FRn1	FRn0	Number of conversion clocks	Conversion Operation Time[Note 1]	
					f_{XX} = 64 MHz	A/D Stabilization Time[Note 2]
0	0	0	0	128	2.0 μs	$64/f_{XX}$
0	0	0	1	256	4.0 μs	$128/f_{XX}$
0	0	1	0	384	6.0 μs	$160/f_{XX}$
0	0	1	1	512	8.0 μs	$160/f_{XX}$
0	1	0	0	640	Setting prohibited	$160/f_{XX}$
0	1	0	1	768		$160/f_{XX}$
Others than above				Setting prohibited		

Notes: 1. Set the conversion operation time in the range of 2 to 10 μs.

2. After the ADCEn bit is set from 0 to 1 to secure the stabilization time of the A/D converter, conversion is started after the A/D stabilization time has elapsed only before the first A/D conversion is executed.

Cautions: 1. Do not change the set value of the A/D conversion time (FRn3 to FRn0 bits) during an A/D conversion operation (ADCEn bit = 1). To change the value, clear the ADCEn bit to 0.

2. When the trigger mode (TRGn1 and TGRn0 bits) is changed midway, A/D conversion can be started immediately without having to secure the A/D stabilization time by re-setting the ADCE bit to 1.

Remarks: 1. f_{XX}: Main clock

2. n = 0, 1

(3) A/D converter n mode register 2 (ADMn2)

The ADMn2 register is an 8-bit register that specifies the analog input pin of the A/D converter n (n = 0, 1).

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

Reset input sets this register to 00H.

Cautions: 1. If a channel for which no analog input pin exists is specified, the result of A/D conversion is undefined.

2. Changing the setting of the ANISn3 to ANISn0 bits is prohibited while A/D conversion is enabled (ADCEn bit of the ADMn0 register = 1).

3. When data is written to the ADMn2 register during an A/D conversion operation, the conversion operation is initialized and conversion is executed from the beginning.

Figure 14-4: A/D Converter n Mode Register 2 (ADMn2)

After reset: 00H R/W Address: ADM01 FFFFF201H, ADM11 FFFFF241H

	7	6	5	4	3	2	1	0
ADMn2 (n = 0, 1)	0	0	0	0	ANISn3	ANISn2	ANISn1	ANIn0

ANIn3	ANIn2	ANIn2	ANIn0	Specification of Analog Input Pins for A/D Conversion	
				Select Mode	Scan Mode
0	0	0	0	ANIn0	ANIn0
0	0	0	1	ANIn1	ANIn0, ANIn1
0	0	1	0	ANIn2	ANIn0 to ANIn2
0	0	1	1	ANIn3	ANIn0 to ANIn3
0	1	0	0	ANIn4	ANIn0 to ANIn4
0	1	0	1	ANIn5	ANIn0 to ANIn5
0	1	1	0	ANIn6	ANIn0 to ANIn6
0	1	1	1	ANIn7	ANIn0 to ANIn7
1	0	0	0	ANIn8	ANIn0 to ANIn8
1	0	0	1	ANIn9	ANIn0 to ANIn9
Others than above				Setting prohibited	

Remark: n = 0, 1

(4) A/D converter n trigger source select register (ADTRSELn)

The ADTRSELn register is an 8-bit register that specifies the timer trigger signal in the timer trigger mode (TRGn1, TRGn0 bits of ADMn1 register = 01B).
This register can be read or written in 8-bit units.
Reset input sets this register to 00H.

Caution: Before changing the setting of the ADTRSELn register, stop the A/D conversion operation (by clearing the ADCEn bit of the ADMn0 register to 0). The operation is not guaranteed if the setting of the ADTRSELn register is changed while A/D conversion is enabled (ADCEn bit = 1).

Figure 14-5: A/D Converter n Trigger Source Select Register (ADTRSELn)

After reset: 00H R/W Address: ADTRSEL0 FFFFF270H, ADTRSEL1 FFFFF272H

	7	6	5	4	3	2	1	0
ADTRSELn (n = 0, 1)	0	0	0	0	TSELn3	TSELn2	TSELn1	TSELn0

TSELn3	TSELn2	TSELn2	TSELn0	Trigger Source Selection in Timer Trigger Mode
0	0	0	0	None. All trigger sources are ignored.
0	0	0	1	TR0ADTRG0 signal (from TMR0)
0	0	1	0	TR0ADTRG1 signal (from TMR0)
0	0	1	1	TR1ADTRG0 signal (from TMR1)
0	1	0	0	TR1ADTRG1 signal (from TMR1)
0	1	0	1	INTTR0OD interrupt (from TMR0)
0	1	1	0	INTTR0CD interrupt (from TMR0)
0	1	1	1	INTTR1OD interrupt (from TMR1)
1	0	0	0	INTTR1CD interrupt (from TMR1)
Others than above				Setting prohibited

Remark: n = 0, 1

(5) A/D conversion result registers n0 to n9, n0H to n9H (ADCRn0 to ADCRn9, ADCRn0H to ADCRn9H)

The ADCRnm register is a 10-bit register holding the A/D conversion results (n = 0, 1)(m = 0 to 9). These registers are read-only in 16-bit or 8-bit units. When 16-bit access is performed, the ADCRnm register is specified, and when 8 bit access is performed, the ADCRnmH register holding the higher 8 bits of the conversion result is specified.

When reading the 10-bit data of the A/D conversion results from the ADCRnm register, only the higher 10 bits are valid and the lower 6 bits are always read as 0.

Reset input causes an undefined register content.

Figure 14-6: A/D Conversion Result Registers n0 to n9, n0H to n9H (ADCRn0 to ADCRn9, ADCRn0H to ADCRn9H)

After reset: Undefined R Address: ADCR00 FFFFF210H, ADCR10 FFFFF250H,
ADCR01 FFFFF212H, ADCR11 FFFFF252H,
ADCR02 FFFFF214H, ADCR12 FFFFF254H,
ADCR03 FFFFF216H, ADCR13 FFFFF256H,
ADCR04 FFFFF218H, ADCR14 FFFFF258H,
ADCR05 FFFFF21AH, ADCR15 FFFFF25AH,
ADCR06 FFFFF21CH, ADCR16 FFFFF25CH,
ADCR07 FFFFF21EH, ADCR17 FFFFF25EH,
ADCR08 FFFFF220H, ADCR18 FFFFF260H,
ADCR09 FFFFF222H, ADCR19 FFFFF262H,

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCRnm	ADnm9	ADnm8	ADnm7	ADnm6	ADnm5	ADnm4	ADnm3	ADnm2	ADnm1	ADnm0	0	0	0	0	0	0

ADCRnmH

After reset: Undefined R Address: ADCR00H FFFFF210H, ADCR10H FFFFF250H,
ADCR01H FFFFF212H, ADCR11H FFFFF252H,
ADCR02H FFFFF214H, ADCR12H FFFFF254H,
ADCR03H FFFFF216H, ADCR13H FFFFF256H,
ADCR04H FFFFF218H, ADCR14H FFFFF258H,
ADCR05H FFFFF21AH, ADCR15H FFFFF25AH,
ADCR06H FFFFF21CH, ADCR16H FFFFF25CH,
ADCR07H FFFFF21EH, ADCR17H FFFFF25EH,
ADCR08H FFFFF220H, ADCR18H FFFFF260H,
ADCR09H FFFFF222H, ADCR19H FFFFF262H,

	7	6	5	4	3	2	1	0
ADCRnmH	ADnm9	ADnm8	ADnm7	ADnm6	ADnm5	ADnm4	ADnm3	ADnm2

Remark: n = 0, 1
m = 0 to 9

The correspondence between each analog input pin and the ADCRnm register is shown in Table 14-1 below.

Table 14-1: Assignment of A/D Conversion Result Registers to Analog Input Pins

Analog Input Pin	Assignment of A/D Conversion Result Registers	
	Select 1 Buffer Mode/ Scan Mode	Select 4 Buffer Mode
ANIn0	ADCRn0, ADCRn0H	ADCRn0 to ADCRn3, ADCRn0H to ADCRn3H
ANIn1	ADCRn1, ADCRn1H	
ANIn2	ADCRn2, ADCRn2H	
ANIn3	ADCRn3, ADCRn3H	
ANIn4	ADCRn4, ADCRn4H	ADCRn4 to ADCRn7, ADCRn4H to ADCRn7H
ANIn5	ADCRn5, ADCRn5H	
ANIn6	ADCRn6, ADCRn6H	
ANIn7	ADCRn7, ADCRn7H	
ANIn8	ADCRn8, ADCRn8H	ADCRn8 to ADCRn9, ADCRn8H to ADCRn9H
ANIn9	ADCRn9, ADCRn9H	

The relationship between the analog voltage input to the analog input pins (ANIn0 to ANIn9) and the A/D conversion result (of the A/D conversion result register (ADCRnm)) is as follows:

$$ADCR = INT\left(\frac{V_{IN}}{AV_{REF}} \times 1024 + 0{,}5\right)$$

or,

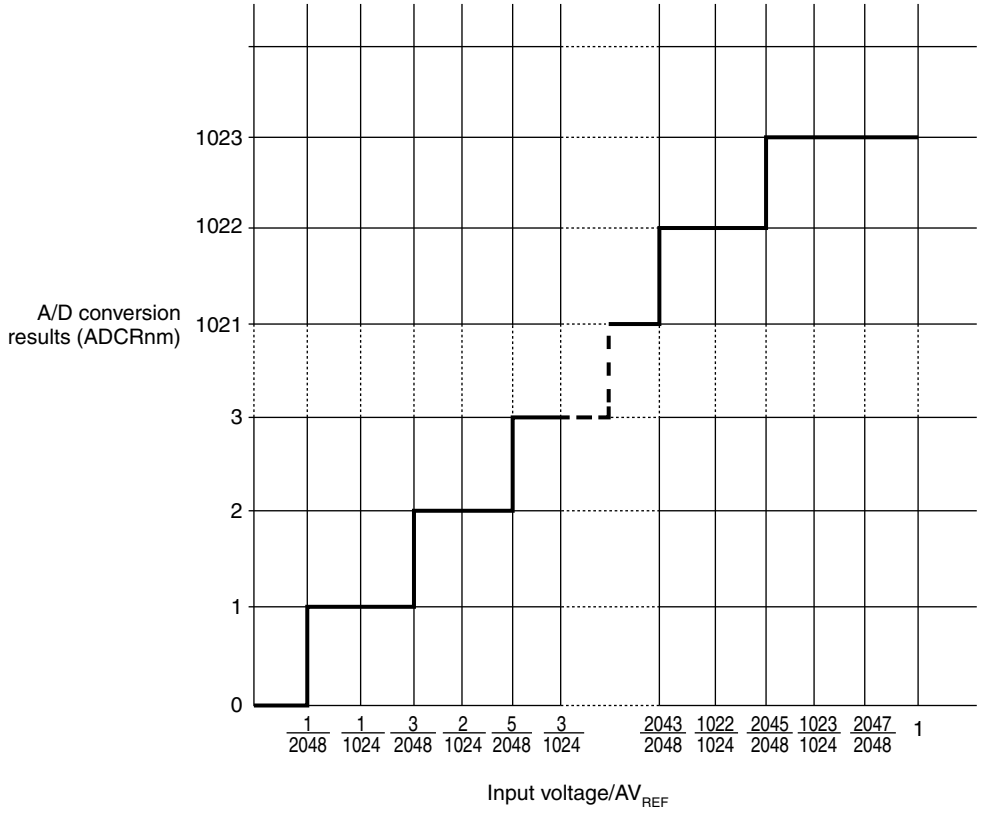
$$(ADCR - 0{,}5) \times \frac{AV_{REF}}{1024} \leq V_{IN} < (ADCR + 0{,}5) \times \frac{AV_{REF}}{1024}$$

INT( ): Function that returns the integer value
V_{IN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR: Value of A/D conversion result register (ADCRnm)

Figure 14-7 shows the relationship between the analog input voltage and the A/D conversion results.

Remark: n = 0, 1
m = 0 to 9

Figure 14-7: Relationship Between Analog Input Voltage and A/D Conversion Results



Remark: n = 0, 1
m = 0 to 9

(6) A/D conversion result register n for DMA (ADDMAn)

The ADDMAn register is a 16-bit register holding the result of the latest A/D conversion operation, and is used for DMA transfer of ADCn results into the internal RAM. It has an overrun detection flag indicating an overrun situation of the DMA transfer mechanism (n = 0, 1).
This register is read-only in 16-bit units.
Reset input causes an undefined register content.

Caution: Do not read the ADDMAn register by CPU during DMA transfer activities. If this register is read by CPU, overflow detection cannot be ensured.

Figure 14-8: A/D Conversion Result Registers n0 to n9, n0H to n9H (ADCRn0 to ADCRn9, ADCRn0H to ADCRn9H)

After reset: Undefined R Address: ADDMA0 FFFFF224H, ADDMA1 FFFFF264H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDMAn	ADDMAn9	ADDMAn8	ADDMAn7	ADDMAn6	ADDMAn5	ADDMAn4	ADDMAn3	ADDMAn2	ADDMAn1	ADDMAn0	0	0	0	0	0	ODFn

ADDMAn9 to ADDMAn0	A/D Conversion Result for DMA Transfer
000H to 3FFH	Latest A/D conversion result value

ODFn	Overrun Detection Flag
0	No A/D conversion result overrun was detected.
1	At least one A/D conversion result was overrun since the last read of the ADDMAn register.
• The ODFn flag is used for indicating a DMA transfer failure of the A/D conversion results. • The ODFn flag is cleared (0), when the A/D conversion is stopped (ADCEn bit of the ADMn0 register is cleared to 0).	

Remark: n = 0, 1

14.4 Operation

14.4.1 Basic operation

A/D conversion is executed by the following procedure.

<1> The selection of the analog input and specification of the operation mode, trigger mode, etc. should be specified using the ADMn0, ADMn1 or ADMn2 registers[Note 1] (n = 0, 1).
When the ADCEn bit of the ADMn0 register is set to 1, A/D conversion starts in the A/D trigger mode. In the timer trigger mode and external trigger mode, the trigger standby state[Note 2] is set.

<2> When A/D conversion is started, the C-array voltage on the analog input side and the C-array voltage on the reference side are compared by the comparator.

<3> When the comparison of the 10 bits ends, the conversion results are stored in the ADCRnm register. When A/D conversion has been performed the specified number of times, the A/D conversion end interrupt (INTADn) is generated (n = 0, 1), (m = 0 to 9).

Notes: 1. If the setting of the ADMn0, ADMn1 or ADMn2 registers (n = 0, 1) is changed during A/D conversion, the operation immediately before is stopped, and the result of the conversion is not stored in the ADCRnm register (m = 0 to 9). The A/D conversion operation is then initialized, and conversion is executed from the beginning again.

2. During the timer trigger mode and external trigger mode, if the ADCEn bit of the ADMn0 register is set to 1, the mode changes to the trigger standby state. The A/D conversion operation is started by the trigger signal (ADCSn bit in the ADMn0 register = 1), and the trigger standby state (ADCSn bit = 0) is returned when the A/D conversion operation ends.

14.4.2 Operation mode and trigger mode

Various conversion operations can be specified for the A/D converter by specifying the operation mode and trigger mode. The operation mode and trigger mode are set by the ADMn0 and ADMn1registers. The following table shows the relationship between the operation mode and trigger mode.

Table 14-2: Relationship Between Operation Mode and Trigger Mode

Trigger Mode	Operation Mode		Register Set Value	
			ADMn0	ADMn1
A/D trigger	Select	1 buffer	xx010000B	xx000xxxB
		4 buffers	xx110000B	
	Scan		xx000000B	
Timer trigger	Select	1 buffer	xx010000B	xx010xxxB
		4 buffers	xx110000B	
	Scan		xx000000B	
External trigger	Select	1 buffer	xx010000B	xx100xxxB
		4 buffers	xx110000B	
	Scan		xx000000B	

(1) Trigger mode

There are three types of trigger modes that serve as the start timing of A/D conversion processing: A/D trigger mode, timer trigger mode, and external trigger mode. These trigger modes are set by the TRGn1 and TRGn0 bits of the ADMn1 register.

(a) A/D trigger mode

This mode starts the conversion timing of the analog input set to the ANIn0 to ANIn9 pins, and by setting the ADCEn bit of the ADMn0 register to 1, starts A/D conversion. Unless the ADCEn bit is cleared to 0 after conversion, the next conversion operation is repeated. If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and then executed from the beginning again.

(b) Timer trigger mode

This mode specifies the conversion timing of the analog input set for the ANIn0 to ANIn9 pins using signals from the inverter timer R (TMR0, TMR1).
The ADTRSELn register specifies the analog input conversion timing by selecting either one of the A/D converter trigger signals (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1) or one of the top and bottom reversal interrupts (INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD) connected to the 16-bit inverter timer R (TMR0, TMR1).
If the ADCEn bit of the ADMn0 register is set to 1, the A/D converter waits for an event input (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1, INTTR0CD, INTR0OD, INTTR1CD, or INTTR1OD), and starts conversion when the event occurs (ADCSn bit of the ADMn0 register = 1). When conversion has finished, the converter waits for an event input again (ADCSn bit = 0). If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and then executed from the beginning again.

(c) External trigger mode

This mode specifies the conversion timing of the analog input to the ANIn0 to ANIn9 pins using the ADTRGn pin.
The EGAn1 and EGAn0 bits of the ADMn1 register are used to specify the valid edge to be input to the ADTRGn pin.
When the ADCEn bit of the ADMn0 register is set to 1, the A/D converter waits for an external trigger (ADTRGn), and starts conversion when the valid edge of ADTRGn is detected (ADCSn bit of the ADMn0 register = 1). When the converter has finished its conversion operation, it waits for an external trigger again (ADCSn bit = 0).
If the valid edge is detected at the ADTRGn pin during conversion, conversion is executed from the beginning again.
If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and then executed from the beginning again.

(2) Operation mode

There are two operation modes that set the ANIn0 to ANIn9 pins: select mode and scan mode. The select mode has sub-modes that consist of 1-buffer mode and 4-buffer mode. These modes are set by the BSn and MSn bits of the ADMn0 register.

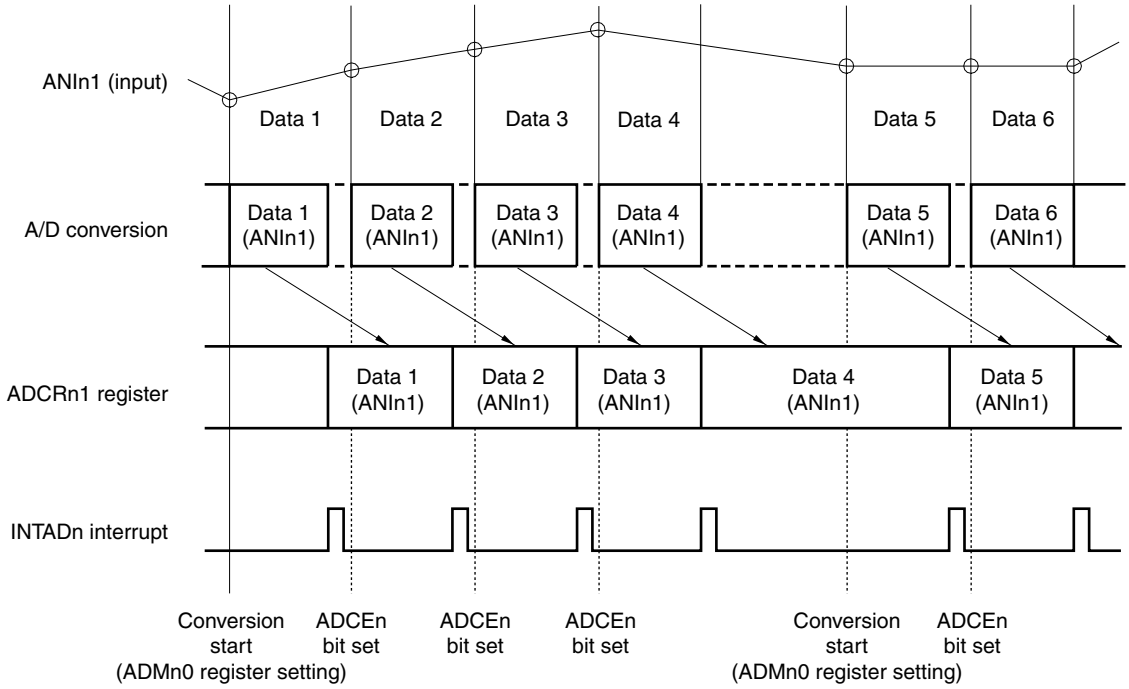
(a) Select mode

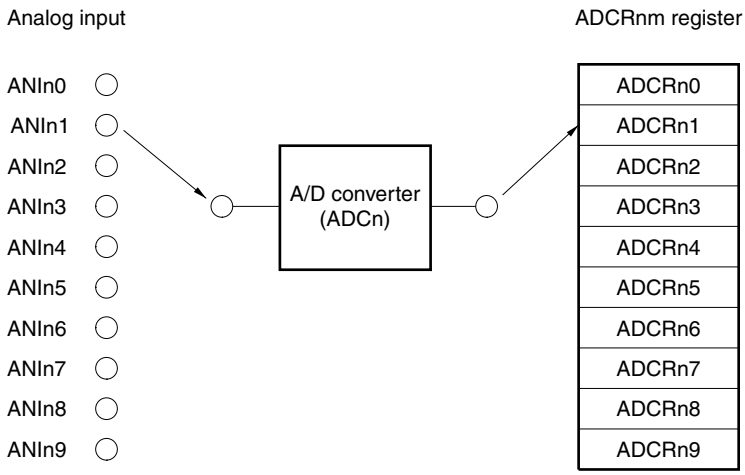
In this mode, one analog input specified by the ADMn2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input (ANInm). For this mode, the 1-buffer mode and 4-buffer mode are provided for storing the A/D conversion results (m = 0 to 9).

- **1-buffer mode**

In this mode, one analog input specified by the ADM2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input (ANInm) (m = 0 to 9). The ANInm and ADCRnm register correspond one to one, and an A/D conversion end interrupt (INTADn) is generated each time one A/D conversion ends. After conversion has finished, the next conversion operation is repeated, unless the ADCEn bit of the ADMn0 register is cleared to 0.

Figure 14-9: Select Mode Operation Timing: 1-Buffer Mode (ANIn1)



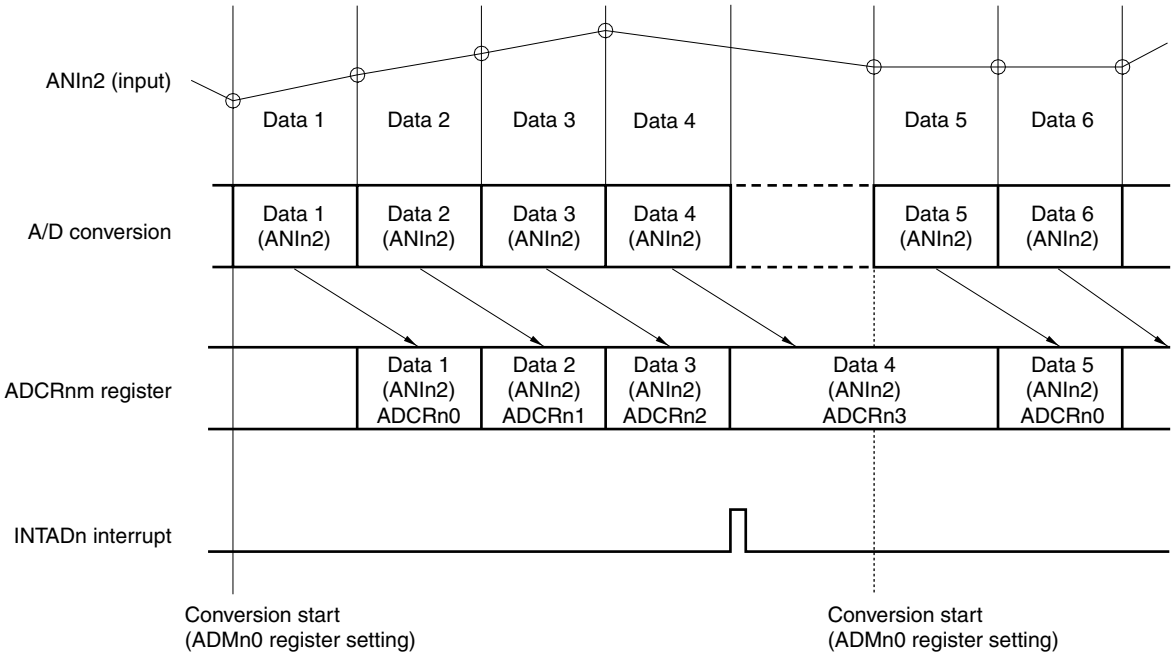


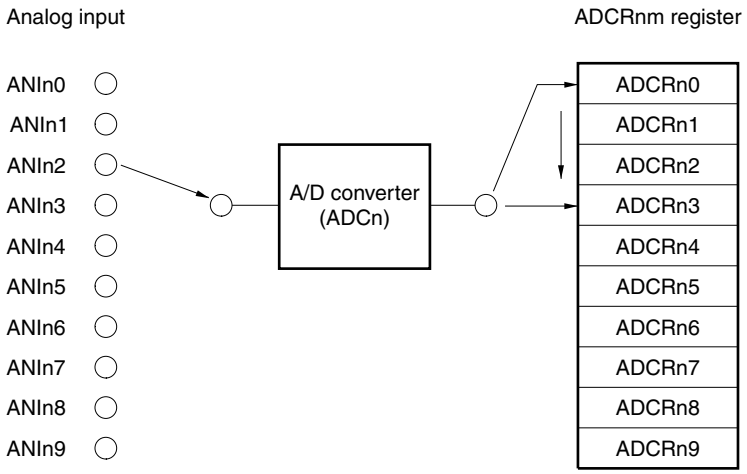
Remark: n = 0, 1
m = 0 to 9

- **4-buffer mode**

In this mode, one analog input is A/D converted and the results are stored in the ADCRnm registers. The A/D conversion end interrupt (INTADn) is generated when the four A/D conversions end (m = 0 to 3 when one of the analog input channels ANIn0 to ANIn3 is specified,
m = 4 to 7 when one of analog input channels ANIn4 to ANIn7 is specified, and m = 8 to 9 when one of the analog input channels ANIn8 or ANIn9 is specified).
After conversion has finished, the next conversion operation is repeated, unless the ADCEn bit of the ADM0 register is cleared to 0.

Figure 14-10: Select Mode Operation Timing: 4-Buffer Mode (ANIn2)



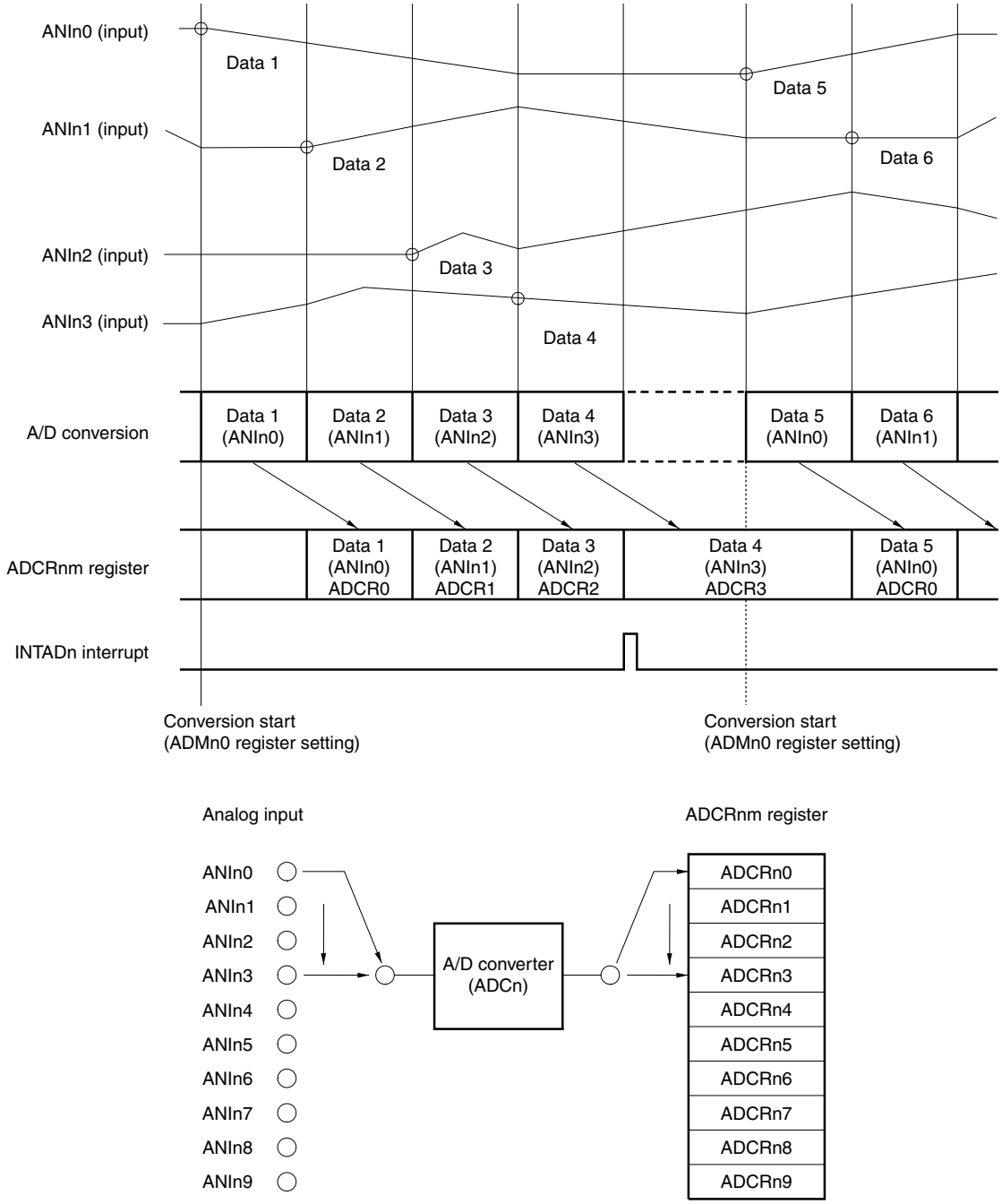


Remark: n = 0, 1
m = 0 to 9

(b) Scan mode

In this mode, the analog inputs specified by the ADMn2 register are selected sequentially from the ANIn0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRnm register corresponding to the analog input (m = 0 to 9). When the conversion of the specified analog input ends, the A/D conversion end interrupt (INTADn) is generated. After conversion has finished, the next conversion operation is repeated, unless the ADCEn bit of the ADMn0 register is cleared to 0.

Figure 14-11: Scan Mode Operation Timing: 4-Channel Scan (ANI0 to ANI3)



Remark: n = 0, 1
m = 0 to 9

14.5 Operation in A/D Trigger Mode

When the ADCEn bit of the ADMn0 register is set to 1, A/D conversion is started.

14.5.1 Select mode operation

In this mode, the analog input specified by the ADMn2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input. In the select mode, the 1-buffer mode and 4-buffer mode are supported according to the storing method of the A/D conversion results (n = 0, 1), (m = 0 to 9).

(1) 1-buffer mode (A/D trigger select: 1 buffer)

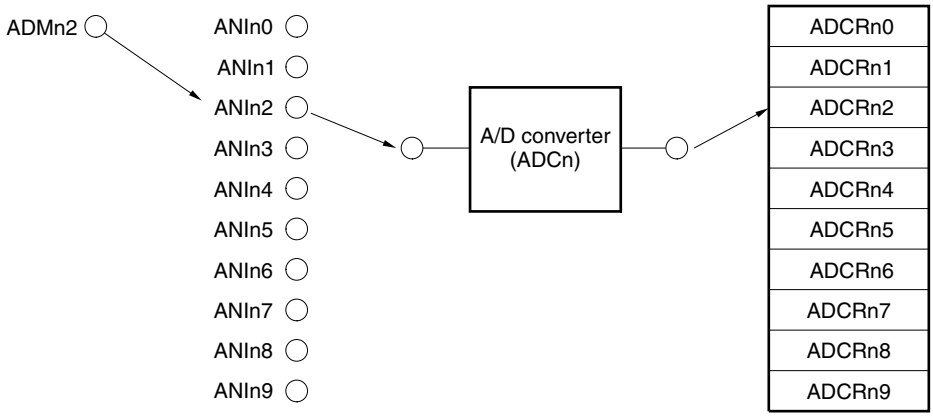
In this mode, one analog input is A/D converted once. The conversion results are stored in one ADCRn register. The analog input and ADCRn register correspond one to one.
Each time an A/D conversion is executed, an A/D conversion end interrupt (INTAD) is generated and A/D conversion ends. The next conversion operation is repeated, unless the ADCE bit of the ADM0 register is cleared to 0.

Table 14-3: Correspondence Between Analog Input Pins and ADCRnm Register (A/D Trigger Select: 1 Buffer)

Analog Input	A/D Conversion Result Register
ANInm	ADCRnm

This mode is most appropriate for applications in which the results of each first-time A/D conversion are read.

Figure 14-12: Example of 1-Buffer Mode Operation (A/D Trigger Select: 1 Buffer)



<1> The ADCEn bit of ADMn0 register is set to 1 (enable)
<2> ANIn2 is A/D converted
<3> The conversion result is stored in ADCRn2 register
<4> The INTAD interrupt is generated

Remark: n = 0, 1
m = 0 to 9

(2) 4-buffer mode (A/D trigger select: 4 buffers)

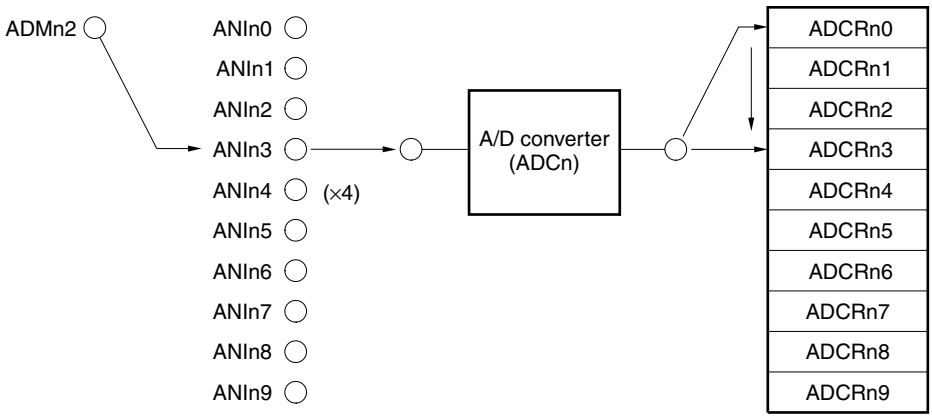
In this mode, one analog input is A/D converted four times (two times for analog input ANIn8 or ANIn9) and the results are stored in the ADCRnm register. When the 4th A/D conversion ends, an A/D conversion end interrupt (INTADn) is generated and the A/D conversion is stopped. The next conversion operation is repeated, unless the ADCEn bit of the ADMn0 register is cleared to 0.

Table 14-4: Correspondence Between Analog Input Pins and ADCRnm Register (A/D Trigger Select: 4 Buffers)

Analog Input	A/D Conversion Result Register
ANI0 to ANI3	ADCRn0 (1st time)
	ADCRn1 (2nd time)
	ADCRn2 (3rd time)
	ADCRn3 (4th time)
ANI4 to ANI7	ADCRn4 (1st time)
	ADCRn5 (2nd time)
	ADCRn6 (3rd time)
	ADCRn7 (4th time)
ANIn8, ANIn9	ADCRn8 (1st time)
	ADCRn9 (2nd time)

This mode is suitable for applications in which the average of the A/D conversion results is calculated.

Figure 14-13: Example of 4-Buffer Mode Operation (A/D Trigger Select: 4 Buffers)



<1> The ADCEn bit of ADMn0 register is set to 1 (enable)
<2> ANIn3 is A/D converted
<3> The conversion result is stored in ADCRn0 register
<4> ANIn3 is A/D converted
<5> The conversion result is stored in ADCRn1 register
<6> ANIn3 is A/D converted
<7> The conversion result is stored in ADCRn2 register
<8> ANIn3 is A/D converted
<9> The conversion result is stored in ADCRn3 register
<10> The INTAD interrupt is generated

Remark: n = 0, 1
m = 0 to 9

14.5.2 Scan mode operations

In this mode, the analog inputs specified by the ADMn2 register are selected sequentially from the ANIn0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRnm register corresponding to the analog input (m = 0 to 9).
When conversion of all the specified analog input ends, the A/D conversion end interrupt (INTADn) is generated, and A/D conversion is stopped. The next conversion operation is repeated, unless the ADCEn bit of the ADMn0 register is cleared to 0.

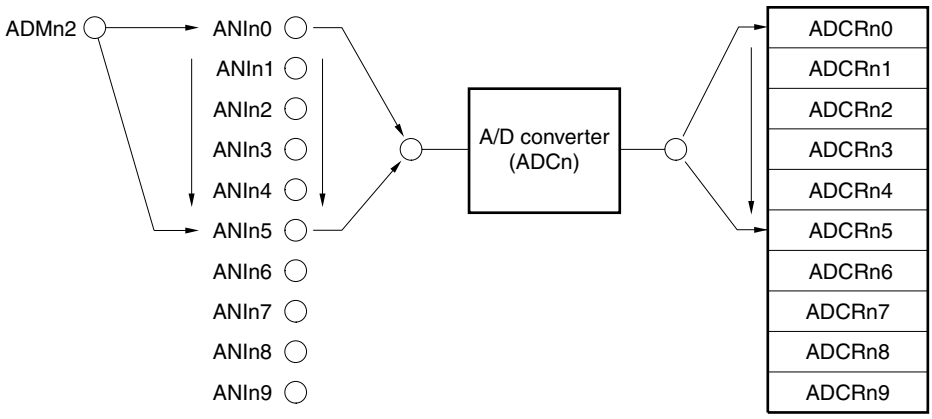
Table 14-5: Correspondence Between Analog Input Pins and ADCRnm Register (A/D Trigger Scan)

Analog Input	A/D Conversion Result Register
ANIn0 . . . ANInm**Note**	ADCRn0 . . . ADCRnm

Note: Set by the ANISn3 to ANISn0 bits of the ADMn2 register.

This mode is most appropriate for applications in which multiple analog inputs are constantly monitored.

Figure 14-14: Example of Scan Mode Operation (A/D Trigger Scan)



<1> The ADCEn bit of ADMn0 register is set to 1 (enable)
<2> ANIn0 is A/D converted
<3> The conversion result is stored in ADCRn0
<4> ANIn1 is A/D converted
<5> The conversion result is stored in ADCRn1
<6> ANIn2 is A/D converted
<7> The conversion result is stored in ADCRn2
<8> ANIn3 is A/D converted
<9> The conversion result is stored in ADCRn3
<10> ANIn4 is A/D converted
<11> The conversion result is stored in ADCRn4
<12> ANIn5 is A/D converted
<13> The conversion result is stored in ADCRn5
<14> The INTAD interrupt is generated

Remark: n = 0, 1
m = 0 to 9

14.6 Operation in Timer Trigger Mode

In this mode, the conversion timing of the analog input signal set by the ANIn0 to ANIn9 pins is defined by a timer event signal (A/D converter trigger signal, or top and bottom reversal interrupt) of the inverter timers R0 and R1 (TMR0, TMR1).
The analog input conversion timing is generated when an A/D converter trigger signal from the timers (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1), or a top or bottom reversal interrupt (INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD) is generated by inverter timer R0 or R1 (TMR0 or TMR1).
When the ADCEn bit of the ADMn0 register is set to 1, the A/D converter waits for the signal (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1) or interrupt (INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD), and starts conversion when the timer event occurs (ADCSn bit of the ADMn0 register = 1). When conversion is finished, the converter waits for a timer event signal again (ADCSn bit = 0).
If the timer event signal occurs during conversion, the conversion operation is executed from the beginning again.
If data is written to the ADMn0 to ADMn2 registers during conversion, the conversion operation is stopped and executed from the beginning again.

14.6.1 Select mode operation

In this mode, an analog input (ANIn0 to ANIn9) specified by the ADMn2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input. In the select mode, the 1-buffer mode and 4-buffer mode are provided according to the storing method of the A/D conversion results.

(1) 1-buffer mode operation (timer trigger select: 1 buffer)

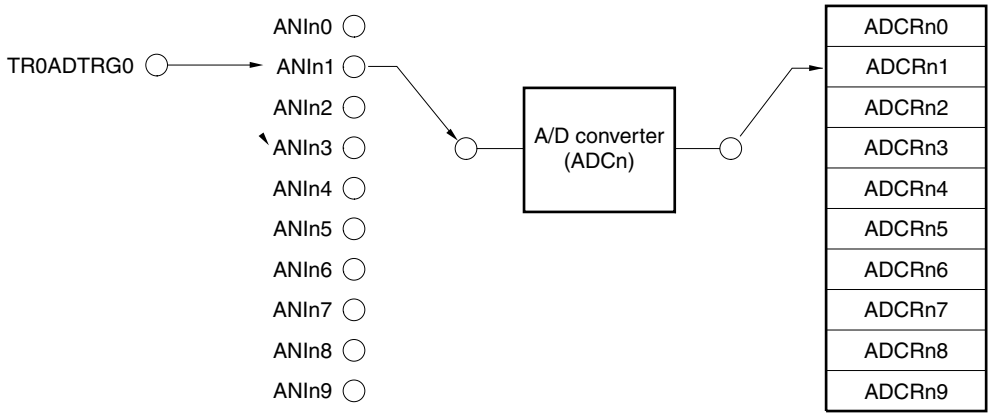
In this mode, one analog input is A/D converted once and the conversion results are stored in one ADCRnm register.
One analog input is A/D converted once using the trigger of the timer event signals (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1, INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD) and the results are stored in one ADCRnm register. An A/D conversion end interrupt (INTADn) is generated for each A/D conversion.
Unless the ADCEn bit of the ADMn0 register is cleared to 0, A/D conversion is repeated each time a timer event signal is generated.

Table 14-6: Correspondence Between Analog Input Pins and ADCRnm Register (1-Buffer Mode (Timer Trigger Select: 1 Buffer))

Trigger	Analog Input	A/D Conversion Result Register
Timer event signal (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1, INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD)	ANIn0	ADCRn0
	ANIn1	ADCRn1
	ANIn2	ADCRn2
	ANIn3	ADCRn3
	ANIn4	ADCRn4
	ANIn5	ADCRn5
	ANIn6	ADCRn6
	ANIn7	ADCRn7
	ANIn8	ADCRn8
	ANIn9	ADCRn9

Remark: n = 0, 1
m = 0 to 9

Figure 14-15: Example of 1-Buffer Mode Operation (Timer Trigger Select: 1 Buffer) (ANIn1)



<1> The ADCEn bit of ADMn0 register is set to 1 (enable)
<2> The TR0ADTRG0 signal is generated
<3> ANIn1 is A/D converted
<4> The conversion result is stored in ADCRn1
<5> The INTADn interrupt is generated

Remark: n = 0, 1

(2) 4-buffer mode operation (timer trigger select: 4 buffers)

In this mode, A/D conversion of one analog input is executed four times, and the results are stored in the ADCRnm register.

One analog input is A/D converted four times using the timer event signals (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1, INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD) as a trigger, and the results are stored in four ADCRnm registers. The A/D conversion end interrupt (INTADn) is
generated when the four A/D conversions end.

After conversion has finished, the next conversion is repeated when a timer event signal is generated, unless the ADCEn bit of the ADMn0 register is cleared to 0.

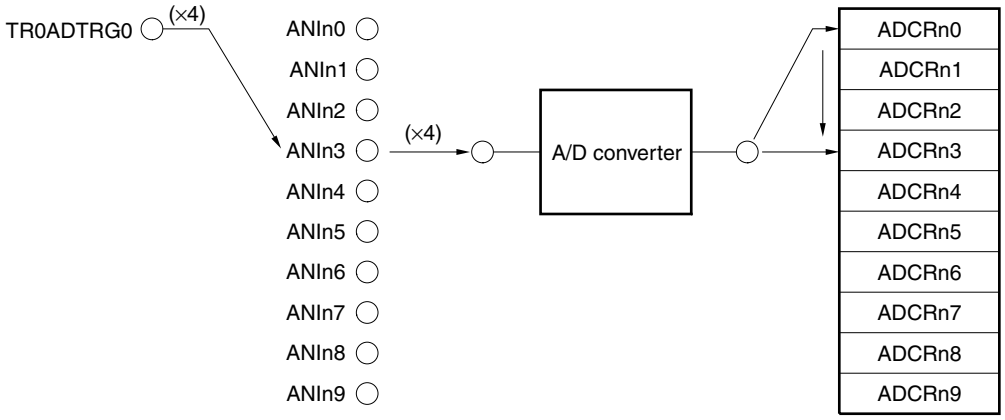
This mode is suitable for applications in which the average of the A/D conversion results is calculated.

Table 14-7: Correspondence Between Analog Input Pins and ADCRnm Register (4-Buffer Mode (Timer Trigger Select: 4 Buffers))

Trigger	Analog Input	A/D Conversion Result Register
Timer event signal (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1, INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD)	ANI0 to ANI3	ADCRn0 (1st time)
		ADCRn1 (2nd time)
		ADCRn2 (3rd time)
		ADCRn3 (4th time)
	ANI4 to ANI7	ADCRn4 (1st time)
		ADCRn5 (2nd time)
		ADCRn6 (3rd time)
		ADCRn7 (4th time)
	ANIn8, ANIn9	ADCRn8 (1st time)
		ADCRn9 (2nd time)

Remark: n = 0, 1
m = 0 to 9

Figure 14-16: Example of 4-Buffer Mode Operation (Timer Trigger Select: 4 Buffers) (ANIn3)



<1> The ADCEn bit of ADMn0 register is set to 1 (enable)
<2> TheTR0ADTRG0 signal is generated
<3> ANIn3 is A/D converted
<4> The conversion result is stored in ADCR0
<5> TheTR0ADTRG0 signal is generated
<6> ANIn3 is A/D converted
<7> The conversion result is stored in ADCR1
<8> TheTR0ADTRG0 signal is generated
<9> ANIn3 is A/D converted
<10> The conversion result is stored in ADCR2
<11> TheTR0ADTRG0 signal is generated
<12> ANIn3 is A/D converted
<13> The conversion result is stored in ADCR3
<14> The INTADn interrupt is generated

Remark: n = 0, 1

14.6.2 Scan mode operation

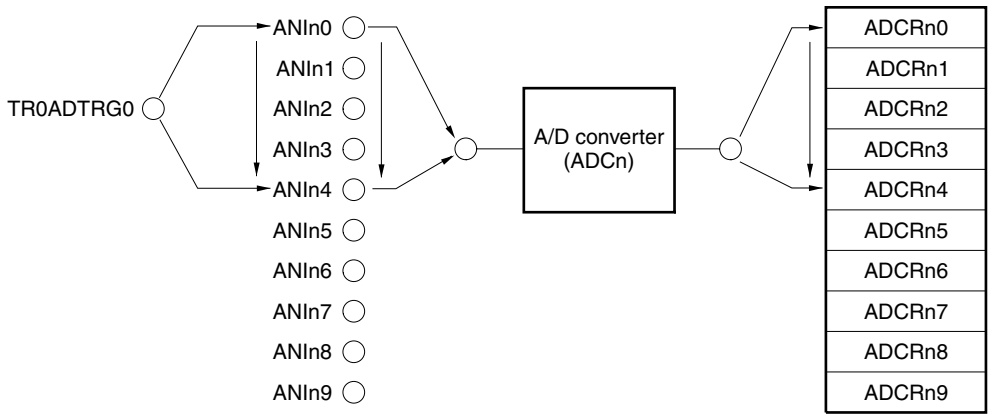
In this mode, the analog inputs specified by the ADMn2 register are selected sequentially from the ANIn0 pin and are A/D converted the specified number of times using the timer event signal as a trigger.
The result of conversion is stored in the ADCRnm register corresponding to the analog input. When all the specified analog input signals have been converted, an A/D conversion end interrupt (INTADn) occurs.
After conversion has finished, the A/D converter waits for a trigger unless the ADCEn bit of the ADMn0 register is cleared to 0. When a timer event occurs again, the converter starts A/D conversion again, starting from the ANIn0 input.
This mode is most appropriate for applications in which multiple analog inputs are constantly monitored.

Table 14-8: Correspondence Between Analog Input Pins and ADCRnm Register (Scan Mode (Timer Trigger Scan))

Trigger	Analog Input	A/D Conversion Result Register
Timer event signal (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1, INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD)	ANIn0	ADCRn0
	ANIn1	ADCRn1
	ANIn2	ADCRn2
	ANIn3	ADCRn3
	ANIn4	ADCRn4
	ANIn5	ADCRn5
	ANIn6	ADCRn6
	ANIn7	ADCRn7
	ANIn8	ADCRn8
	ANIn9	ADCRn9

Remark: n = 0, 1
m = 0 to 9

Figure 14-17: Example of Scan Mode Operation (Timer Trigger Scan) (ANIn0 to ANIn4)



<1> The ADCEn bit of ADMn0 register is set to 1 (enable)
<2> The TR0ADTRG0 signal is generated
<3> ANIn0 is A/D converted
<4> The conversion result is stored in ADCRn0
<5> ANIn1 is A/D converted
<6> The conversion result is stored in ADCRn1
<7> ANIn2 is A/D converted
<8> The conversion result is stored in ADCRn2
<9> ANIn3 is A/D converted
<10> The conversion result is stored in ADCRn3
<11> ANIn4 is A/D converted
<12> The conversion result is stored in ADCRn4
<13> The INTADn interrupt is generated

Remark: n = 0, 1

14.7 Operation in External Trigger Mode

In this mode, the conversion timing of the analog signals input to the ANIn0 to ANIn9 pins is specified by the ADTRGn pin.
Detection of the valid edge at the ADTRGn input pin is specified by using the EGAn1 and EGAn0 bits of the ADMn1 register.
When the ADCEn bit of the ADMn0 register is set to 1, the A/D converter waits for an external trigger (ADTRGn), and starts conversion when the valid edge of ADTRGn is detected (ADCSn bit of the ADMn0 register = 1). When the converter has ended conversion, it waits for the external trigger again (ADCSn bit = 0).
If the valid edge is detected at the ADTRGn pin during conversion, conversion is executed from the beginning again.
If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and executed from the beginning again.

14.7.1 Select mode operations

In this mode, one analog input (ANIn0 to ANIn9) specified by the ADMn2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input. In the select mode, there are two select modes: 1-buffer mode and 4-buffer mode, according to the storing method of the conversion results.

(1) 1-buffer mode (external trigger select: 1 buffer)

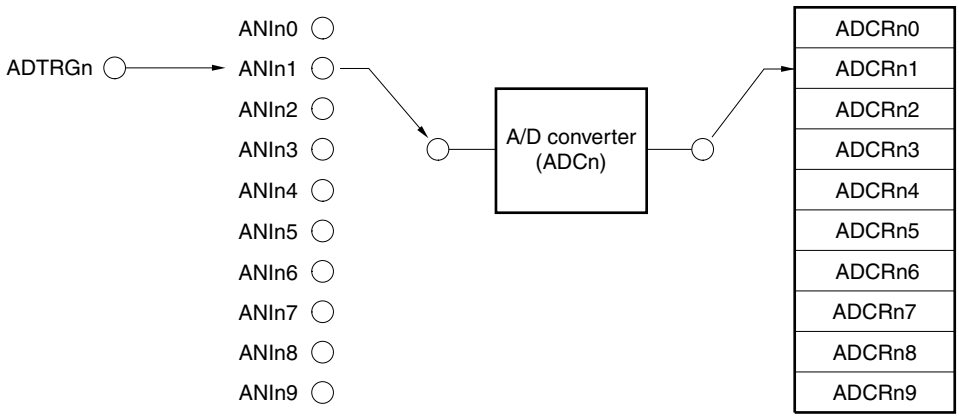
In this mode, one analog input is A/D converted using the ADTRGn signal as a trigger. The conversion results are stored in one ADCRnm register. The analog input and the A/D conversion results register correspond one to one. The A/D conversion end interrupt (INTADn) is generated for each A/D conversion, and A/D conversion is stopped.

Table 14-9: Correspondence Between Analog Input Pins and ADCRnm Register (External Trigger Select: 1 Buffer)

Trigger	Analog Input	A/D Conversion Result Register
ADTRGn signal	ANInm	ADCRnm

While the ADCEn bit of the ADMn0 register is 1, A/D conversion is repeated every time a trigger is input from the ADTRGn pin.
This mode is most appropriate for applications in which the results are read after each A/D conversion.

Figure 14-18: Example of 1-Buffer Mode Operation (External Trigger Select: 1 Buffer) (ANIn1)



<1> The ADCEn bit of ADMn0 is set to 1 (enable)
<2> The external trigger is generated
<3> ANIn1 is A/D converted
<4> The conversion result is stored in ADCRn1
<5> The INTADn interrupt is generated

Remark: n = 0, 1
m = 0 to 9

(2) 4-buffer mode (external trigger select: 4 buffers)

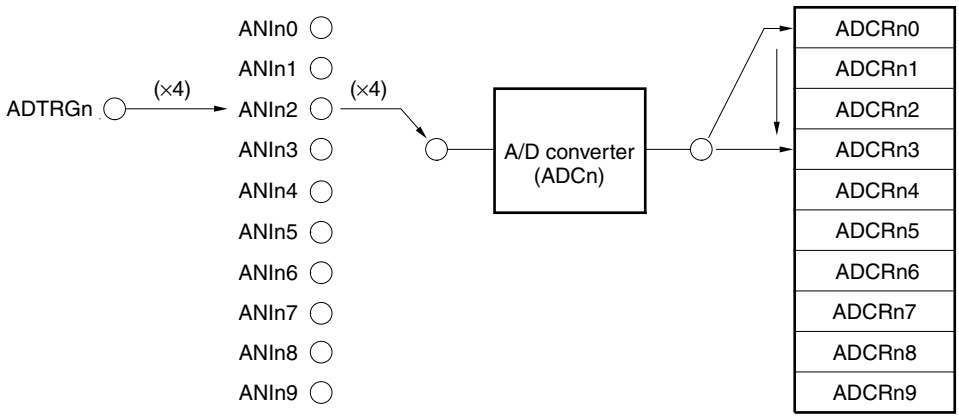
In this mode, one analog input is A/D converted four times using the ADTRGn signal as a trigger and the results are stored in the ADCRnm register. The A/D conversion end interrupt (INTADn) is generated and A/D conversion is stopped after the 4th A/D conversion.

Table 14-10: Correspondence Between Analog Input Pins and ADCRnm Register (External Trigger Select: 4 Buffers))

Trigger	Analog Input	A/D Conversion Result Register
ADTRGn signal	ANI0 to ANI3	ADCRn0 (1st time)
		ADCRn1 (2nd time)
		ADCRn2 (3rd time)
		ADCRn3 (4th time)
	ANI4 to ANI7	ADCRn4 (1st time)
		ADCRn5 (2nd time)
		ADCRn6 (3rd time)
		ADCRn7 (4th time)
	ANIn8, ANIn9	ADCRn8 (1st time)
		ADCRn9 (2nd time)

While the ADCEn bit of the ADMn0 register is 1, A/D conversion is repeated every time a trigger is input from the ADTRGn pin.
This mode is suitable for applications in which the average of the A/D conversion results is calculated.

Figure 14-19: Example of 4-Buffer Mode Operation (External Trigger Select: 4 Buffers) (ANIn2)



<1> The ADCEn bit of ADMn0 register is set to 1 (enable)
<2> The external trigger is generated
<3> ANIn2 is A/D converted
<4> The conversion result is stored in ADCRn0
<5> The external trigger is generated
<6> ANIn2 is A/D converted
<7> The conversion result is stored in ADCRn1
<8> The external trigger is generated
<9> ANIn2 is A/D converted
<10> The conversion result is stored in ADCRn2
<11> The external trigger is generated
<12> ANIn2 is A/D converted
<13> The conversion result is stored in ADCRn3
<14> The INTAD interrupt is generated

Remark: n = 0, 1
m = 0 to 9

14.7.2 Scan mode operation

In this mode, the analog inputs specified by the ADMn2 register are selected sequentially from the ANIn0 pin using the ADTRGn signal as a trigger, and A/D converted. The A/D conversion results are stored in the ADCRnm register corresponding to the analog input ANInm (n = 0, 1)(m = 0 to 9).
When conversion of all the specified analog inputs has ended, the A/D conversion end interrupt (INTADn) is generated.n Unless the ADCE bit of the ADMn0 register is cleared to 0 after end of conversion, the A/D converter waits for a trigger. The converter starts A/D conversion from the ANIn0 input when a trigger is input to the ADTRGn pin again.

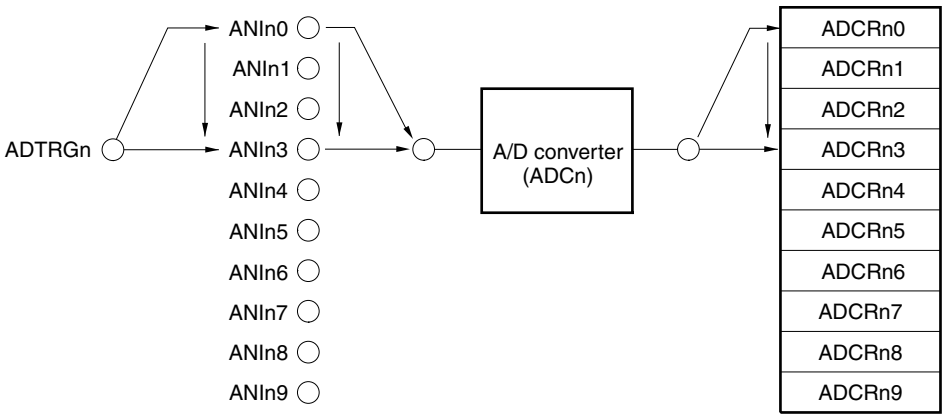
Table 14-11: Correspondence Between Analog Input Pins and ADCRnm Register (External Trigger Scan)

Trigger	Analog Input	A/D Conversion Result Register
ADTRGn signal	ANIn0	ADCRn0
	ANIn1	ADCRn1
	ANIn2	ADCRn2
	ANIn3	ADCRn3
	ANIn4	ADCRn4
	ANIn5	ADCRn5
	ANIn6	ADCRn6
	ANIn7	ADCRn7
	ANIn8	ADCRn8
	ANIn9	ADCRn9

When a trigger is input to the ADTRGn pin while the ADCEn bit of the ADMn0 register is 1, A/D conversion is started again.
This is most appropriate for applications in which multiple analog inputs are constantly monitored.

Remark: n = 0, 1
m = 0 to 9

Figure 14-20: Example of Scan Mode Operation (External Trigger Scan) (ANIn0 to ANIn3)



<1> The ADCEn bit of ADMn0 register is set to 1 (enable)
<2> The external trigger is generated
<3> ANIn0 is A/D converted
<4> The conversion result is stored in ADCRn0
<5> ANIn1 is A/D converted
<6> The conversion result is stored in ADCRn1
<7> ANIn2 is A/D converted
<8> The conversion result is stored in ADCRn2
<9> ANIn3 is A/D converted
<10> The conversion result is stored in ADCRn3
<11> The INTADn interrupt is generated

Remark: n = 0, 1

14.8 Precautions

(1) Stopping conversion operation

When the ADCEn bit of the ADMn0 register is cleared to 0 during a conversion operation, the conversion operation stops and the conversion results are not stored in the ADCRnm register (n = 0, 1), (m = 0 to 9).

(2) External/timer trigger interval

Set the interval (input time interval) of the trigger in the external or timer trigger mode longer than the conversion time specified by the FRn3 to FRn0 bits of the ADMn1 register.

When 0 < interval ≤conversion operation time

When the following external trigger or timer trigger is input during a conversion operation, the conversion operation is aborted and the conversion starts according to the last external trigger input or timer trigger input.

When conversion operations are aborted, the conversion results are not stored in the ADCRnm register (n = 0, 1) (m = 0 to 9). However, the number of times the trigger has been input is counted. When an interrupt occurs, the values that have been converted are stored in the ADCRnm register.

(3) Operation in HALT mode

A/D conversion continues in the HALT mode. When this mode is released by NMI input or unmasked maskable interrupt input (see section **8.3.2 (2) ”Releasing HALT mode” on page 246**), the ADMn0, ADMn1, and ADMn2 registers as well as the ADCRnm register hold the value (n = 0, 1) (m = 0 to 9).

(4) Input range of ANIn0 to ANIn9

Use the input voltage at ANIn0 to ANIn9 within the specified range. If a voltage outside the range of AV_{REF} is input to any of these pins (even within the absolute maximum rating range), the converted value of the channel is undefined. In addition, the converted value of the other channels may also be affected.

(5) Conflicts

(a) Conflict between writing A/D conversion result registers (ADCRnm, ADCRnmH) at end of conversion and reading ADCRnm and ADCRnmH registers by instruction

Reading the ADCRnm and ADCRnmH registers takes precedence. After these registers have been read, the new conversion result is written to the ADCRnm and ADCRnmH registers.

(b) Conflict between writing ADCRnm and ADCRnmH at end of conversion and input of external trigger signal

The external trigger signal is not accepted during A/D conversion. Therefore, it is not accepted while ADCRnm and ADCRnmH are being written.

(c) Conflict between writing ADCRnm and ADCRnmH at end of conversion and writing ADMn1 or ADMn2 register

If ADMn1 or ADMn2 register is written immediately after ADCRnm and ADCRnmH have been written on completion of A/D conversion, the conversion result is written to the ADCRnm and ADCRnmH registers, but the A/D conversion end interrupt (INTADn) may not occur depending on the timing.

Chapter 15 Asynchronous Serial Interface C (UARTC)

15.1 Features

- Transfer speed: 16 bps to 2000 kbps
- Full-duplex communication: Internal UARTC receive data register n (UCnRX)
  Internal UARTC transmit data register n (UCnTX)
- 2-pin configuration: TXDCn: Transmit data output pin
  RXDCn: Receive data input pin
- Receive error output function
  - Parity error
  - Framing error
  - Overrun error
- Interrupt sources: 3
  - Receive error interrupt (INTUCnRE)
  - Reception complete interrupt (INTUCnR)
  - Transmission enable interrupt (INTUCnT)
- Character length: 7, 8 bits
- Parity function: Odd, even, 0, none
- Transmission stop bit: 1, 2 bits
- On-chip dedicated baud rate generator
- MSB/LSB-first transfer selectable
- Transmit/receive data level inversion possible
- 13 to 20 bits selectable for the SBF (Sync Break Field) in the LIN (Local Interconnect Network) communication format
- Recognition of 11 bits or more possible for SBF reception in LIN communication format
- SBF reception flag provided
- Extension bit operation possible (uses parity bit as 9th data bit)
- Transfer and reception status flags

Remark: n = 0, 1

15.2 Configuration

(1) UARTCn control register 0 (UCnCTL0)

The UCnCTL0 register is an 8-bit register used to specify the asynchronous serial interface operation.

(2) UARTCn control register 1 (UCnCTL1)

The UCnCTL1 register is an 8-bit register used to select the input clock for the asynchronous serial interface.

(3) UARTCn control register 2 (UCnCTL2)

The UCnCTL2 register is an 8-bit register used to control the baud rate for the asynchronous serial interface.

(4) UARTCn option control register 0 (UCnOPT0)

The UCnOPT0 register is an 8-bit register used to control serial transfer for the asynchronous serial interface.

(5) UARTCn option control register 1 (UCnOPT1)

The UCnOPT1 register is an 8-bit register used to control the extension bit operation.

(6) UARTCn status register (UCnSTR)

The UCnSTR register consists of flags indicating the error contents when a reception error occurs. Each one of the reception error flags is set (to 1) upon occurrence of a reception error and is reset (to 0) by reading the UCnSTR register.

(7) UARTCn status register 1 (UCnSTR1)

The UCnSTR1 register indicates the operating status during a reception.

(8) UARTCn receive shift register

This is a shift register used to convert the serial data input to the RXDCn pin into parallel data. Upon reception of 1 byte of data and detection of the stop bit, the receive data is transferred to the UCnRX register.
This register cannot be manipulated directly.

(9) UARTCn receive data register (UCnRX)

The UCnRX register is an 8-bit register that holds receive data. When 7 characters are received, 0 is stored in the highest bit (when LSB first received).
In the reception enabled status, receive data is transferred from the UARTCn receive shift register to the UCnRX register in synchronization with the completion of shift-in processing of 1 frame.
Transfer to the UCnRX register also causes reception complete interrupt (INTUCnR) to be output.

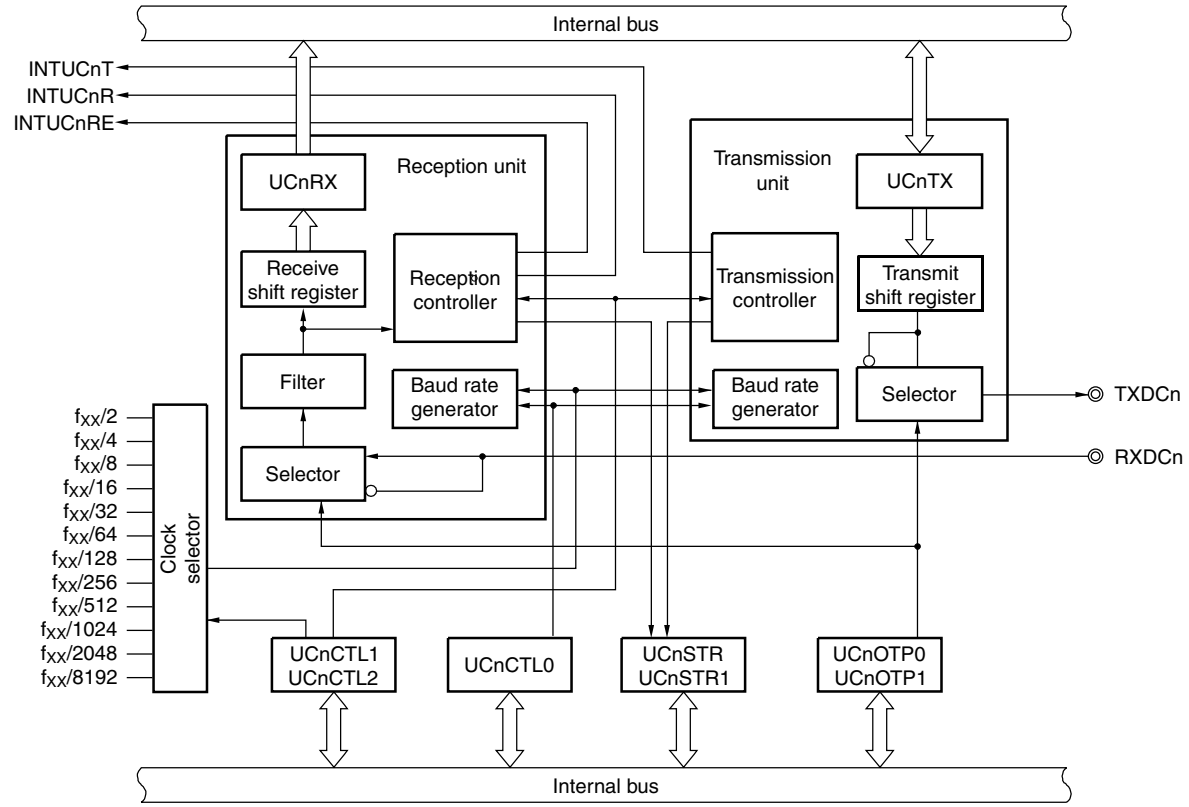
(10) UARTCn transmit shift register

The transmit shift register is a shift register used to convert the parallel data transferred from the UCnTX register into serial data.
When 1 byte of data is transferred from the UCnTX register, the shift register data is output from the TXDCn pin.
This register cannot be manipulated directly.

(11) UARTCn transmit data register (UCnTX)

The UCnTX register is an 8-bit transmit data buffer. Transmission starts when transmit data is written to the UCnTX register. When data can be written to the UCnTX register (when data of one frame is transferred from the UCnTX register to the UARTCn transmit shift register), the transmission enable interrupt (INUCnT) is generated.

Figure 15-1: Block Diagram of Asynchronous Serial Interface n



Remarks: 1. n = 0, 1

2. f_{XX}: Internal system clock

15.3 Control Registers

(1) UARTCn control register 0 (UCnCTL0)

The UCnCTL0 register is an 8-bit register that controls the UARTCn serial transfer operation.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 10H.

Caution: Be sure to set the UCnPWR bit = 1 and the UCnRXE bit = 1 while the RXDCn pin is high level (when UCnRDL bit of UCnOP0 register = 0).
If the UCnPWR bit = 1 and the UCnRXE bit = 1 are set while the RXDCn pin is low level, reception will inadvertently start.

Figure 15-2: UARTCn Control Register 0 (UCnCTL0) (1/2)

After reset: 10H R/W Address: UC0CTL0 FFFFFA00H, UC1CTL0 FFFFFA20H

	7	6	5	4	3	2	1	0
UCnCTL0 (n = 0, 1)	UCnPWR	UCnTXE	UCnRXE	UCnDIR	UCnPS1	UCnPS0	UCnCL	UCnSL

UCnPWR	UARTCn Operation Control
0	Stops clock operation (UARTCn reset asynchronously)
1	Enables operating clock operation
Operating clock control and UARTCn asynchronous reset are performed with the UCnPWR bit. The TXDCn pin output is fixed to high level by setting the UCnPWR bit to 0.	

UCnTXE	Transmission Operation Enable
0	Stops transmission operation
1	Enables transmission operation
• The TXDCn pin output is fixed to high level by setting the UCnPWR bit to 0. Since the UCnTXE bit is initialized by the operating clock, to initialize the transmission unit, set UCnTXE from 0 to 1, and 2 clocks later, the transmission enabled status is entered. • When UCnPWR bit = 0, the value written to the UCnTXE bit is ignored.	

UCnRXE	Reception Operation Enable
0	Stops reception operation
1	Enables reception operation
• The receive operation is stopped by setting the UCnRXE bit to 0. Therefore, even if the prescribed data is transferred, no reception completion interrupt is output and the UARTCn reception data register (UCnRX) is not updated. Since the UCnRXE bit is synchronized using the operating clock, to initialize the reception unit, set UCnRXE from 0 to 1, and 2 clocks later, the reception enabled status is entered. • When UCnPWR bit = 0, the value written to the UCnRXE bit is ignored.	

Figure 15-2: UARTCn Control Register 0 (UCnCTL0) (2/2)

UCnDIR	Transfer Direction Selection
0	MSB-first transfer
1	LSB-first transfer
This bit can be rewritten only when UCnPWR bit = 0 or UCnTXE bit = UCnRXE bit = 0.	

UCnPS1	UCnPS0	Parity Selection	
		During Transmission	During Reception
0	0	No parity output	Reception with no parity
0	1	0 parity output	Reception with 0 parity
1	0	Odd parity output	Odd parity check
1	1	Even parity output	Even parity check

- These bits can be rewritten only when UCnPWR bit = 0 or UCnTXE bit = UCnRXE bit = 0.
- If “Reception with 0 parity” is selected during reception, a parity check is not performed. Therefore, since the UCnPE bit of the UCnSTA0 register is not set, no error interrupt is output.
- When transmission and reception are performed in the LIN format, set the UCnPS1 and UCnPS0 bits to 00B.

UCnCL	Data Character Length Specification
0	7 bits
1	8 bits
This bit can be rewritten only when UCnPWR bit = 0 or UCnTXE bit = UCnRXE bit = 0.	

UCnSL	Stop Bit Length Specification
0	1 bit
1	2 bits
This bit can be rewritten only when UCnPWR bit = 0 or UCnTXE bit = UCnRXE bit = 0.	

Remark: For details of parity, see **15.5.9 ”Parity types and operations” on page 621**.

(2) UARTCn control register 1 (UCnCTL1)

The UCnCTL1 register is an 8-bit register that selects the UARTCn base clock (f_{XCLK}).
This register can be read or written in 8-bit units.
Reset input clears this register to 00H.

Figure 15-3: UARTCn Control Register 1 (UCnCTL1)

After reset: 00H R/W Address: UC0CTL1 FFFFFA01H, UC1CTL1 FFFFFA21H

	7	6	5	4	3	2	1	0
UCnCTL1 (n = 0, 1)	0	0	0	0	UCnCKS3	UCnCKS2	UCnCKS1	UCnCKS0

UCnCKS3	UCnCKS2	UCnCKS1	UCnCKS0	Base clock (f_{XCLK}) selection
0	0	0	0	$f_{XX}/2$
0	0	0	1	$f_{XX}/4$
0	0	1	0	$f_{XX}/8$
0	0	1	1	$f_{XX}/16$
0	1	0	0	$f_{XX}/32$
0	1	0	1	$f_{XX}/64$
0	1	1	0	$f_{XX}/128$
0	1	1	1	$f_{XX}/256$
1	-	0	0	$f_{XX}/512$
1	-	0	1	$f_{XX}/1024$
1	-	1	0	$f_{XX}/2048$
1	-	1	1	$f_{XX}/8192$

Remark: f_{XX}: Internal system clock

(3) UARTCn control register 2 (UCnCTL2)

The UCnCTL2 register is an 8-bit register that specifies the divisor to control the baud rate (serial transfer speed) clock of UARTCn.
This register can be read or written in 8-bit units.
Reset input sets this register to FFH.

Figure 15-4: UARTCn Control Register 2 (UCnCTL2)

After reset: FFH R/W Address: UC0CTL2 FFFFFA02H, UC1CTL2 FFFFFA22H

	7	6	5	4	3	2	1	0
UCnCTL2 (n = 0, 1)	UCnBRS7	UCnBRS6	UCnBRS5	UCnBRS4	UCnBRS3	UCnBRS2	UCnBRS1	UCnBRS0

UCn BRS7	UCn BRS6	UCn BRS5	UCn BRS4	UCn BRS3	UCn BRS2	UCn BRS1	UCn BRS0	Default (k)	Serial clock
0	0	0	0	0	0	-	-	-	Setting prohibited
0	0	0	0	0	1	0	0	4	$f_{XCLK}/4$
0	0	0	0	0	1	0	1	5	$f_{XCLK}/5$
0	0	0	0	0	1	1	0	6	$f_{XCLK}/6$
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	$f_{XCLK}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK}/255$

Remark: f_{XCLK}: Clock frequency selected by the UCnCKS3 to UCnCKS0 bits of the UCnCTL1 register

(4) UARTCn option control register 0 (UCnOPT0)

The UCnOPT0 register is an 8-bit register that controls the serial transfer operation of the UARTCn register.

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

Reset input sets this register to 14H.

Figure 15-5: UARTCn Option Control Register 0 (UCnOPT0) (1/2)

After reset: 14H R/W Address: UC0OPT0 FFFFFA03H, UC1OPT0 FFFFFA23H

	7	6	5	4	3	2	1	0
UCnOPT0 (n = 0, 1)	UCnSRF	UCnSRT	UCnSTT	UCnSLS2	UCnSLS1	UCnSLS0	UCnTDL	UCnRDL

UCnSRF	SBF Reception Flag
0	When UCnPWR of UCnCTL0 register = 0 and UCnRXE of UCnCTL0 register = 0 are set. Also upon normal end of SBF reception.
1	During SBF reception
• SBF (Sync Brake Field) reception is judged during LIN communication. • The UCnSRF bit is held high when a SBF reception error occurs, and then SBF reception is started again.	

UCnSRT	SBF Reception Trigger
0	–
1	SBF reception trigger
• This is the SBF reception trigger bit during LIN communication, and when read, “0” is always read. For SBF reception, set the UCnSRT bit (to 1) to enable reception. • Set the UCnSRT bit after setting the UCnPWR bit of the UCnCTL0 register to 1 and the UCnRXE bit of the UCnCTL0 register to 1.	

UCnSTT	SBF Transmission Trigger
0	–
1	SBF transmission trigger
• This is the SBF transmission trigger bit during LIN communication, and when read, “0” is always read. • Set the UCnSRT bit after setting the UCnPWR bit of the UCnCTL0 register to 1 and the UCnRXE bit of the UCnCTL0 register to 1.	

Figure 15-5: UARTCn Option Control Register 0 (UCnOPT0) (2/2)

UCnSLS2	UCnSLS1	UCnSLS0	SBF Length Selection
1	0	1	13-bit output (reset value)
1	0	0	14-bit output
1	1	1	15-bit output
0	1	0	16-bit output
0	0	1	17-bit output
0	0	0	18-bit output
0	1	1	19-bit output
1	1	0	20-bit output
This register can be set when the UCnPWR bit of the UCnCTL0 register is 0 or when the UCnRXE bit of the UCnCTL0 register is 0.			

UCnTDL	Transmit Data Level
0	Normal output of transfer data
1	Inverted output of transfer data
• The value of the TXDCn pin can be inverted using the UCnTDL bit. • This bit can be set when the UCnPWR bit of the UCnCTL0 register is 0 or when the UCnTXE bit of the UCnCTL0 register is 0.	

UCnRDL	Receive Data Level
0	Normal input of transfer data
1	Inverted input of transfer data
• The value of the RXDCn pin can be inverted using the UCnRDL bit. • This bit can be set when the UCnPWR bit of the UCnCTL0 register is 0 or the UCnRXE bit of the UCnCTL0 register is 0.	

(5) UARTCn option control register 1 (UCnOPT1)

The UCnOPT1 register is an 8-bit register that controls the extension bit operation of the UARTCn.
The register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Figure 15-6: UARTCn Option Control Register 1 (UCnOPT1)

After reset: 00H R/W Address: UC0OPT1 FFFFFA0AH, UC1OPT1 FFFFFA2AH

	7	6	5	4	3	2	1	0
UCnOPT1 (n = 0, 1)	0	0	0	0	0	0	0	UCnEBE

UCnEBE	Extension Bit Operation Enable
0	Extension bit operation disabled. Transfer data length set by UCnCL bit of the UCnCTL0 register.
1	Extension bit operation enabled.

- During extension bit operation a 9-th data bit is sent or received instead of the parity bit.
- Extension bit operation is only effective when the parity selection is set to no parity (UCnPS1, UCnPS0 bits = 00B), and the character length is set to 8 bits (UCnCL bit = 1). In all other cases the setting of UCnEBE bit is ignored.

Table 15-1: Relation between UARTCn Register Settings and Data Format

Register Bit Settings					Data Format				
UCnEBE	UCnPS1	UCnPS0	UCnCL	UCnSL	D0 - D6	D7	D8	D9	D10
0	0	0	0	0	Data	Stop			
			0	1	Data	Stop	Stop		
			1	0	Data	Data	Stop		
			1	1	Data	Data	Stop	Stop	
	other than 00B		0	0	Data	Parity	Stop		
			0	1	Data	Parity	Stop	Stop	
			1	0	Data	Data	Parity	Stop	
			1	1	Data	Data	Parity	Stop	Stop
1	0	0	0	0	Data	Stop			
			0	1	Data	Stop	Stop		
			1	0	Data	Data	Data[Note]	Stop	
			1	1	Data	Data	Data[Note]	Stop	Stop
	other than 00B		0	0	Data	Parity	Stop		
			0	1	Data	Parity	Stop	Stop	
			1	0	Data	Data	Parity	Stop	
			1	1	Data	Data	Parity	Stop	Stop

Note: Insertion of extension bit

(6) UARTCn status register (UCnSTR)

The UCnSTR register is an 8-bit register that displays the UARTCn transfer status and reception error contents.

This register can be read or written in 8-bit or 1-bit units, but the UCnTSF bit is a read-only bit, while the UCnPE, UCnFE, and UCnOVE bits can both be read and written. However, these bits can only be cleared by writing 0 and they cannot be set by writing 1. (If 1 is written to them, the hold status is entered.)

The initialization conditions are shown below.

Register/Bit	Initialization Conditions
UCnSTR register	Reset input UCnPWR bit of UCnCTL0 register = 0
UCnTSF bit	UCnTXE bit of UCnCTL0 register = 0
UCnPE, UCnFE, UCnOVE bits	0 write UCnRXE bit of UCnCTL0 register = 0

Figure 15-7: UARTCn Status Register (UCnSTR) (1/2)

After reset: 00H R/W Address: UC0STR FFFFFA04H, UC1STR FFFFFA24H

	7	6	5	4	3	2	1	0
UCnSTR	UCnTSF	0	0	0	0	UCnPE	UCnFE	UCnOVE

(n = 0, 1)

UCnTSF	Transfer Status Flag
0	• When UCnPWR bit of UCnCTL0 register = 0 or UCnTXE bit of UCnCTL0 register = 0 has been set. • When, following transfer completion, there was no next data transfer from UCnTX
1	Write to UCnTXB bit
The UCnTSF bit is always 1 when performing continuous transmission. When initializing the transmission unit, check that the UCnTSF bit = 0 before performing initialization. The transmit data is not guaranteed when initialization is performed while UCnTSF bit = 1.	

UCnPE	Parity Error Flag
0	• When UCnPWR bit of UCnCTL0 register = 0 or UCnRXE bit of UCnCTL0 register = 0 has been set. • When 0 has been written
1	• When parity of data and parity bit do not match during reception.
• The operation of the UCnPE bit is controlled by the settings of the UCnPS1 and UCnPS0 bits of the UCnCTL0 register. • The UCnPE bit can be read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the hold status is entered.	

Figure 15-7: UARTCn Status Register (UCnSTR) (2/2)

UCnFE	Framing Error Flag
0	• When UCnPWR bit of UCnCTL0 register = 0 or UCnRXE bit of UCnCTL0 register = 0 has been set • When 0 has been written
1	When no stop bit is detected during reception
• Only the first bit of the receive data stop bits is checked, regardless of the value of the UCnSL bit of the UCnCTL0 register. • The UCnFE bit can be both read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the hold status is entered.	

UCnOVE	Overrun Error Flag
0	• When UCnPWR bit of UCnCTL0 register = 0 or UCnRXE bit of UCnCTL0 register = 0 has been set. • When 0 has been written
1	When receive data has been set to the UCnRXB register and the next receive operation is completed before that receive data has been read
• When an overrun error occurs, the data is discarded without the next receive data being written to the receive buffer. • The UCnOVE bit can be both read and written, but it can only be cleared by writing 0 to it. When 1 is written to this bit, the hold status is entered.	

(7) UARTCn status register 1 (UCnSTR1)

The UCnSTR1 register is an 8-bit register that displays the UARTCn reception status.
The register is read only, and be read in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Figure 15-8: UARTCn Status Register 1 (UCnSTR1)

After reset: 00H R Address: UC0STR1 FFFFFA0BH, UC1STR1 FFFFFA2BH

	7	6	5	4	3	2	1	0
UCnSTR1 (n = 0, 1)	0	0	0	0	0	0	0	UCnRSF

UCnRSF	Receive Status Flag
0	• When UCnPWR bit of UCnCTL0 register = 0 or UCnRXE bit of UCnCTL0 register = 0 has been set. • When the stop bit has been detected.
1	During reception, when the start bit has been detected.
The UCnRSF flag is set (1) by the start bit detection, and it is cleared (0) by detection of the first stop bit condition. In case of a two stop bit setting (UCnSL bit of UCnCTL0 register = 1), the UCnRSF flag is cleared during the first stop bit timing, simultaneously with the reception complete interrupt timing (INTUCnR)	

(8) UARTCn receive data register (UCnRX, UCnRXL)

The UCnRX register is a 16-bit buffer register that stores parallel data converted by receive shift register. It is overlayed by an 8-bit register UCnRXL on the lower 8 bits, which stores the lower byte of the received data.

The data stored in the receive shift register is transferred to the UCnRX register upon completion of reception of one data frame.

When extension bit operation is enabled (UCnEBE bit of UCnOPT1 register = 1) the 9th data bit is received in bit 8 of the UCnRX register. When the extension bit operation is disabled (UCnEBE bit = 0) the data bits are received in the lower byte of the UCnRX register. The lower byte can be read also by 8-bit access of the UCnRXL register.

During LSB-first reception when the data length has been specified as 7 bits and th extension bit operation is disabled, the receive data is transferred to bits 6 to 0 of the UXnRXL register and the MSB always becomes 0. During MSB-first reception, the receive data is transferred to bits 7 to 1 of the UCnRXL register and the LSB always becomes 0.

When an overrun error (UCnOVE bit = 1) occurs, the receive data at this time is not transferred to the UCnRX and UXnRXL register respectively.

The UCnRX register is read-only, in 16-bit units.

The UCnRXL register is read-only, in 8-bit units.

In addition to reset input, the UCnRX register can be set to 1FFH, and the UCnRXL register can be set to FFH respectively, by clearing the UCnPWR bit of the UCnCTL0 register to 0.

Figure 15-9: UARTCn Receive Data Register (UCnRX, UCnRXL)

After reset: 1FFH R Address: UC0RX FFFFFA06H, UC1RX FFFFFA26H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UCnRX (n = 0, 1)	0	0	0	0	0	0	0									
									UCnRXL							

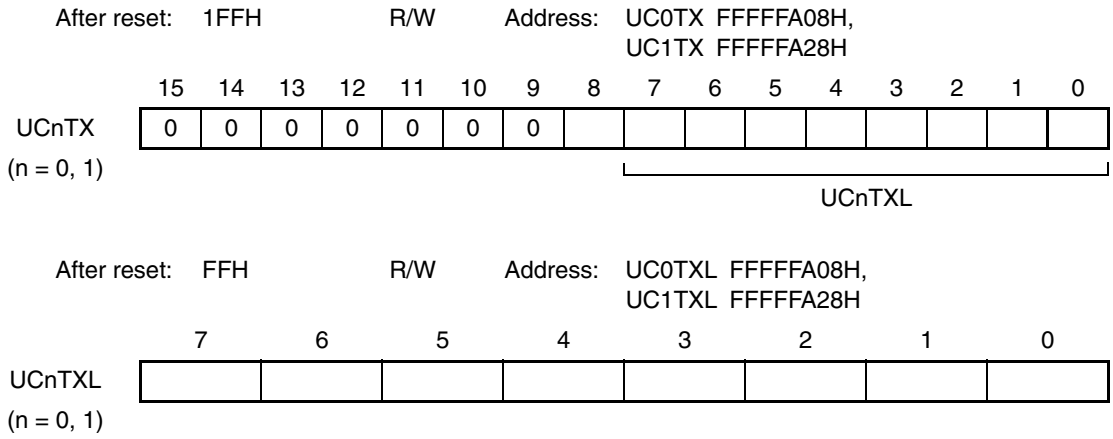
After reset: FFH R Address: UC0RXL FFFFFA06H, UC1RXL FFFFFA26H

	7	6	5	4	3	2	1	0
UCnRXL (n = 0, 1)								

(9) UARTCn transmit data register (UCnTX, UCnTXL)

The UCnTX register is a 16-bit buffer register used to set transmit data. It is overlayed by an 8-bit register UCnTXL on the lower 8 bits. The UCnTXL register is used for setting the transmit data when 7-bit or 8-bit data character length is specified (UCnEBE bit = 0).
The UCnTX register can be read or written in 16-bit units.
The UCnTXL register can be read or written in 8-bit units.
Reset input sets the UCnTX register to 1FFH, and the UCnTXL register to FFH.

Figure 15-10: UARTCn Transmit Data Register (UCnTX, UCnTXL)



15.4 Interrupt Requests

The following three interrupt requests are generated from UARTCn.

- Receive error interrupt (INTUCnRE)
- Reception complete interrupt (INTUCnR)
- Transmission enable interrupt (INTUCnT)

The default priority for these three interrupt requests is highest for the receive error interrupt, followed by the reception complete interrupt, and the transmission enable interrupt.

Table 15-2: Default Priorities of UARTCn Interrupts

Interrupt	Priority
Receive error (INTUCnRE)	High
Reception complete (INTUCnR)	↓
Transmission enable (INTUCnT)	Low

(1) Receive error interrupt (INTUCnRE)

A receive error interrupt is generated when one or more of the three types of receive errors (parity error, framing error, or overrun error) occur. (refer to **15.3 (6) UARTCn status register (UCnSTR)**)

(2) Reception complete interrupt (INTUCnR)

A reception complete interrupt is output when data is shifted into the UARTCn receive shift register and transferred to the UCnRX register in the reception enabled status.
A reception complete interrupt will not be generated when a reception error has occurred.
No reception complete interrupt is generated in the reception disabled status.

(3) Transmission enable interrupt (INTUCnT)

A transmission enable interrupt is generated when transmit data is transferred from the UCnTX register to the UARTCn transmit shift register in the transmission enabled status.

15.5 Operation

15.5.1 Data format

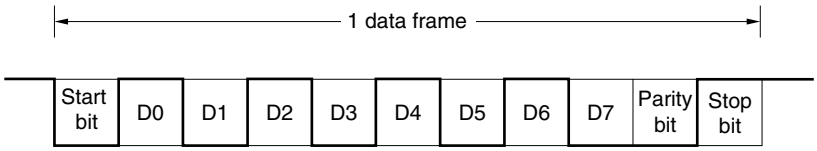
Full-duplex serial data reception and transmission is performed.
As shown in Figure 15-11, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).
Specification of the character bit length within 1 data frame, parity selection, specification of the stop bit length, and specification of MSB/LSB-first transfer are performed using the UCnCTL0 register. UARTCn features additionally the extension bit operation for a ninth transfer data bit, which can be specified in the UCnOPT1 register.
Moreover, control of UART output/inverted output for the TXDCn bit is performed using the UCnTDL bit of the UCnOPT0 register.

- Start bit 1 bit
- Character bits 7 bits/8 bits/9 bits
- Parity bit Even parity/odd parity/0 parity/no parity[Note]
- Stop bit 1 bit/2 bits

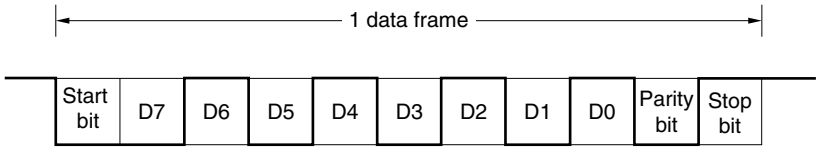
Note: Extension bit operation presumes no parity setting.

Figure 15-11: UARTC Transmit/Receive Data Format (1/2)

(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, TXDCn inversion

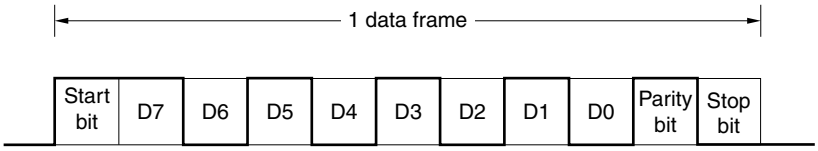
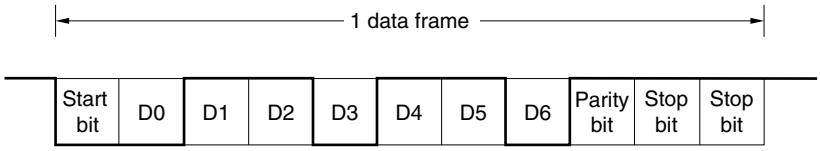
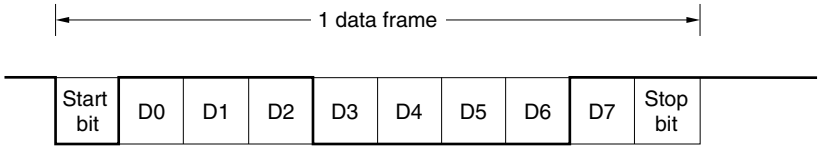


Figure 15-11: UARTC Transmit/Receive Data Format (2/2)

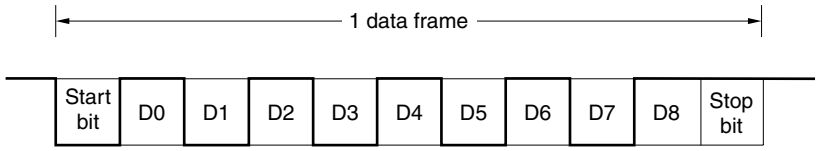
(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36H



(e) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87H



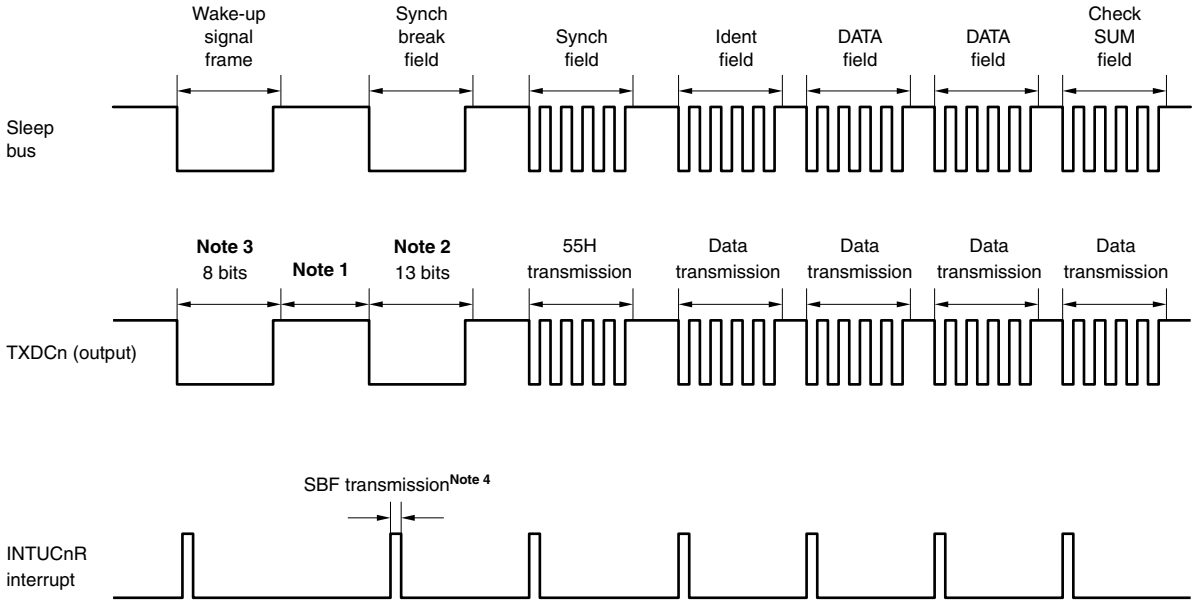
(f) 9-bit data length, LSB first, no parity, 1 stop bit, transfer data: 155H



15.5.2 SBF transmission/reception format

The UARTC has a SBF (Sync Break Field) transmission/reception control function to enable use of the LIN (Local Interconnect Network) function.

Figure 15-12: LIN Transmission Manipulation Outline



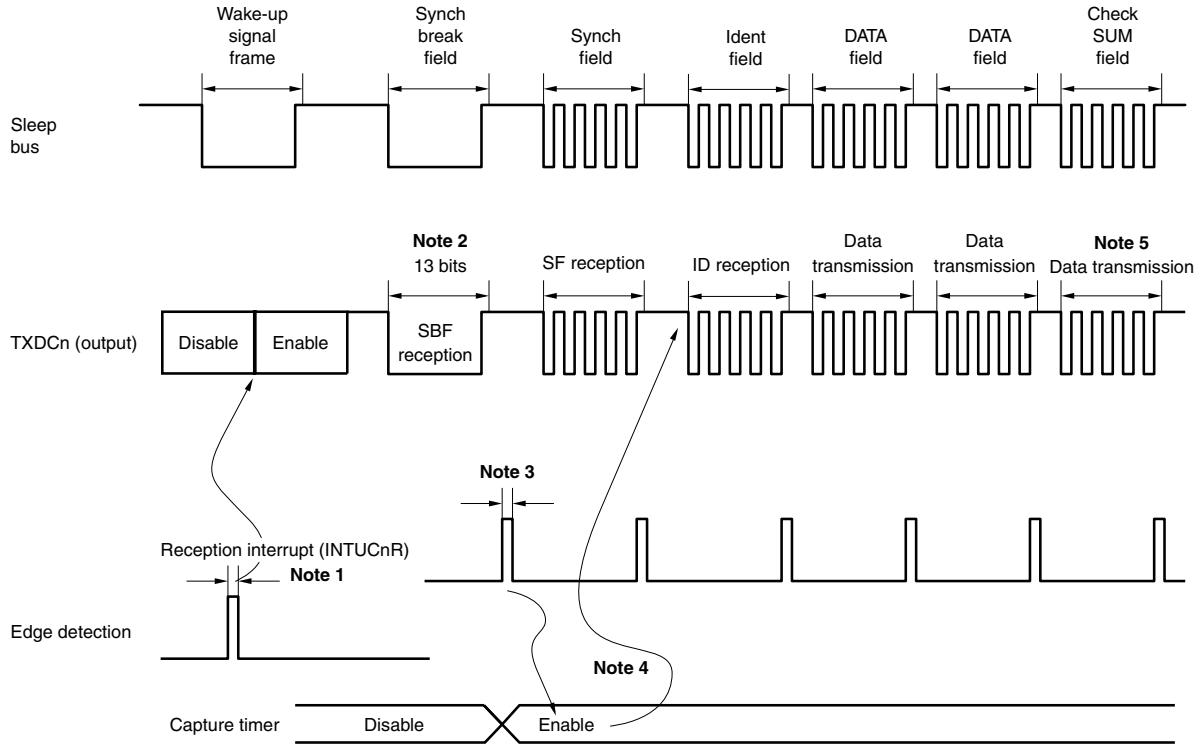
Notes: 1. The interval between each field is controlled by software.

2. SBF output is performed by hardware. The output width is the bit length set by bits UCnSBL2 to UCnSBL0 of the UCnOPT0 register. If even finer output width adjustments are required, such adjustments can be performed using bits UCnBRS7 to UCnBRS0 of the UCnCTLn register.

3. 80H transfer in the 8-bit mode is substituted for the wake-up signal frame.

4. A transmission enable interrupt (INTUCnT) is output at the start of each transmission. The INTUCnT signal is also output at the start of each SBF transmission.

Figure 15-13: LIN Reception Manipulation Outline



Notes: 1. The wakeup signal is sent by the pin edge detector, UARTC is enabled, and the SBF reception mode is set.

2. The receive operation is performed until detection of the stop bit. Upon detection of SBF reception of 11 or more bits, normal SBF reception end is judged, and an interrupt signal is output. Upon detection of SBF reception of less than 11 bits, a SBF reception error is judged, no interrupt signal is output, and the mode returns to the SBF reception mode.

3. If SBF reception ends normally, an interrupt signal is output. The timer is enabled by a SBF reception complete interrupt. Moreover, error detection for the UCnOVE, UCnPE, and UCnFE bits of the UCnSTR register is suppressed and UART communication error detection processing and UARTCn receive shift register and data transfer of the UCnRX register are not performed. The UARTCn receive shift register holds the initial value, FFH.

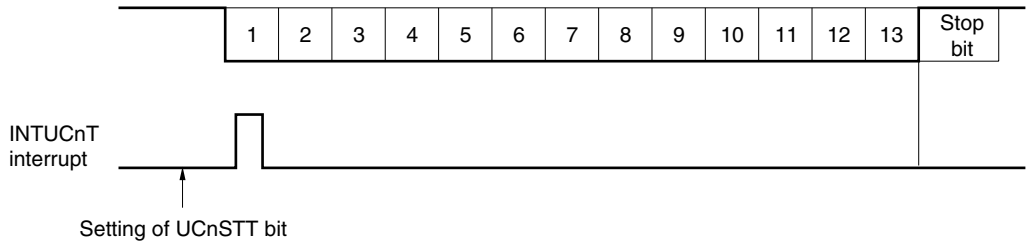
4. The RXDCn pin is connected to TI (capture input) of the timer, the transfer rate is calculated, and the baud rate error is calculated. The value of the UCnCTL2 register obtained by compensating the baud rate error after dropping UARTC enable is set again, causing the status to become the reception status.

5. Check-sum field distinctions are made by software. The UARTC is initialized following CSF reception, and the processing for setting the SBF reception mode again is performed by software.

15.5.3 SBF transmit operation

When the UCnPWR bit = the UCnTXE bit of the UCnCTL0 register = 1, the transmission enabled status is entered, and SBF transmission is started by setting (to 1) the SBF transmission trigger (UCnSTT bit of UCnOPT0 register).
Thereafter, a low-level width of bits 13 to 20 specified by the UCnSLS2 to UCnSLS0 bits of the UCnOPT0 register is output. A transmission enable interrupt (INTUCnT) is generated upon SBF transmission start. Following the end of SBF transmission, the UCnSTT bit is automatically cleared. Thereafter, the UART transmission mode is restored.
Transmission is suspended until the data to be transmitted next is written to the UCnTX register, or until the SBF transmission trigger (UCnSTT bit) is set.

Figure 15-14: SBF Transmission Timing

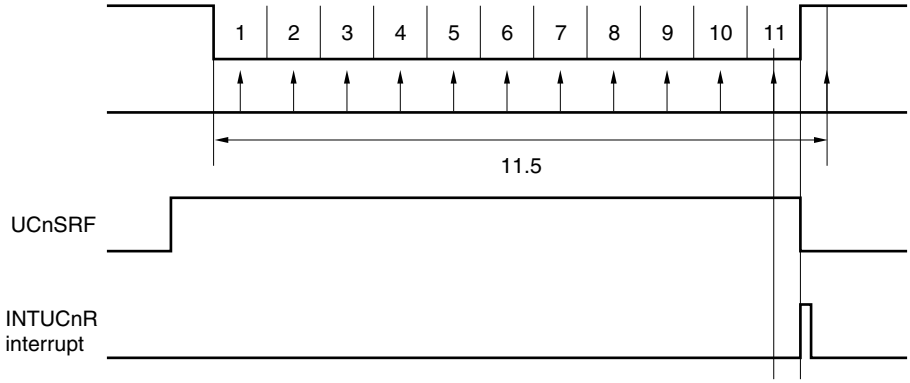


15.5.4 SBF receive operation

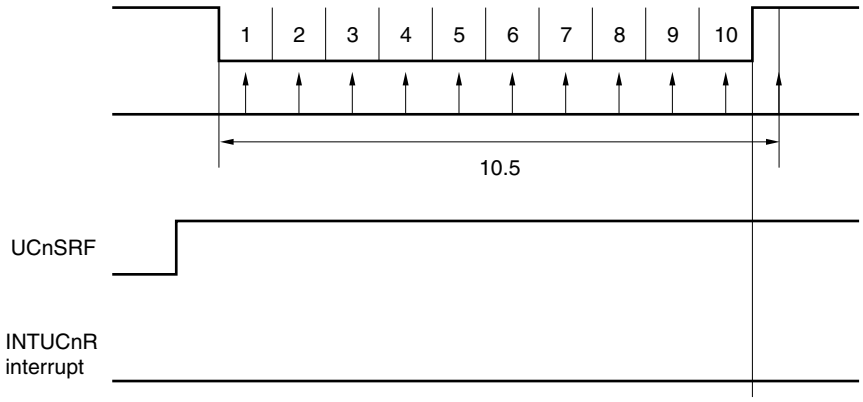
The reception enabled status is achieved by setting the UCnPWR bit of the UCnCTL0 register to 1 and then setting the UCnRX bit of the UCnCTL0 register to 1.
The SBF reception wait status is set by setting the SBF reception trigger (UCnSRT bit of the UCnOPT0 register) to 1.
In the SBF reception wait status, similarly to the UART reception wait status, the RXDCn pin is monitored and start bit detection is performed.
Following detection of the start bit, reception is started and the internal counter counts up according to the set baud rate.
When a stop bit is received, if the SBF width is 11 or more bits, normal processing is judged and a reception complete interrupt (INTUCnR) is output. Error detection for the UCnOVE, UCnPE, and UCnFE bits of the UCnSTR register is suppressed and UART communication error detection processing is not performed. Moreover, UARTCn reception shift register and data transfer of the UCnRX register are not performed and FFH, the initial value, is held. If the SBF width is 10 or fewer bits, reception is terminated as error processing without outputting an interrupt, and the SBF reception mode is returned to. The UCnSRF bit is not cleared at this time.

Figure 15-15: SBF Reception Timing

(a) Normal SBF reception (detection of stop bit in more than 10.5 bits)



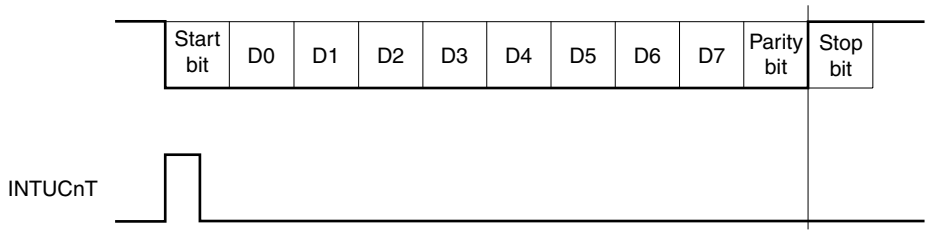
(b) SBF reception error (detection of stop bit in 10.5 or fewer bits)



15.5.5 UART transmit operation

The transmission enabled status is set by setting the UCnTXE bit of the UCnCTL0 register to 1, after UCnPWR bit was set to 1, and transmission is started by writing transmit data to the UCnTX register. The start bit, parity bit, and stop bit are automatically added.
The data in the UCnTX register is transferred to the UARTCn transmit shift register upon the start of the transmit operation.
A transmission enable interrupt (INTUCnT) is generated upon completion of transmission of the data of the UCnTX register to the UARTCn transmit shift register, and thereafter the contents of the UARTCn transmit shift register are output to the TXDCn pin LSB first.
Write of the next transmit data to the UCnTX register is enabled by generating the INTUCnT signal.
Continuous transmission is enabled by writing the data to be transmitted next to the UCnTX register during transfer.

Figure 15-16: UART Transmission



Remark: If new data is written to the UCnTX register due to a transmission enable interrupt (INTUCnT) before the complete frame has been transferred, the next transmission enable interrupt occurs at that time the stop bit begins.

15.5.6 Continuous transmit operation

UARTCn can write the next transmit data to the UCnTX register when the UARTCn transmit shift register starts the shift operation. The transfer timing of the UARTCn transmit shift register can be judged from the transmission enable interrupt (INTUCnT). Transmission can be performed without interruption even during interrupt processing following the transmission of 1 data frame via the INTUCnT signal, and an efficient communication rate can thus be achieved.
During continuous transmission, overrun (the completion of the next transmission before the first transmission completion processing has been executed) may occur.
An overrun can be detected by incorporating a program that can count the number of transmit data and by referencing transfer status flag (UCnTSF bit of UCnSTR register).

Caution: During continuous transmission execution, perform initialization after checking that the UCnTSF bit is 0. The transmit data cannot be guaranteed when initialization is performed when the UCnTSF bit is 1.

Figure 15-17: Continuous Transmission Processing Flow

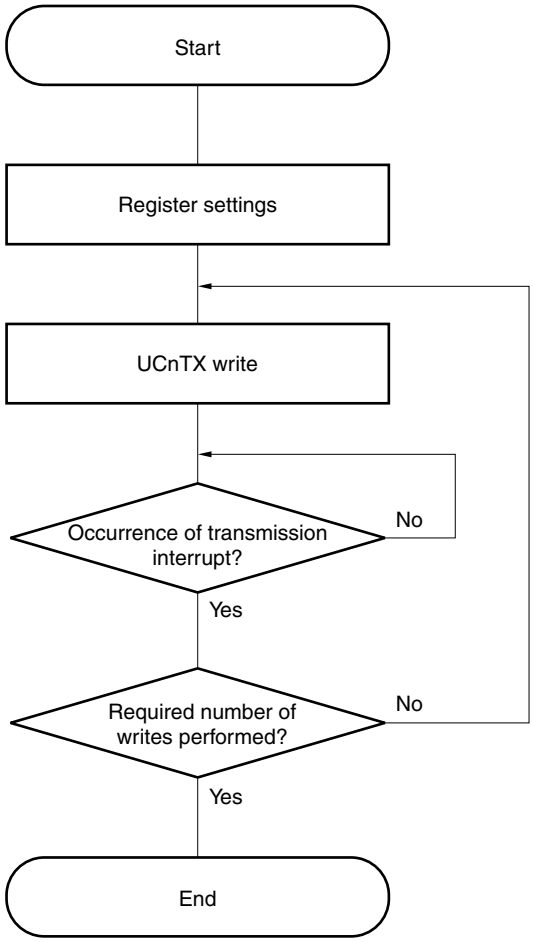
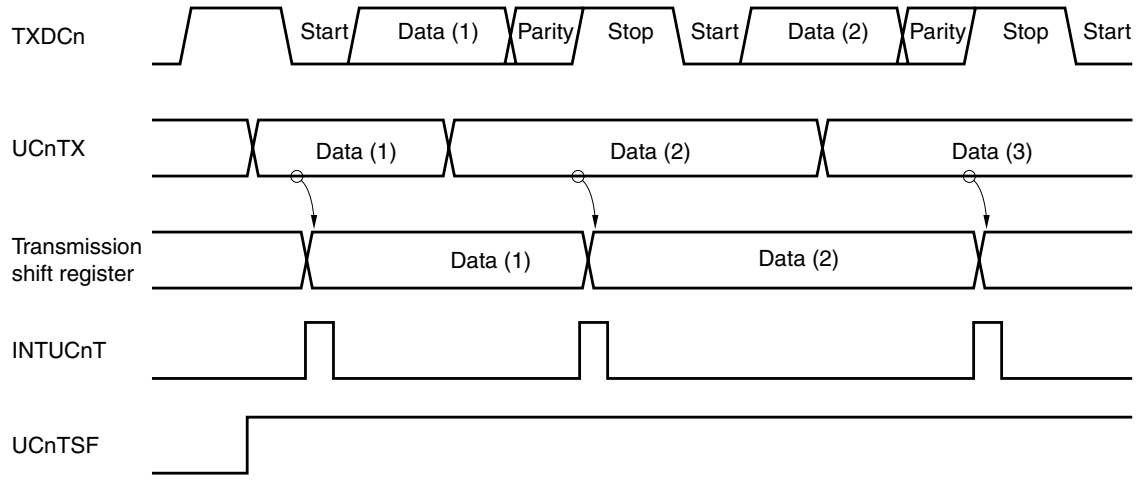
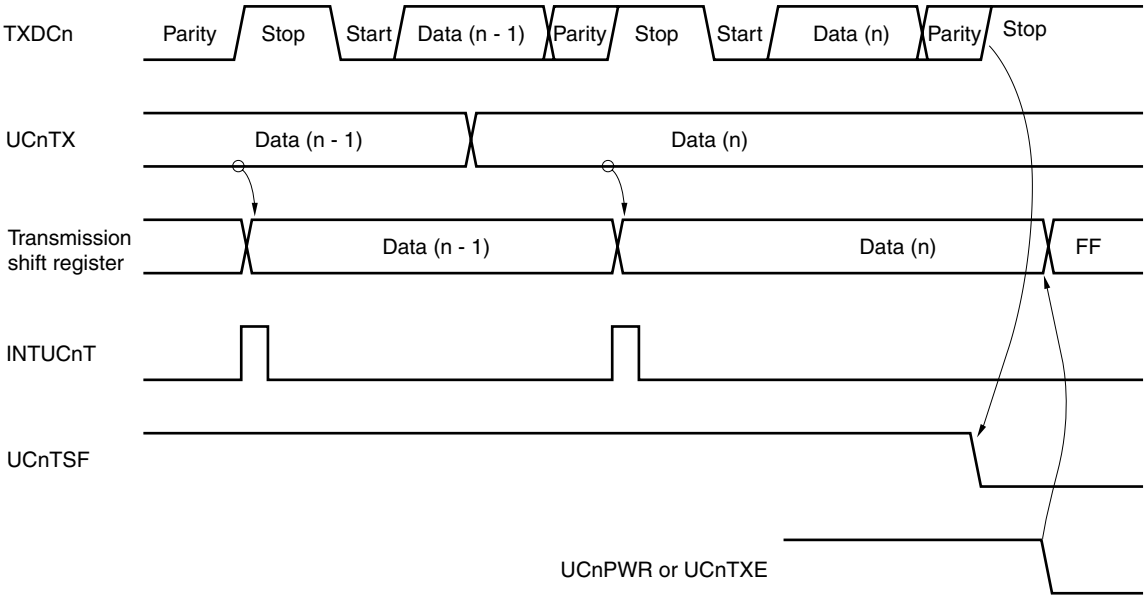


Figure 15-18: Continuous Transfer Operation Timing

(a) Transmission start



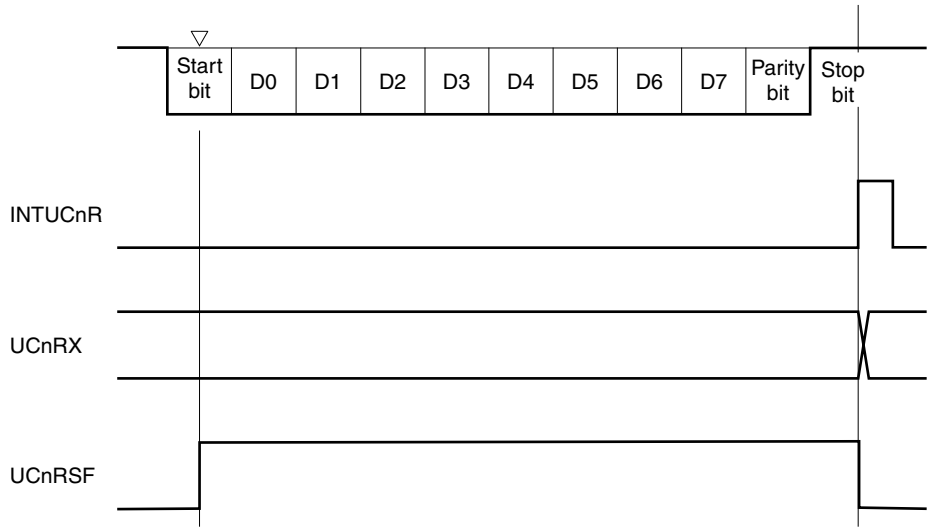
(b) Transmission end



15.5.7 UART receive operation

The reception wait status is set by setting the UCnPWR bit of the UCnCTL0 register to 1 and then setting the UCnRX bit of the UCnCTL0 register to 1. In the reception wait status, the RXDCn pin is monitored and start bit detection is performed.
Start bit detection is performed using a two-step detection routine.
First, an 8-bit counter starts upon detection of the falling edge of the RXDCn pin. When the 8-bit counter has counted the UCnCTL2 register setting value, the level of the RXDCn pin is monitored again (corresponds to the ∇ mark in Figure 15-19). If the RXDCn pin is low level at this time too, a start bit is recognized. After a start bit has been recognized, the receive operation starts, and serial data is saved to the UARTCn receive shift register according to the set baud rate. Additionally the UCnRSF flag of UCnSTR1 register is set (1) to indicate the receive operation status.
When the reception complete interrupt (INTUCnR) is output upon reception of the stop bit, the data of the UARTCn receive shift register is written to the UCnRX register, and the UCnRSF flag is cleared (0) simultaneously. However, if an overrun error occurs (UCnOVE bit = 1), the receive data at this time is not written to the UCnRX register, and a reception error interrupt (INTUCnRE) is output.
Even if a parity error (UCnPE bit = 1) or a framing error (UCnFE bit = 1) occurs during reception, reception continues until the stop bit reception position, but a reception error interrupt (INTUCnRE) is output following reception completion.

Figure 15-19: UART Reception Timing



Cautions: 1. Be sure to read the UCnRX register even when a reception error occurs. If the UCnRX register is not read, an overrun error occurs during reception of the next data, and reception errors continue occurring indefinitely.

2. The operation during reception is performed assuming that there is only one stop bit. A second stop bit is ignored.

15.5.8 Receive error

Errors during a receive operation are of three types: parity errors, framing errors, and overrun errors. A data reception result error flag is set to the UCnSTR register and a reception error interrupt (INTUCnRE) is output.
During reception error interrupt processing, it is possible to ascertain which error occurred during reception by reading the contents of the UCnSTR register.
The reception error flag is cleared by writing 0 to it.

Table 15-3: Reception Error Causes

Error Flag	Reception Error	Cause
UCnPE	Parity error	Received parity bit does not match the setting
UCnFE	Framing error	Stop bit not detected
UCnOVE	Overrun error	Reception of next data completed before data was read from receive buffer

Cautions: 1. In case of a reception error the reception complete interrupt (INTUCnR) is not generated. Instead of this a reception error interrupt (INTUCnRE) can be received.

2. Be sure to read the UCnRX register even when a reception error occurs. If the UCnRX register is not read, an overrun error occurs during reception of the next data, and reception errors continue occurring indefinitely.

15.5.9 Parity types and operations

Caution: When using the LIN function, fix the UCnPS1 and UCnPS0 bits of the UCnCTL0 register to 00.

The parity bit is used to detect bit errors in the communication data. Normally the same parity is used on the transmission side and the reception side.
In the case of even parity and odd parity, it is possible to detect “1” bit errors (odd count). In the case of 0 parity and no parity, errors cannot be detected.

(1) Even parity

(a) During transmission

The number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so as to be an even number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 1
- Even number of bits whose value is “1” among transmit data: 0

(b) During reception

The number of bits whose value is “1” among the reception data, including the parity bit, is counted, and if it is an odd number, a parity error is output.

(2) Odd parity

(a) During transmission

Opposite to even parity, the number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so that it is an odd number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 0
- Even number of bits whose value is “1” among transmit data: 1

(b) During reception

The number of bits whose value is “1” among the receive data, including the parity bit, is counted, and if it is an even number, a parity error is output.

(3) 0 parity

During transmission, the parity bit is always made 0, regardless of the transmit data.
During reception, parity bit check is not performed. Therefore, no parity error is generated, regardless of whether the parity bit is 0 or 1.

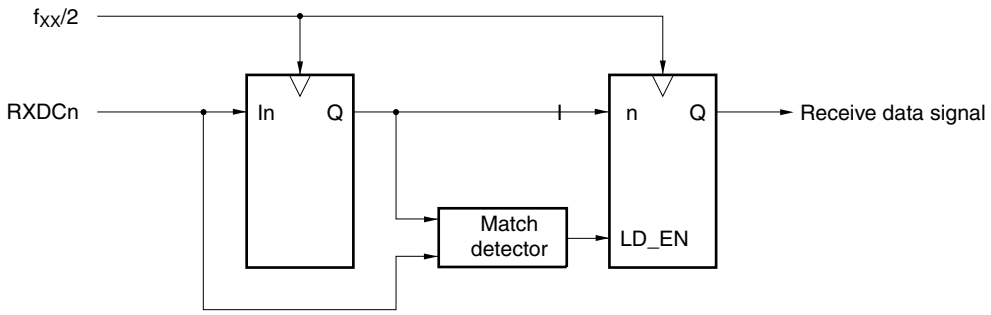
(4) No parity

No parity bit is added to the transmit data.
Reception is performed assuming that there is no parity bit. No parity error occurs since there is no parity bit.

15.5.10 Receive data noise filter

This filter performs the RXDCn pin sampling using the internal system clock ($f_{XX}/2$).
When the same sampling value is read twice, the match detector output changes and sampling as the input data is performed.
Moreover, since the circuit is as shown in Figure 15-20, the processing that goes on within the receive operation is delayed by 2 clocks in relation to the external signal status.

Figure 15-20: Noise Filter Circuit

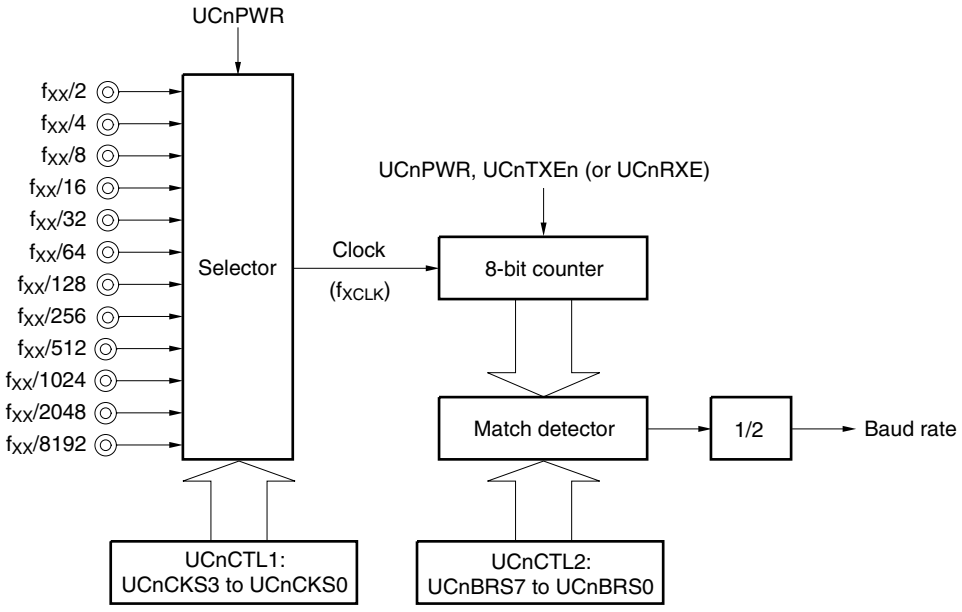


15.6 Dedicated Baud Rate Generator

15.6.1 Baud rate generator configuration

The dedicated baud rate generator consists of a source clock selector block and an 8-bit programmable counter, and generates a serial clock during transmission and reception with UARTCn. Regarding the serial clock, a dedicated baud rate generator output can be selected for each channel.
There is an 8-bit counter for transmission and another one for reception.

Figure 15-21: Configuration of Baud Rate Generator



Remarks: **1.** n = 0, 1

2. f_{XX}: Internal system clock

(1) Base clock (Clock)

When the UCnPWR bit of the UCnCTL0 register is 1, the clock selected by bits UCnCKS3 to UCnCKS0 of the UCnCTL1 register is supplied to the 8-bit counter. This clock is called the base clock (Clock) and its frequency is called f_{XCLK}. When the UCnPWR bit = 0, the clock is fixed to the low level.

(2) Serial clock generation

A serial clock can be generated by setting the UCnCTL1 register and the UCnCTL2 register (n = 0, 1).
The base clock is selected by UCnCKS3 to UCnCKS0 bits of the UCnCTL1 register.
The frequency division value for the 8-bit counter can be set using bits UCnBRS7 to UCnBRS0 of the UCnCTL2 register.

15.6.2 Baud rate

The baud rate is obtained by the following equation.

$$\text{Baud rate} = \frac{f_{XCLK}}{2 \times k} \text{ [bps]}$$

f_{XCLK} = Frequency of base clock (Clock) selected by bits UCnCKS3 to UCnCKS0 of UCnCTL1 register
k = Value set using bits UCnBRS7 to UCnBRS0 of UCnCTL2 register (k = 4, 5, 6,..., 255)

15.6.3 Baud rate error

The baud rate error is obtained by the following equation.

$$\text{Error} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100 \quad [\%]$$

Cautions: 1. The baud rate error during transmission must be within the error tolerance on the receiving side.

2. The baud rate error during reception must satisfy the range indicated in section 15.6.5 ”Allowable baud rate range during reception” on page 626.

Example

Base clock (f_{XCLK}) frequency = 16 MHz = 16,000,000 Hz
Setting value of bits UCnBRS7 to UCnBRS0 of UCnCTL2 register = 00110100B (k = 52)
Target baud rate = 153,600

Baud rate = 16,000,000/ (2 × 52) = 153,846 [bps]

Error = (153,846/153,600 - 1) × 100
= 0.160 [%]

15.6.4 Baud rate setting example

Table 15-4: Baud Rate Generator Setting Data

Baud Rate [bps]	f_{XX} = 64 MHz		
	UCnCTL1	UCnCTL2	Error [%]
50	0BH	4EH	0.16
300	09H	68H	0.16
600	08H	68H	0.16
1200	07H	68H	0.16
2400	06H	68H	0.16
4800	05H	68H	0.16
9600	04H	68H	0.16
10400	04H	60H	0.16
19200	03H	68H	0.16
31250	02H	80H	0.00
38400	02H	68H	0.16
56000	01H	8FH	-0.10
76800	01H	68H	0.16
125000	01H	40H	0.00
153600	01H	34H	0.16
250000	01H	20H	0.00
312500	00H	33H	0.39
1000000	00H	10H	0.00
2000000	00H	08H	0.00

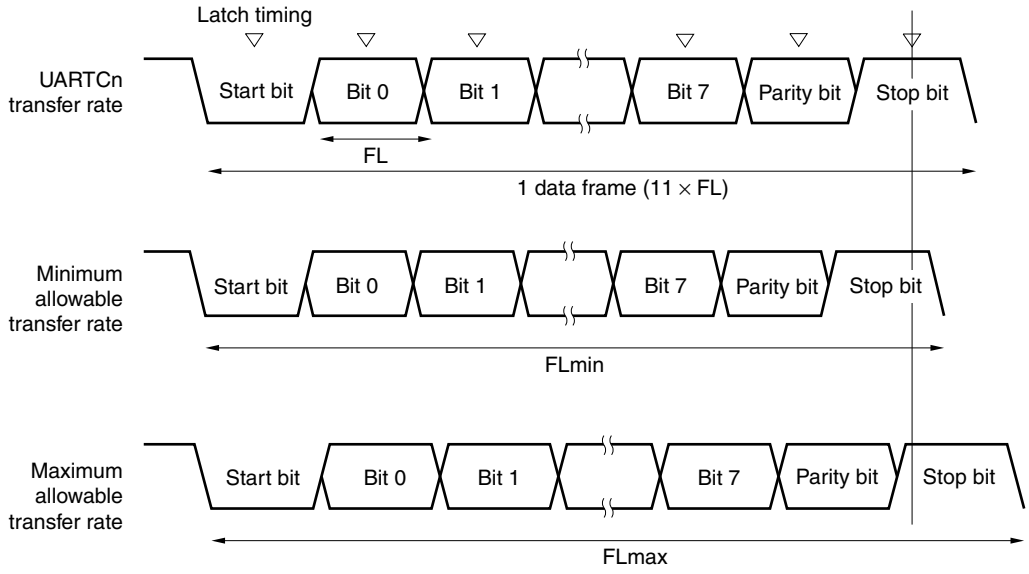
Remark: f_{XX}: Internal system clock
Error: Baud rate error

15.6.5 Allowable baud rate range during reception

The baud rate error range at the destination that is allowable during reception is shown below.

Caution: The baud rate error during reception must be set within the allowable error range using the following equation.

Figure 15-22: Allowable Baud Rate Range During Reception



Remark: n = 0 to 2

As shown in Figure 15-22, the receive data latch timing is determined by the counter set using the UCnCTL2 register following start bit detection. The transmit data can be normally received if up to the last data (stop bit) can be received in time for this latch timing.
When this is applied to 11-bit reception, the following results in terms of logic.

$$FL = (BR)^{-1}$$

BR: UARTCn baud rate (n = 0, 1)
k: UCnCTL2 setting value (n = 0, 1)
FL: 1-bit data length
Latch timing margin: 2 clocks

Minimum allowable transfer rate:

$$FLmin = 11 \times FL - \frac{k - 2}{2k} \times FL = \frac{21k + 2}{2k} FL$$

Therefore, the maximum baud rate that can be received by the destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} BR$$

Similarly, obtaining the following maximum allowable transfer rate yields the following.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FLmax = \frac{21k - 2}{20k} FL \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} BR$$

Obtaining the allowable baud rate error for UARTCn and the destination from the above-described equations for obtaining the minimum and maximum baud rate values yields the following.

Table 15-5: Maximum/Minimum Allowable Baud Rate Error

Divide Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
8	+3.53%	-3.61%
20	+4.26%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

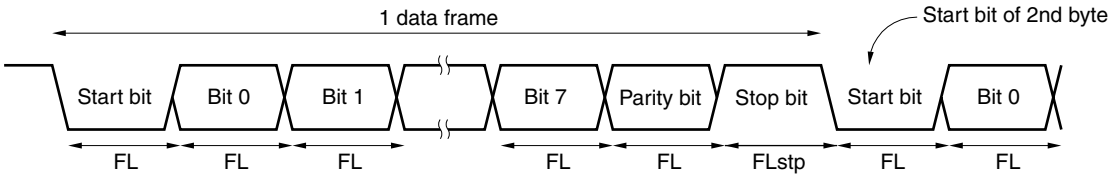
Remarks: 1. The reception accuracy depends on the bit count in 1 frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the accuracy is.

2. k: UCnCTL2 setting value (n = 0, 1)

15.6.6 Baud rate during continuous transmission

During continuous transmission, the transfer rate from the stop bit to the next start bit is usually 2 clocks longer. However, timing initialization is performed through start bit detection by the receiving side, so this has no influence on the transfer result.

Figure 15-23: Transfer Rate During Continuous Transfer



Assuming 1 bit data length: FL,
stop bit length: FLstp,
and base clock frequency: f_{XCLK},
we obtain the following equation.

$$FLstp = FL + 2/f_{XCLK}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + 2/f_{XCLK}$$

Chapter 16 Clocked Serial Interface B (CSIB)

16.1 Features

- Transfer rate: Maximum 8 Mbps
- Master mode and slave mode selectable
- Serial clock and data phase switchable
- Transmission data length: 8 to 16 bits (selectable in 1-bit units)
- Transfer data MSB-first/LSB-first switchable
- Transmission mode, reception mode, and transmission/reception mode selectable
- 3-wire serial interface
  - SOBn: Serial data output
  - SIBn: Serial data input
  - $\overline{\text{SCK}}$Bn: Serial clock output
- Slave select function supported
  - $\overline{\text{SS}}$Bn: Serial slave select input
- Interrupt request signals × 3
  - Reception error interrupt (INTCBnRE)
  - Reception complete interrupt (INTCBnR)
  - Transmission enable interrupt (INTCBnT)

Remark: n = 0, 1

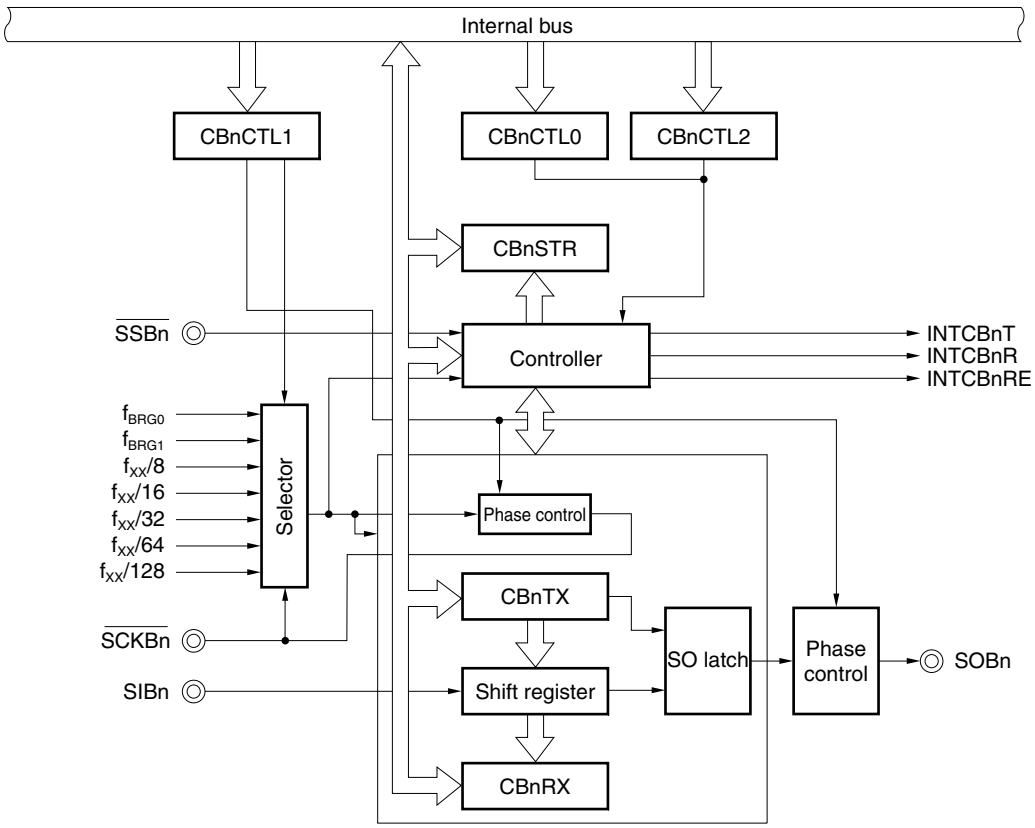
16.2 Configuration

CSIB includes the following hardware.

Table 16-1: CSIBn Configuration

Item	Configuration
Registers	CSIBn receive data register (CBnRX)
	CSIBn transmit data register (CBnTX)
Control registers	CSIBn control register 0 (CBnCTL0)
	CSIBn control register 1 (CBnCTL1)
	CSIBn control register 2 (CBnCTL2)
	CSIBn status register (CBnSTR)

Figure 16-1: Block Diagram of CSIBn



Remarks: 1. n = 0, 1

2. f_{XX}: Internal system clock
f_{BRG0}: Clock from BRG0
f_{BRG1}: Clock from BRG1

(1) CSIBn receive data register (CBnRX, CBnRXL)

The CBnRX register is a 16-bit buffer register that holds receive data. It is overlayed by an 8-bit register CBnRXL on the lower 8 bits, which is used when the transfer data length is 8 bits.
The receive operation is started by reading the CBnRX or CBnRXL registers during reception enabled status.
The CBnRX register is read-only, in 16-bit units.
The CBnRXL register is read-only, in 8-bit units.
Reset input clears the CBnRX register to 0000H, and the CBnRXL register to 00H accordingly.
In addition to reset input, the CBnRX or CBnRXL registers can be initialized by clearing (0) the CBnPWR bit of the CBnCTL0 register.

Figure 16-2: CSIBn Receive Data Register (CBnRX, CBnRXL)

(2) CSIB transmit data register (CBnTX)

The CBnTX register is a 16-bit buffer register used to write the CSIB transfer data. It is overlayed by an 8-bit register CBnTXL on the lower 8 bits, which is used when the transfer data length is 8 bits.

The transmit operation is started by writing data to the CBnTX or CBnTXL registers during transmission enabled status.

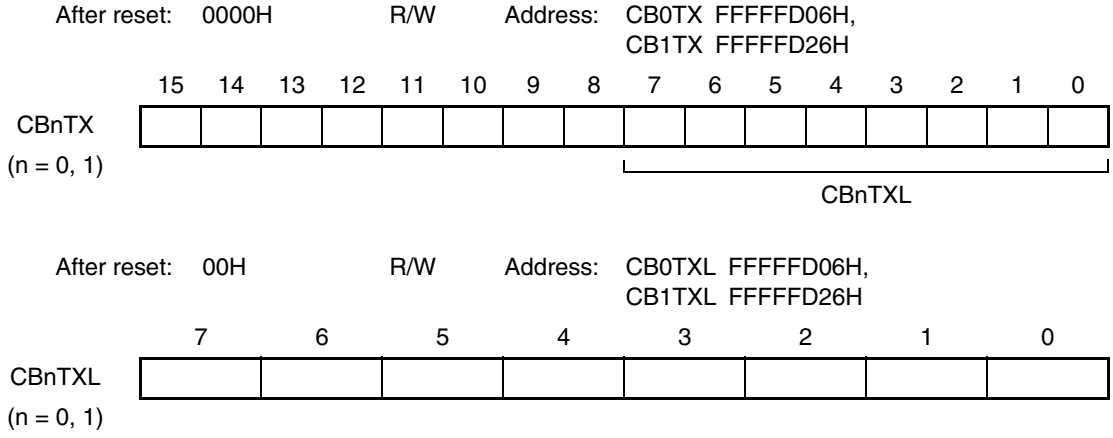
The CBnTX register can be read or written in 16-bit units.

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

Reset input clears the CBnTX register to 0000H, and the CBnRXL register to 00H accordingly.

In addition to reset input, the CBnTX and CBnTXL registers can be initialized by clearing (to 0) the CBnPWR bit of the CBnCTL0 register.

Figure 16-3: CSIBn Transmit Data Register (CBnTX, CBnTXL)



16.3 Control Registers

The following registers are used to control CSIB.

- CSIBn control register 0 (CBnCTL0)
- CSIBn control register 1 (CBnCTL1)
- CSIBn control register 2 (CBnCTL2)
- CSIBn status register (CBnSTR)

(1) CSIBn control register 0 (CBnCTL0)

The CBnCTL0 register is a register that controls the CSIB serial transfer operation.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 01H.

Caution: Be sure to set bit 3 to 0 when written to the CBnCTL0 register.

Figure 16-4: CSIBn Control Register 0 (CBnCTL0) (1/2)

After reset: 01H R/W Address: CB0CTL0 FFFFFD00H, CB1CTL0 FFFFFD20H

CBnCTL0 (n = 0, 1)

7	6	5	4	3	2	1	0
CBnPWR	CBnTXE Note	CBnRXE Note	CBnDIR Note	0	CBnSSE Note	CBnTMS Note	CBnSCE

CBnPWR	CSIBn Operation Control
0	Stops clock operation and reset the internal circuit
1	Enables operating clock operation
The CBnPWR bit controls the CSIB operating clock and resets the internal circuit.	

CBnTXE Note	Transmission Operation Enable
0	Stops transmission operation
1	Enables transmission operation
The SOBn serial output pin is fixed to low level and communication is stopped by clearing the CBnTXE bit to 0.	

CBnRXE Note	Reception Operation Enable
0	Stops reception operation
1	Enables reception operation
When the CBnRXE bit is cleared to 0, no reception complete interrupt is output even when the prescribed data is transferred in order to stop the receive operation, and the CBnRX register is not updated.	

Note: Rewrite is possible only when the CBnPWR bit = 0. However, CBnPWR bit = 1 can also be set at the same time.

Figure 16-4: CSIBn Control Register 0 (CBnCTL0) (2/2)

CBnDIR[Note]	Transfer Direction Selection
0	MSB-first transfer
1	LSB-first transfer

CBnSSE[Note]	Slave Selection Operation Enable
0	Slave selection function disabled
1	Slave selection function enabled
When the CSIBn serves as slave, it executes transmission/reception in synchronization with the clock only when a low level is input to the $\overline{SSBn}$ pin.	

CBnTMS[Note]	Transfer Mode Selection
0	Single transfer mode
1	Continuous transfer mode
When the CBnTMS bit = 0, the single transfer mode is entered, so continuous transmission/continuous reception are not supported. Even in the case of transmission only, an interrupt is output upon completion of reception transfer.	

CBnSCE	Serial Clock Enable
0	Clock output stopped
1	Clock output enabled
The transfer clock is stopped after the last data in the master reception mode. Clear (0) the CBnSCE bit prior to when the last data is read in the single transfer mode, and 1 clock before the completion of reception of the last data in the continuous transfer mode. The transfer clock can be output by setting the CBnSCE bit to 1 again after the last data has been read.	

Note: Rewrite is possible only when the CBnPWR bit = 0. However, CBnPWR bit = 1 can also be set at the same time.

(2) CSIBn control register 1 (CBnCTL1)

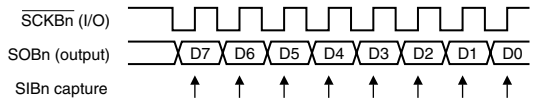
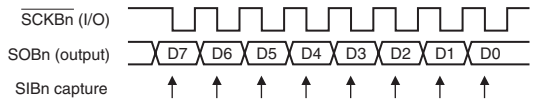
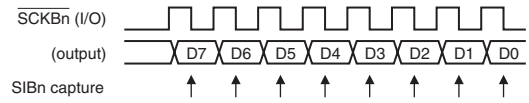
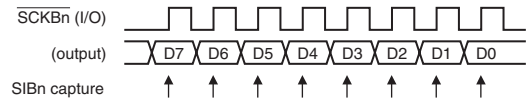
The CBnCTL1 register is an 8-bit register that controls the CSIB serial transfer operation.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Caution: The CBnCTL1 register can be rewritten when the CBnPWR bit of the CBnCTL0 register is 0 or when both the CBnTXE and CBnRXE bits are 0.

Figure 16-5: CSIBn Control Register 1 (CBnCTL1)

After reset: 00H R/W Address: CB0CTL1 FFFFFD01H, CB1CTL1 FFFFFD21H

	7	6	5	4	3	2	1	0
CBnCTL1 (n = 0, 1)	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

CBnCKP	CBnDAP	Specification of Data Transmission/Reception Timing in Relation to Clock Phase
0	0	
0	1	
1	0	
1	1	

CBnCKS2	CBnCKS1	CBnCKS0	Base Clock (f_{XCCLK})	Mode
0	0	0	f_{BRG0} **Note**	Master mode
0	0	1	f_{BRG1} **Note**	Master mode
0	1	0	$f_{XX}/8$	Master mode
0	1	1	$f_{XX}/16$	Master mode
1	0	0	$f_{XX}/32$	Master mode
1	0	1	$f_{XX}/64$	Master mode
1	1	0	$f_{XX}/128$	Master mode
1	1	1	External clock ($\overline{SCKBn}$)	Slave mode

Note: For details on the baud rate generator refer to **16.7 "Baud Rate Generator" on page 657**.

(3) CSIBn control register 2 (CBnCTL2)

The CBnCTL2 register is an 8-bit register that controls the number of CSIB serial transfer bits.
This register can be read or written in 8-bit units.
Reset input clears this register to 00H.

Caution: The CBnCTL2 register can be rewritten only when the CBnPWR bit of the CBnCTL0 register is 0 or when both the CB0TXE and CB0RXE bits are 0.

Figure 16-6: CSIBn Control Register 2 (CBnCTL2)

After reset: 00H R/W Address: CB0CTL2 FFFFFD02H, CB1CTL2 FFFFFD22H

	7	6	5	4	3	2	1	0
CBnCTL2 (n = 0, 1)	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

CBnCL3	CBnCL2	CBnCL1	CBnCL0	Serial Register Bit Length
0	0	0	0	8 bits
0	0	0	1	9 bits
0	0	1	0	10 bits
0	0	1	1	11 bits
0	1	0	0	12 bits
0	1	0	1	13 bits
0	1	1	0	14 bits
0	1	1	1	15 bits
1	×	×	×	16 bits

Caution: **If the number of transfer bits is other than 8 or 16, prepare and use data stuffed from the LSB of the CBnTX and CBnRX registers.**

Remark: ×: don't care

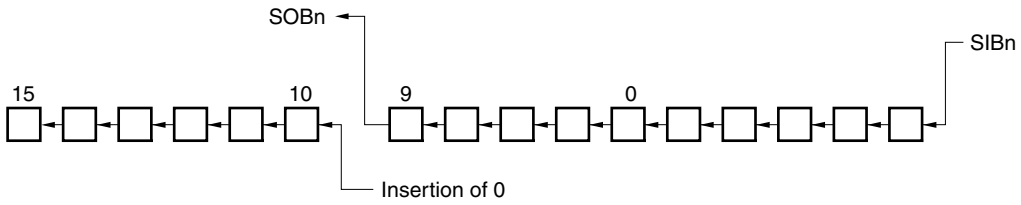
(a) Transfer data length function

The CSIB transfer data length can be set in 1-bit units between 8 and 16 bits using bits CBnCL3 to CBnCL0 of the CBnCTL2 register.

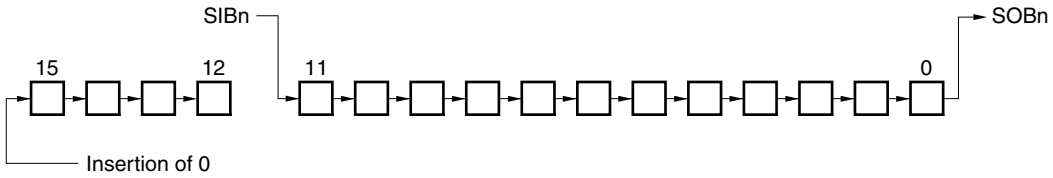
When the transfer bit length is set to a value other than 16 bits, set the data to the CBnTX or CBnRX register starting from the LSB, regardless of whether the transfer start bit is the MSB or LSB. Any data can be set for the higher bits that are not used, but the receive data becomes 0 following serial transfer.

Figure 16-7: Effect of Transfer Data Length Setting

(a) Transfer bit length = 10 bits, MSB first



(b) Transfer bit length = 12 bits, LSB first



(4) CSIBn status register (CBnSTR)

The CBnSTR register is an 8-bit register that displays the CSIB status.
This register can be read or written in 8-bit or 1-bit units, but the CBnSTF flag is a read-only.
Reset input clears this register to 00H.
In addition to reset input, the CBnSTR register can be initialized by clearing (0) the CBnPWR bit of the CBnCTL0 register.

Figure 16-8: CSIBn Status Register (CBnSTR)

After reset: 00H R/W Address: CB0CTL0 FFFFFD03H, CB1CTL0 FFFFFD23H

	7	6	5	4	3	2	1	0
CBnSTR	CBnTSF	0	0	0	0	0	0	CBnOVE

(n = 0, 1)

CBnTSF	CSIBn Operation Control
0	Idle status
1	Operating status
During transmission, this register is set (1) when data is prepared in the CBnTX register, and during reception, it is set (1) when a dummy read of the CBnRX register is performed. The clear timing is after the edge of the last clock.	

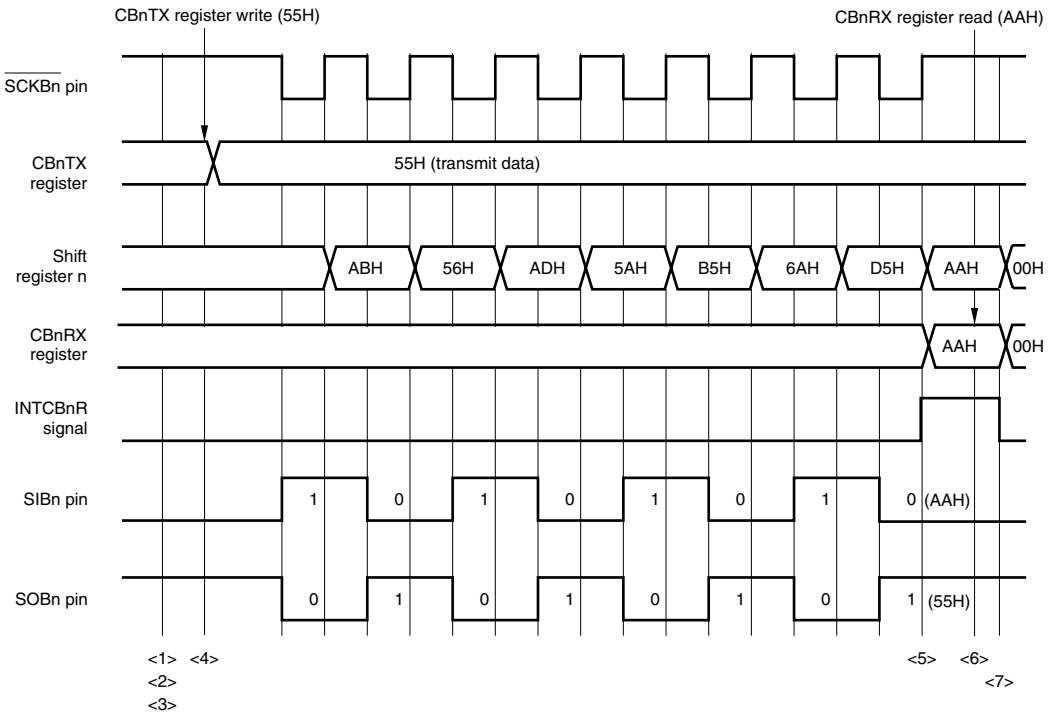
CBnOVE	Overrun Error Flag
0	No overrun
1	Overrun
• An overrun error occurs when the next reception starts without performing a CPU read of the value of the CBnRX register upon completion of the receive operation. In this case the CBnOVE flag displays the overrun error occurrence status, and a reception error interrupt (INTCBnRE) is generated. • The CBnOVE flag is cleared by writing 0 to it. It cannot be set even by writing 1 to it.	

16.4 Operation

16.4.1 Single transfer mode (master mode, transmission/reception mode)

Figure 16-9: Single Transfer Mode (Master Mode, Transmission/Reception Mode)

MSB First (CBnDIR Bit of CBnCTL0 Register = 0),
CBnCKP Bit of the CBnCTL1 Register = 0,
CBnDAP Bit of the CBnCTL1 Register = 0,
Transfer Data Length = 8 Bits (CSnCL3 to CBnCL0 Bits of CBnCTL2 Register = 0000B)



<1> Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
<2> Set the CBnTXE and CBnRXE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit of the CBnCTL0 register, to set the transmission/ reception enable status.
<3> Set the CBnPWR bit of the CBnCTL0 register to 1 to enable CSIB operating clock supply.
<4> Write transfer data to the CBnTX register (transmission start).
<5> The reception complete interrupt (INTCBnR) is output, notifying the CPU that reading the CBnRX (CBnRXL) register is possible.
<6> Read the CBnRX register before clearing the CBnPWR bit to 0.
<7> Check that the CBnTSF bit of the CBnSTR register is 0 and clear the CBnPWR bit to 0 to stop clock supply to CSIB (end of transmission/reception).

To continue transfer, repeat steps <4> to <6> before <7>.

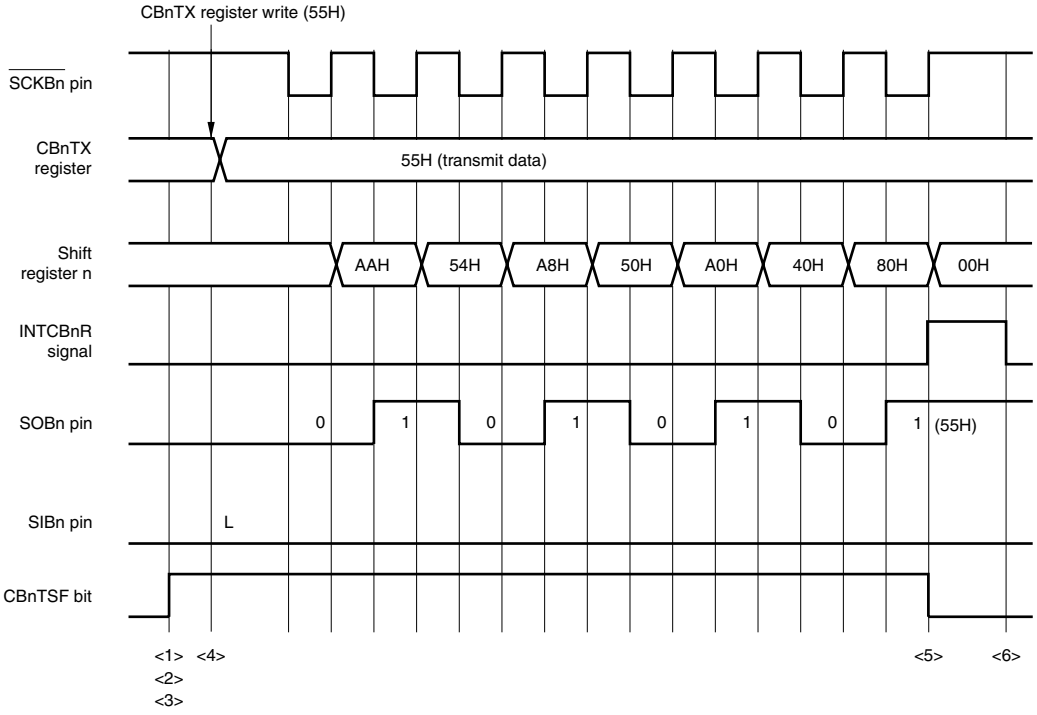
Remarks: 1. The processing of steps <2> and <3> can be set simultaneously.

2. n = 0, 1

16.4.2 Single transfer mode (master mode, transmission mode)

Figure 16-10: Single Transfer Mode (Master Mode, Transmission Mode)

MSB First (CBnDIR Bit of CBnCTL0 Register = 0),
CBnCKP Bit of the CBnCTL1 Register = 0,
CBnDAP Bit of the CBnCTL1 Register = 0,
Transfer Data Length = 8 Bits (CSnCL3 to CBnCL0 Bits of CBnCTL2 Register = 0000B)



<1> Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
<2> Set the CBnTXE bit of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit of the CBnCTL0 register, to set the transmission/reception enable status.
<3> Set the CBnPWR bit of the CBnCTL0 register to 1 to enable CSIB operating clock supply.
<4> Write transfer data to the CBnTX register (transmission start).
<5> The reception complete interrupt (INTCBnR) is output, notifying the CPU that writing the CBnTX (CBnTXL) register is possible.
<6> Check that the CBnTSF bit of the CBnSTR register is 0 and clear the CBnPWR bit to 0 to stop clock supply to CSIB (end of transmission/reception).

To continue transfer, repeat steps <4> and <5> before <6>.

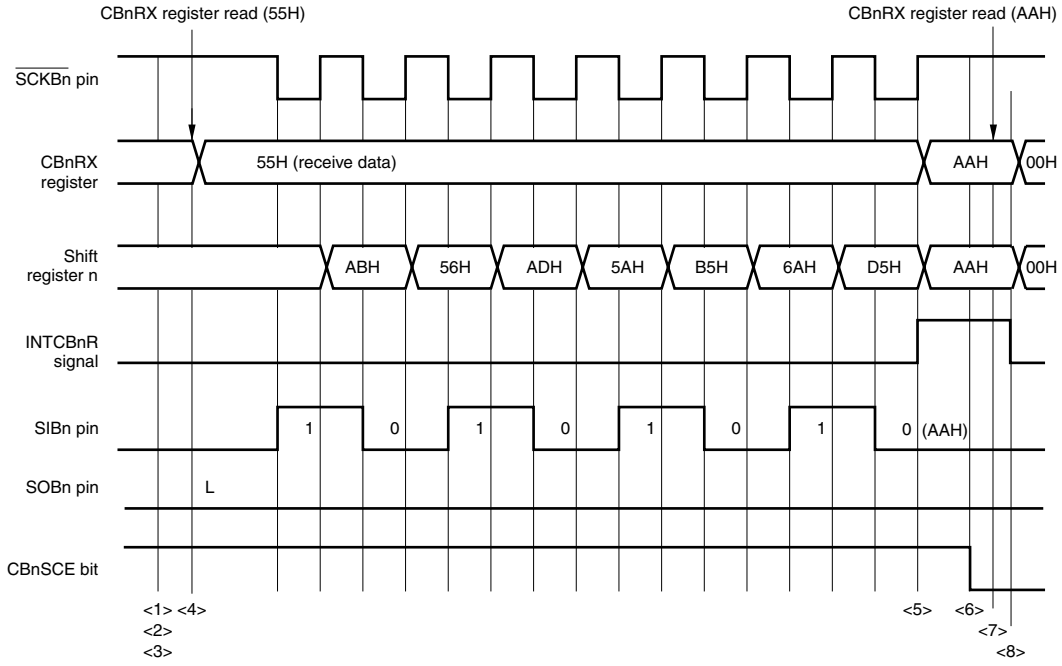
Remarks: 1. The processing of steps <2> and <3> can be set simultaneously.

2. n = 0, 1

16.4.3 Single transfer mode (master mode, reception mode)

Figure 16-11: Single Transfer Mode (Master Mode, Reception Mode)

MSB First (CBnDIR Bit of CBnCTL0 Register = 0),
CBnCKP Bit of the CBnCTL1 Register = 0,
CBnDAP Bit of the CBnCTL1 Register = 0,
Transfer Data Length = 8 Bits (CSnCL3 to CBnCL0 Bits of CBnCTL2 Register = 0000B)



<1> Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
<2> Set the CBnRXE bit of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit of the CBnCTL0 register, to set the reception enabled status.
<3> Set the CBnPWR bit of the CBnCTL0 register to 1 to enable CSIB operating clock supply.
<4> Perform a dummy read of the CBnRX register (reception start trigger).
<5> The reception complete interrupt (INTCBnR) is output, notifying the CPU that reading the CBnRX (CBnRXL) register is possible.
<6> Clear the CBnSCE bit of the CBnCTL0 register to 0 to set the reception end data status.
<7> Read the CBnRX register before clearing the CBnPWR bit to 0.
<8> Check that the CBnTSF bit of the CBnSTR register is 0 and clear the CBnPWR bit to 0 to stop clock supply to CSIB (end of reception).

To continue transfer, repeat steps <4> and <5> before <6>. (At this time, <4> is not a dummy read, but a receive data read combined with the reception trigger.)

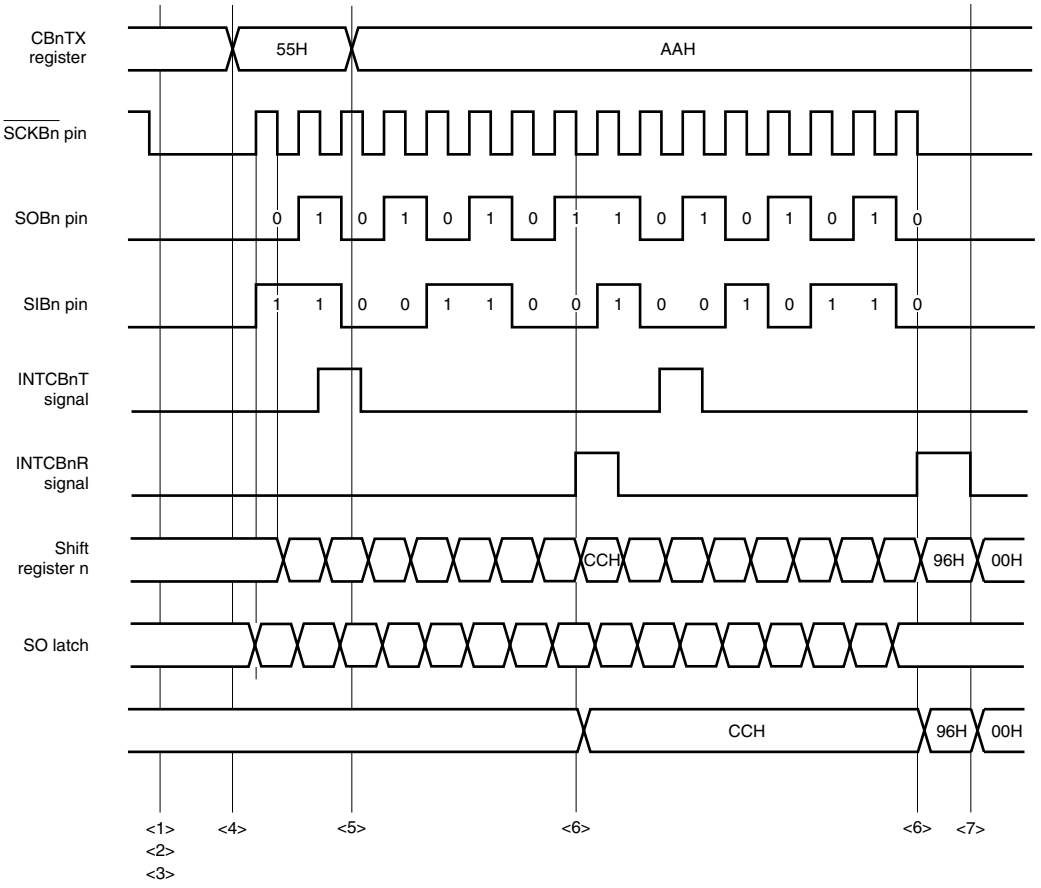
Remarks: 1. The processing of steps <2> and <3> can be set simultaneously.

2. n = 0, 1

16.4.4 Continuous mode (master mode, transmission/reception mode)

Figure 16-12: Continuous Mode (Master Mode, Transmission/Reception Mode)

MSB First (CBnDIR Bit of CBnCTL0 Register = 0),
CBnCKP Bit of the CBnCTL1 Register = 1,
CBnDAP Bit of the CBnCTL1 Register = 0,
Transfer Data Length = 8 Bits (CSnCL3 to CBnCL0 Bits of CBnCTL2 Register = 0000B)



<1> Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
<2> Set the CBnTXE and CBnRXE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit of the CBnCTL0 register, to set the transmission/ reception enabled status.
<3> Set the CBnPWR bit of the CBnCTL0 register is 1 to enable CSIB operating clock supply.
<4> Write transfer data to the CBnTX register (transmission start).
<5> The transmission enable interrupt (INTCBnT) is received and transfer data is written to the CBnTX register.
<6> The reception complete interrupt (INTCBnR) is output, notifying the CPU that reading the CBnRX (CBnRXL) register is possible.
Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
<7> Check that the CBnTSF bit of the CBnSTR register is 0 and clear the CBnPWR bit to 0 to stop clock supply to CSIB (end of transmission/reception).

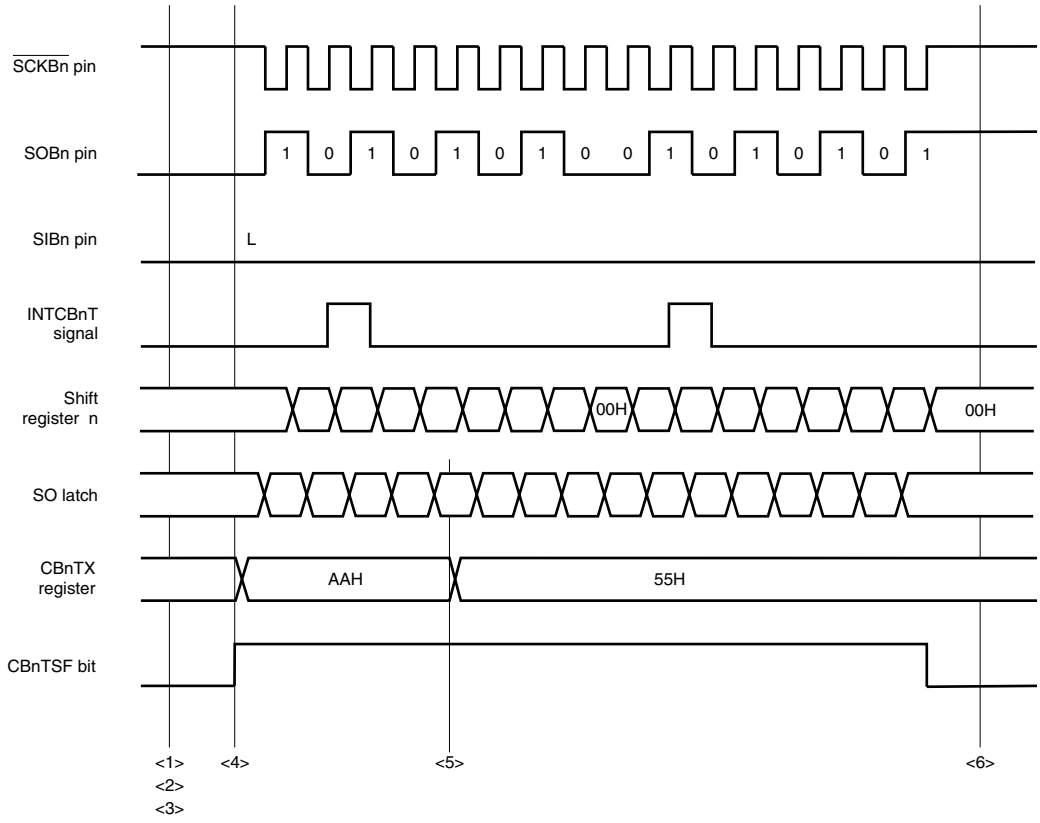
To continue transfer, repeat steps <4> to <6> before <7>.

Remark: n = 0, 1

16.4.5 Continuous mode (master mode, transmission mode)

Figure 16-13: Continuous Mode (Master Mode, Transmission Mode)

MSB First (CBnDIR Bit of CBnCTL0 Register = 0),
CBnCKP Bit of the CBnCTL1 Register = 0,
CBnDAP Bit of the CBnCTL1 Register = 0,
Transfer Data Length = 8 Bits (CSnCL3 to CBnCL0 Bits of CBnCTL2 Register = 0000B)



<1> Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
<2> Set the CBnTXE of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit of the CBnCTL0 register, to set the transmission/reception enabled status.
<3> Set the CBnPWR bit of the CBnCTL0 register is 1 to enable CSIB operating clock supply.
<4> Write transfer data to the CBnTX register (transmission start).
<5> The transmission enable interrupt (INTCBnT) is received and transfer data is written to the CBnTX register.
<6> Check that the CBnTSF bit of the CBnSTR register is 0 and clear the CBnPWR bit to 0 to stop clock supply to CSIB (end of transmission/reception).

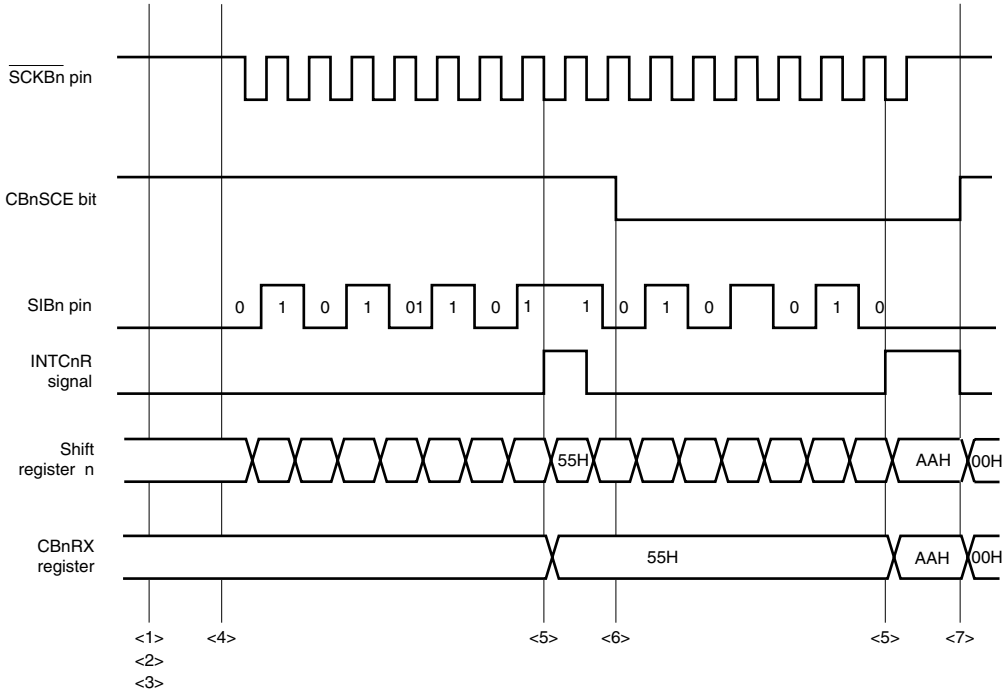
To continue transfer, repeat steps <4> and <5> before <6>.

Remark: n = 0, 1

16.4.6 Continuous mode (master mode, reception mode)

Figure 16-14: Continuous Mode (Master Mode, Reception Mode)

MSB First (CBnDIR Bit of CBnCTL0 Register = 0),
CBnCKP Bit of the CBnCTL1 Register = 0,
CBnDAP Bit of the CBnCTL1 Register = 1,
Transfer Data Length = 8 Bits (CSnCL3 to CBnCL0 Bits of CBnCTL2 Register = 0000B)



<1> Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
<2> Set the CBnRXE bit of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit of the CBnCTL0 register, to set the reception enabled status.
<3> Set the CBnPWR bit of the CBnCTL0 register is 1 to enable CSIB operating clock supply.
<4> Perform a dummy read of the CBnRX register (reception start trigger).
<5> The reception complete interrupt (INTCBnR) is output, notifying the CPU that reading the CBnRX (CBnRXL) register is possible. Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
<6> Clear the CBnSCE bit of the CBnCTL0 register is 0 to set the reception end data status.
<7> Check that the CBnTSF bit of the CBnSTR register is 0 and clear the CBnPWR bit to 0 to stop clock supply to CSIB (end of reception).

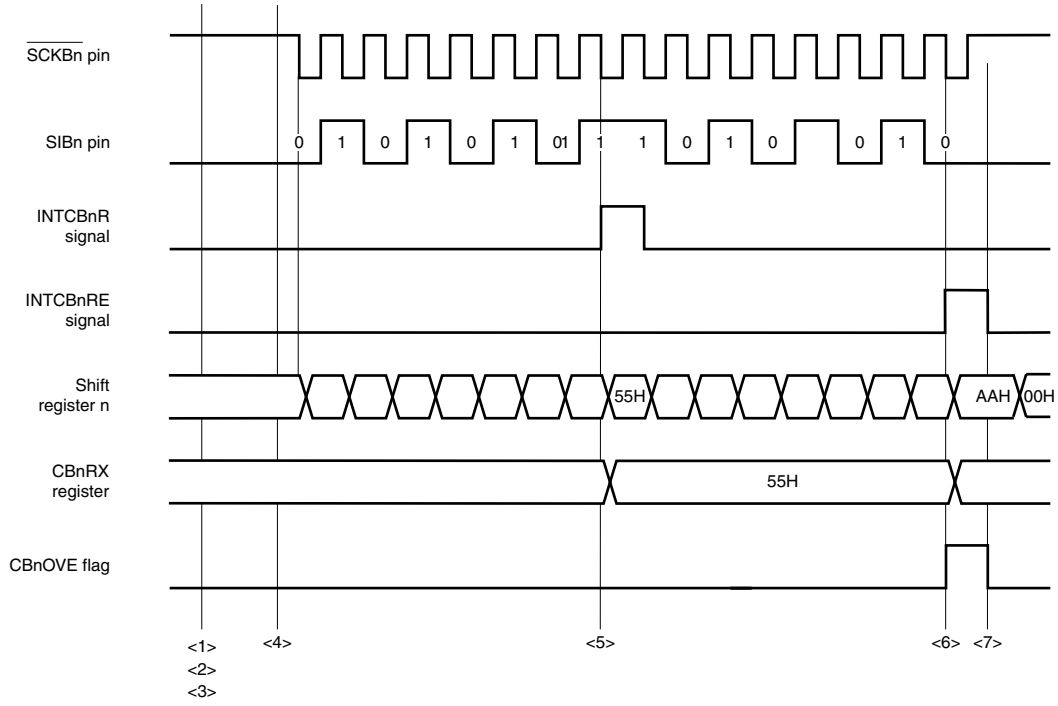
To continue transfer, repeat steps <4> and <5> before <6>.

Remark: n = 0, 1

16.4.7 Continuous reception mode (error)

Figure 16-15: Continuous Reception Mode (Error)

MSB First (CBnDIR Bit of CBnCTL0 Register = 0),
CBnCKP Bit of the CBnCTL1 Register = 0,
CBnDAP Bit of the CBnCTL1 Register = 1,
Transfer Data Length = 8 Bits (CSnCL3 to CBnCL0 Bits of CBnCTL2 Register = 0000B)



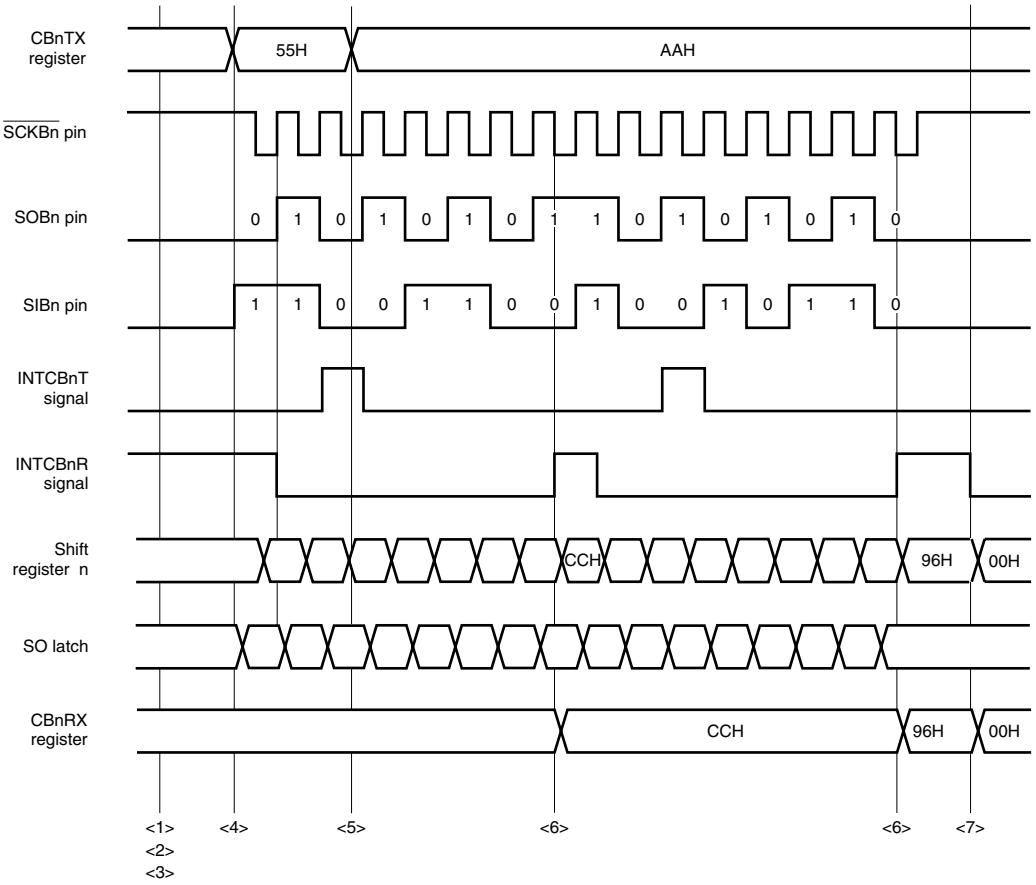
<1> Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
<2> Set the CBnTXE and CBnRXE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit of the CBnCTL0 register, to set the transmission/ reception enable status.
<3> Set the CBnPWR bit of the CBnCTL0 register to 1 to enable CSIB operating clock supply.
<4> Perform a dummy read of the CBnRX register (reception start trigger).
<5> The reception complete interrupt (INTCBnR) is output, notifying the CPU that reading the CBnRX (CBnRXL) register is possible.
<6> If the data could not be read before the end of the next transfer, a receive error interrupt (INTCBnRE) is output and the CBnOVE flag of the CBnSTR register is set (1). The CBnRX register is read as error restore processing.
<7> Check that the CBnTSF bit of the CBnSTR register is 0 and clear the CBnPWR bit to 0 to stop clock supply to CSIBn (end of reception).

Remark: n = 0, 1

16.4.8 Continuous mode (slave mode, transmission/reception mode)

Figure 16-16: Continuous Mode (Slave Mode, Transmission/Reception Mode)

MSB First (CBnDIR Bit of CBnCTL0 Register = 0),
CBnCKP Bit of the CBnCTL1 Register = 0,
CBnDAP Bit of the CBnCTL1 Register = 1,
Transfer Data Length = 8 Bits (CSnCL3 to CBnCL0 Bits of CBnCTL2 Register = 0000B)



<1> Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
<2> Set the CBnTXE and CBnRXE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit of the CBnCTL0 register, to set the transmission/ reception enabled status.
<3> Set the CBnPWR bit of the CBnCTL0 register to 1 to enable CSIB operating clock supply.
<4> Write the transfer data to the CBnTX register.
<5> The transmission enable interrupt (INTCBnT) is received and the transfer data is written to the CBnTX register.
<6> The reception complete interrupt (INTCBnR) is output, notifying the CPU that reading the CBnRX register is possible.
Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
<7> Check that the CBnTSF bit of the CBnSTR register is 0 and clear the CBnPWR bit to 0 to stop clock supply to CSIB (end of transmission/reception).

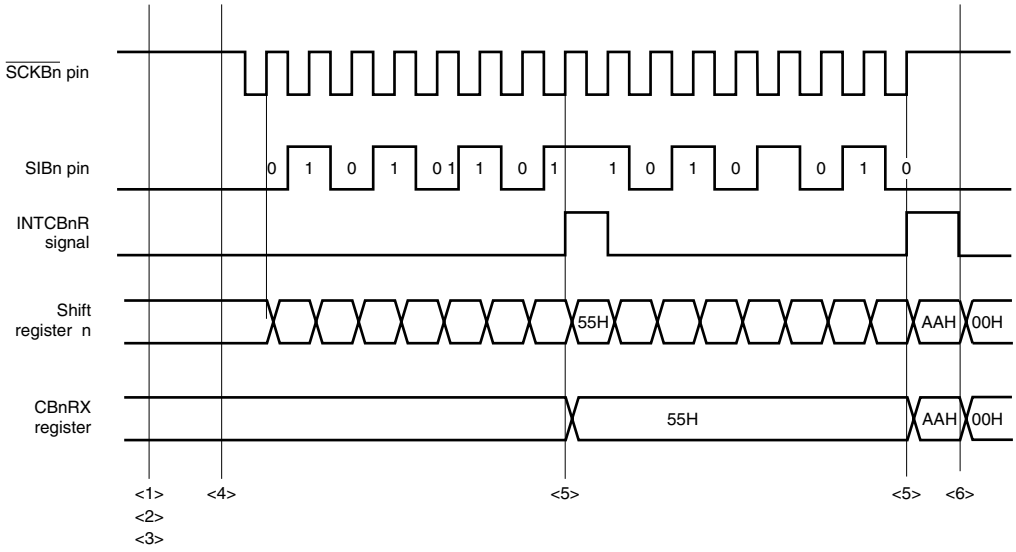
To continue transfer, repeat steps <4> to <6> before <7>.

Remark: n = 0, 1

16.4.9 Continuous mode (slave mode, reception mode)

Figure 16-17: Continuous Mode (Slave Mode, Reception Mode)

MSB First (CBnDIR Bit of CBnCTL0 Register = 0),
CBnCKP Bit of the CBnCTL1 Register = 0,
CBnDAP Bit of the CBnCTL1 Register = 0,
Transfer Data Length = 8 Bits (CSnCL3 to CBnCL0 Bits of CBnCTL2 Register = 0000B)



<1> Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
<2> Set the CBnRXE bit of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit of the CBnCTL0 register, to set the reception enabled status.
<3> Set the CBnPWR bit of the CBnCTL0 register to 1 to enable CSIB operating clock supply.
<4> Perform a dummy read of the CBnRX register (reception start trigger).
<5> The reception complete interrupt (INTCBnR) is output, notifying the CPU that reading the CBnRX register is possible.
Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
<6> Check that the CBnTSF bit of the CBnSTR register is 0 and clear the CBnPWR bit to 0 to stop clock supply to CSIB (end of reception).

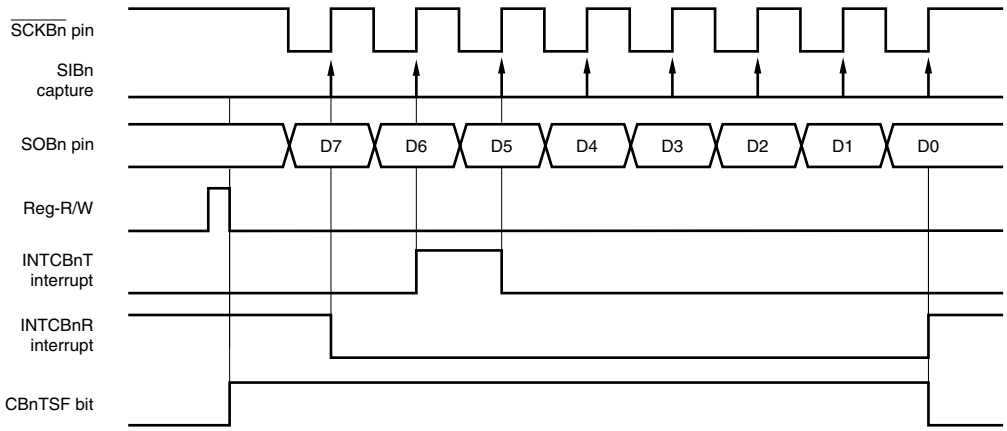
To continue transfer, repeat steps <4> and <5> before <6>.

Remark: n = 0, 1

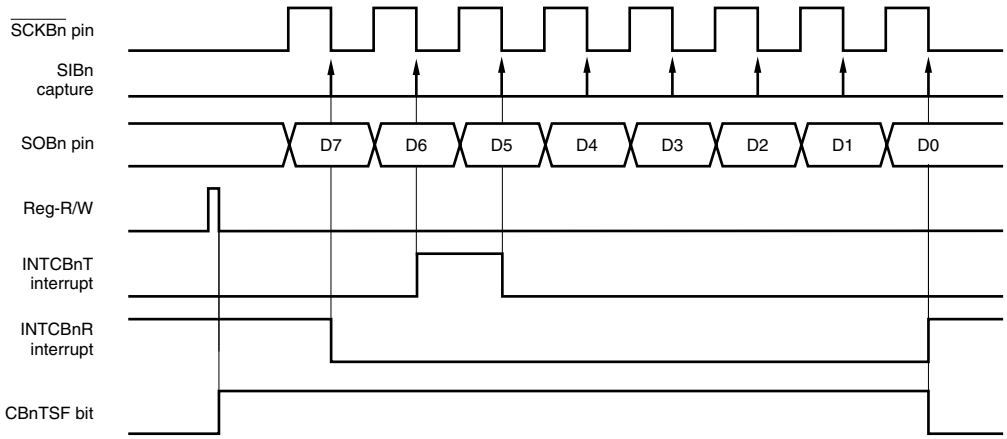
16.4.10 Clock timing

Figure 16-18: CSIBn Clock Timing (1/2)

(a) CBnCKP = 0, CBnDAP = 0



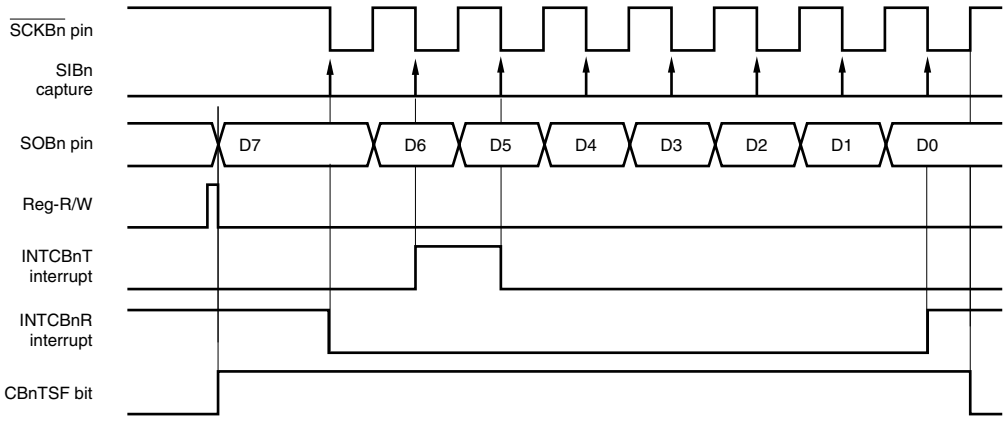
(b) CBnCKP = 1, CBnDAP = 0



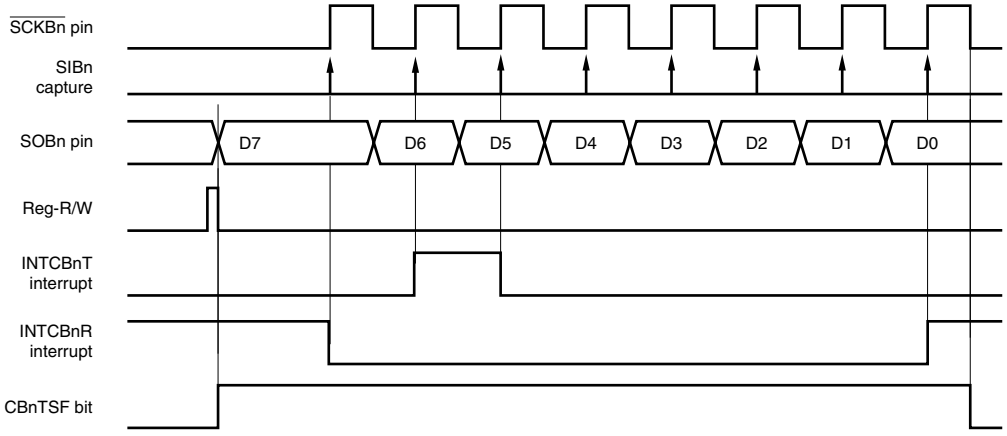
Remark: n = 0, 1

Figure 16-18: CSIBn Clock Timing (2/2)

(c) CBnCKP = 0, CBnDAP = 1



(d) CBnCKP = 1, CBnDAP = 1



Remark: n = 0, 1

16.5 Output Pins

(1) $\overline{\text{SCKBn}}$ pin

When CSIBn operation is disabled (CBnPWR bit of CBnCTL0 register = 0), the $\overline{\text{SCKBn}}$ pin output status is as follows.

CBnCKP	$\overline{\text{SCKBn}}$ Pin Output
0	Fixed to high level
1	Fixed to low level

Remarks: 1. The $\overline{\text{SCKBn}}$ pin output changes when the CBnCKP bit of the CBnCTL1 register is rewritten.

2. n = 0, 1

(2) SOBn pin

When CSIBn operation is disabled (CBnPWR bit = 0), the SOBn pin output status is as follows.

CBnTXE	CBnDAP	CBnDIR	SOBn Pin Output
0	×	×	Fixed to high level
1	0	×	SOBn latch value (low level)
	1	0	CBnTXn value (MSB)
		1	CBnTXn value (LSB)

Remarks: 1. The SOBn pin output changes when any one of the CBnTXE, CBnDAP, and CBnDIR bits of the CBnCTL1 register is rewritten.

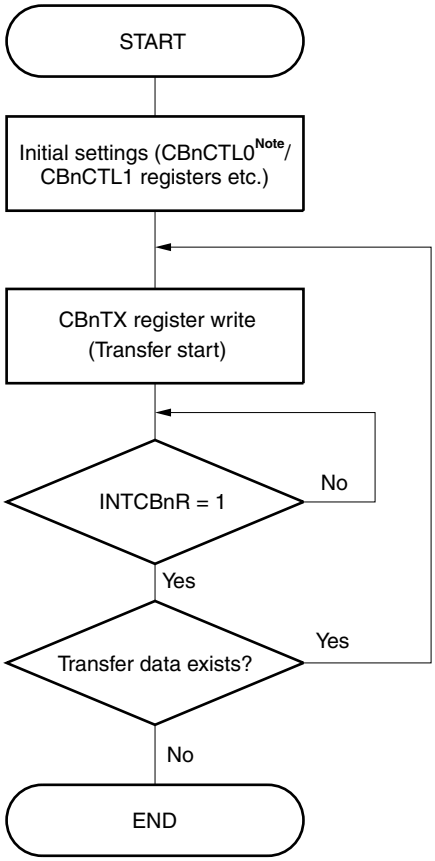
2. n = 0, 1

3. ×: don't care

16.6 Operation Flow

(1) Single transmission

Figure 16-19: Operation Flow of Single Transmission

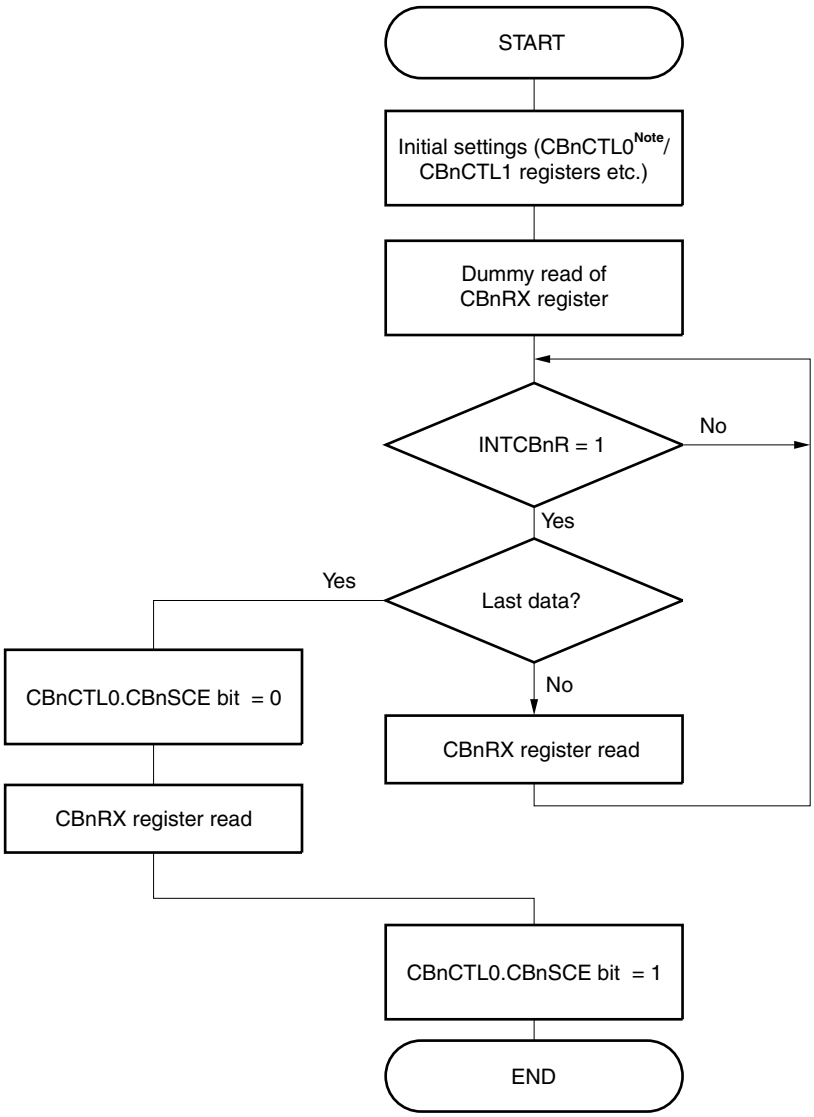


Note: Set the CBnSCE bit of CBnCTL0 register to 1 as part of the initial settings.

Remark: n = 0, 1

(2) Single reception (master)

Figure 16-20: Operation Flow of Single Reception (Master)

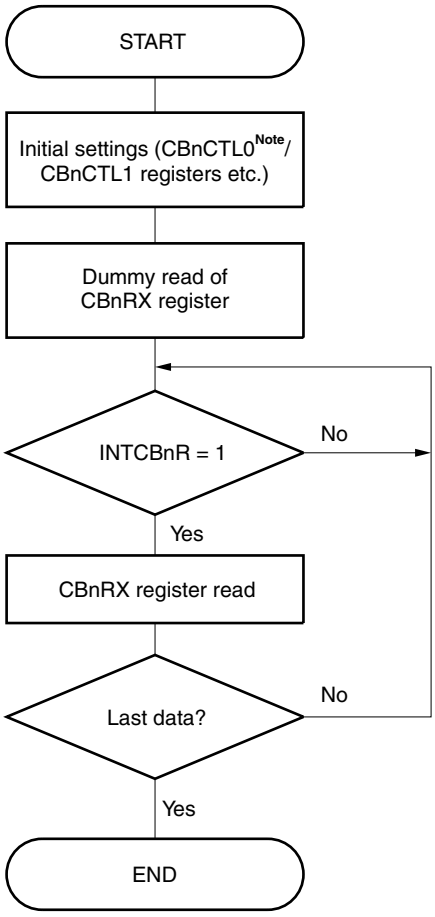


Note: Set the CBnSCE bit of CBnCTL0 register to 1 as part of the initial settings.

Remark: n = 0, 1

(3) Single reception (slave)

Figure 16-21: Operation Flow of Single Reception (Slave)

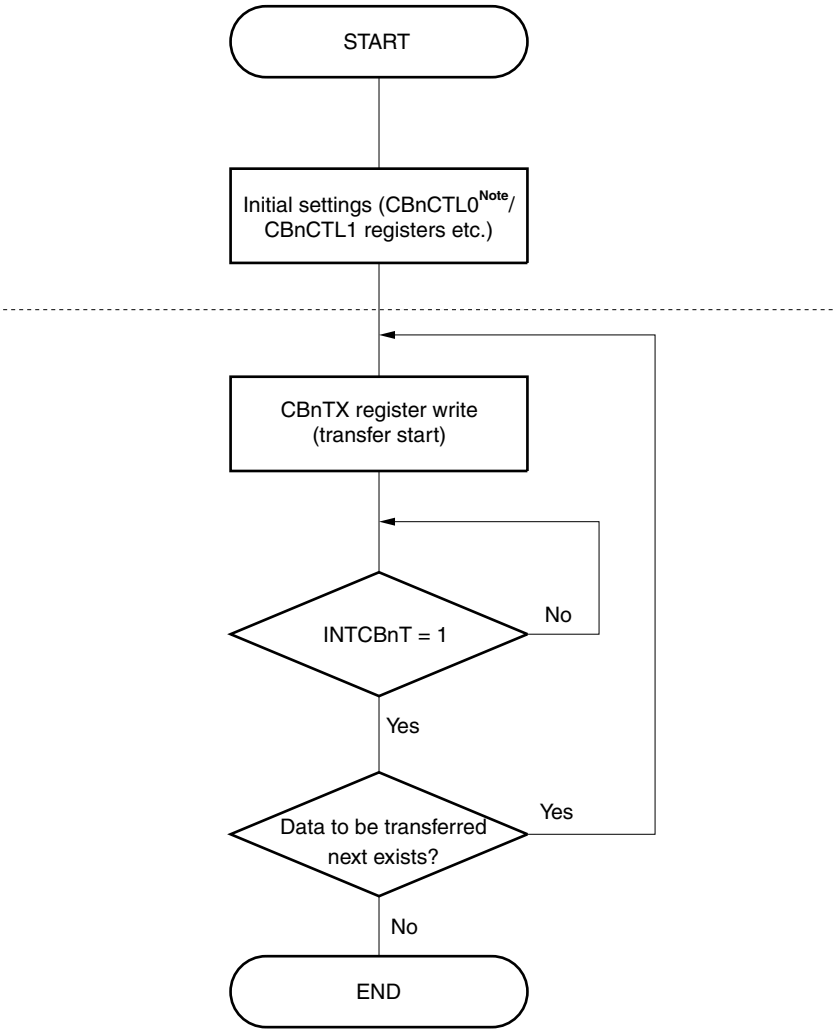


Note: Set the CBnSCE bit of CBnCTL0 register to 1 as part of the initial settings.

Remark: n = 0, 1

(4) Continuous transmission

Figure 16-22: Operation Flow of Continuous Transmission



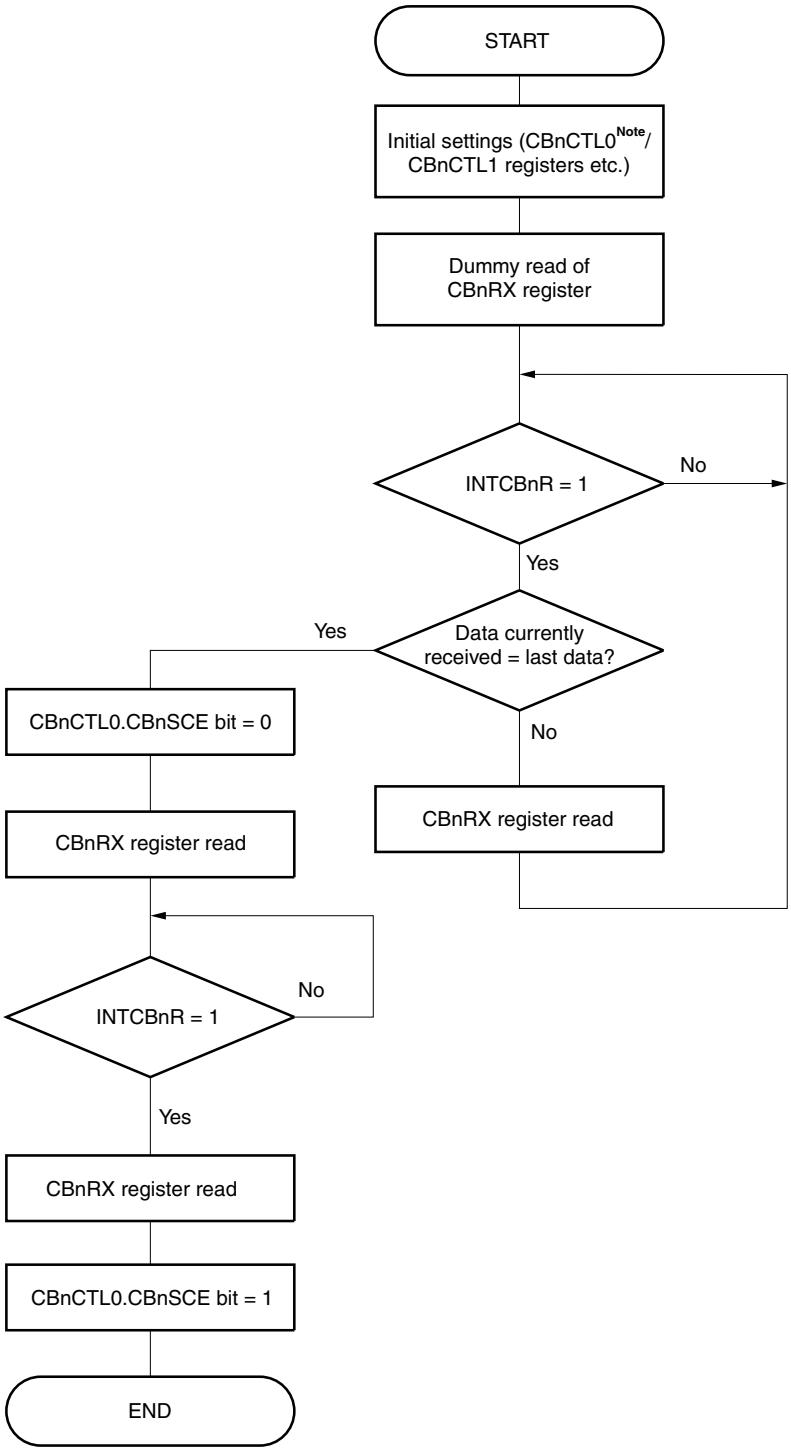
Note: Set the CBnSCE bit of CBnCTL0 register to 1 as part of the initial settings.

Remarks: 1. The steps below the broken line constitute the transmission flow. Execute only steps below the broken line when starting the second and subsequent transmissions.

2. n = 0, 1

(5) Continuous reception (master)

Figure 16-23: Operation Flow of Continuous Reception (Master)



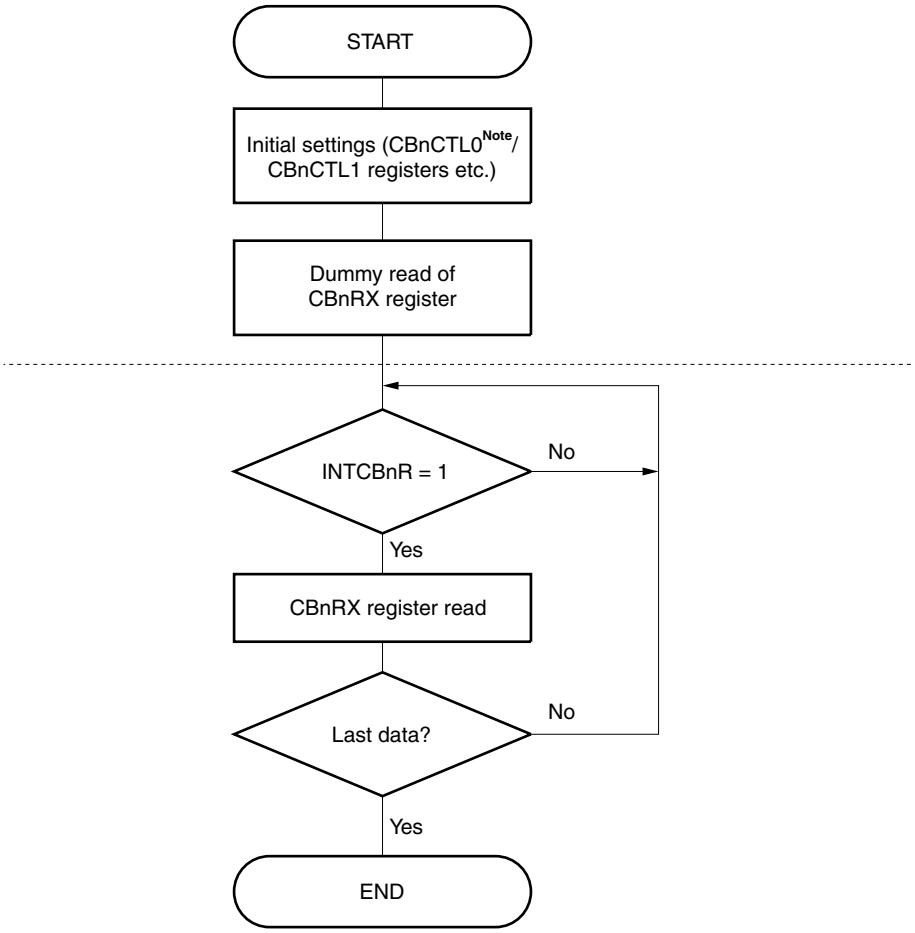
Note: Set the CBnSCE bit of CBnCTL0 register to 1 as part of the initial settings.

Remarks: 1. The steps below the broken line constitute the transmission flow. Execute only steps below the broken line when starting the second and subsequent transmissions.

2. n = 0, 1

(6) Continuous reception (slave)

Figure 16-24: Operation Flow of Continuous Reception (Slave)



Note: Set the CBnSCE bit of CBnCTL0 register to 1 as part of the initial settings.

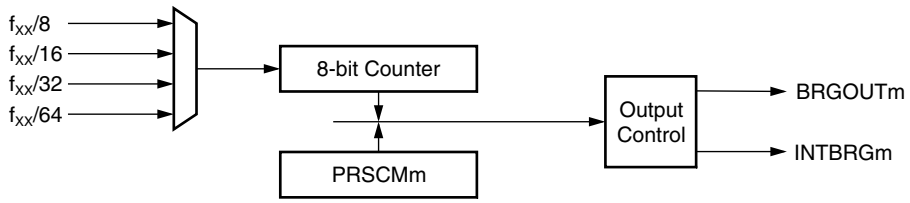
Remarks: 1. The steps below the broken line constitute the transmission flow. Execute only steps below the broken line when starting the second and subsequent transmissions.

2. n = 0, 1

16.7 Baud Rate Generator

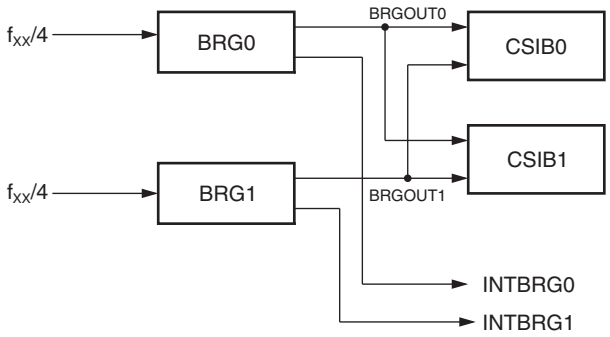
16.7.1 Configuration

Figure 16-25: Block Diagram of Baud Rate Generators 0 and 1 (BRG0, BRG1)



The baud rate generators 0 and 1 (BRG0, BRG1) and CSIB0 and CSIB1 are connected as shown in the following block diagram.

Figure 16-26: Block Diagram of CSIBn Baud Rate Generators



Remarks: 1. An unused baud rate generator (BRGm) can be employed as interval timer generating a dedicated interrupt request (INTBRGm).

2. m = 0, 1

16.7.2 Control registers

(1) Prescaler mode registers 0 and 1 (PRSM0, PRSM1)

The PRSMm register controls generation of a baud rate signal for CSIB (m = 0, 1).
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Figure 16-27: Prescaler Mode Registers 0 and 1 (PRSM0, PRSM1)

After reset: 00H R/W Address: PRSM0 FFFFFDC0H, PRSM1 FFFFFDD0H

	7	6	5	4	3	2	1	0
PRSMm (m = 0, 1)		0	0	BGCEm	0	0	BGCSm1	BGCSm0

BGCEm	Baud Rate Generator Output Control
0	Disabled
1	Enabled

BGCSm1	BGCSm0	Baud Rate Generator Clock Selection (f_{BGCSm})	Setting Value (k)
0	0	$f_{XX}/4$	2
0	1	$f_{XX}/8$	3
1	0	$f_{XX}/16$	4
1	1	$f_{XX}/32$	5

Cautions: 1. Do not rewrite the PRSMm register during operation.

2. Set the BGCSm1, BGCSm0 bits before setting the BGCEm bit to 1.

Remark: m = 0, 1

(2) Prescaler compare registers 0 and 1 (PRSCM0, PRSCM1)

The PRSCMm register is an 8-bit compare register (m = 0, 1).
This register can be read or written in 8-bit units.
Reset input clears this register to 00H.

Figure 16-28: Prescaler Compare Registers 0 and 1 (PRSCM0, PRSCM1)

After reset: 00H R/W Address: PRSM0 FFFFFDC1H, PRSM1 FFFFFDD1H

PRSCMm (m = 0, 1)

7	6	5	4	3	2	1	0
PRSCMm7	PRSCMm6	PRSCMm5	PRSCMm4	PRSCMm3	PRSCMm2	PRSCMm1	PRSCMm0

PRSCMm7	PRSCMm6	PRSCMm5	PRSCMm4	PRSCMm3	PRSCMm2	PRSCMm1	PRSCMm0	Serial Clock	N
0	0	0	0	0	0	0	0	$f_{BGSCm}/512$	256
0	0	0	0	0	0	0	1	$f_{BGSCm}/2$	1
0	0	0	0	0	0	1	0	$f_{BGSCm}/4$	2
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	$f_{BGSCm}/504$	252
1	1	1	1	1	1	0	1	$f_{BGSCm}/506$	253
1	1	1	1	1	1	1	0	$f_{BGSCm}/508$	254
1	1	1	1	1	1	1	1	$f_{BGSCm}/510$	255

Cautions: 1. Do not rewrite the PRSCMm register during operation.

2. Set the PRSCMm register before setting the BGCEm bit of the PRSMm register to 1.

Remarks: 1. f_{BGCSm}: Clock frequency selected by the BGCSm1, BGCSm0 bits of the PRSMm register.

2. m = 0, 1

16.7.3 Baud rate generation

The transmission/reception clock is generated by dividing the main clock. The baud rate generated from the main clock is obtained by the following equation.

$$f_{BRGm} = \frac{f_{BGCSm}}{N \times 2} = \frac{f_{XX}}{2^k \times N \times 2}$$

Remarks: **1.** f_{BRGm}: BRGm count clock

2. f_{BGCSm}: Clock frequency selected by the BGCSm1, BGCSm0 bits of the PRSMm register.

3. f_{XX}: Main clock oscillation frequency

4. k: PRSMm register setting value ($2 \leq k \leq 5$)

5. N: PRSCMm register setting value (1 to 255), when PRSCMm = 01H to FFH, or N = 256, when PRSCMm = 00H.

6. m = 0, 1

Chapter 17 Clocked Serial Interface 3 (CSI3)

17.1 Features

- Transfer rate: Maximum 8 Mbps
- Master mode and slave mode selectable
- Serial clock and data phase switchable
- Transmission data length: 8 to 16 bits (selectable in 1-bit units)
- Transfer data MSB-first/LSB-first switchable
- Transmission mode, reception mode, and transmission/reception mode selectable
- 3-wire serial interface
  - SO3n: Serial data output
  - SI3n: Serial data input
  - $\overline{\text{SCK3n}}$: Serial clock I/O
- Four external chips select signal outputs (SCS3n0 to SCS3n3)
- Interrupt request signals × 2
  - Transmission/reception completion interrupt (INTC3n)
  - CSIBUFn overflow interrupt (INTC3nOVF)
- Sixteen on-chip 20-bit transmit/receive buffers (CSIBUFn)
- On-chip dedicated baud rate generator

Remark: n = 0, 1

17.2 Configuration

CSI3n is controlled by the clocked serial interface mode register 3n (CSIM3n) (n = 0, 1).

(1) Clocked serial interface mode registers 30, 31 (CSIM30, CSIM31)

The CSIM3n register is an 8-bit register for specifying the operation of CSI3n.

(2) Clocked serial interface clock select registers 30, 31 (CSIC30, CSIC31)

The CSIC3n register is an 8-bit register for controlling the operation clock and operating mode of CSI3n.

(3) Serial I/O shift registers 30, 31 (SIO30, SIO30)

The SIO3n register is an 8-bit register for converting between serial data and parallel data. SIO3n is used for both transmission and reception.
Data is shifted in (reception) or shifted out (transmission) beginning at either the MSB side or the LSB side.

(4) Receive data buffer registers 30, 31 (SIRB30, SIRB31)

The SIRB3n register is a 16-bit buffer register that stores receive data. This register is also divided into two registers: the higher 8 bits (SIRB3nH) and lower 8 bits (SIRB3nL).

(5) Chip select CSI buffer register 30, 31 (SFCS30, SFCS31)

The SFCS3n register is a 16-bit buffer register that stores chip select data. The lower 8 bits can also be accessed by an 8-bit buffer register (SFCS3nL).

(6) Transmit data CSI buffer registers 30, 31 (SFDB30, SFDB31)

The SFDB3n register is a 16-bit buffer register that stores transmit data. This register is also divided into two registers: the higher 8 bits (SFDB3nH) and lower 8 bits (SFDB3nL).

(7) CSIBUF status registers 30, 31 (SFA30, SFA31)

The SFA3n register is an 8-bit register that indicates the status of CSI data buffer register n (CSIBUFn) or the transfer status.

(8) Transfer data length select registers 30, 31 (CSIL30, CSIL31)

The CSIL3n register is an 8-bit register that selects the CSI3n transfer data length.

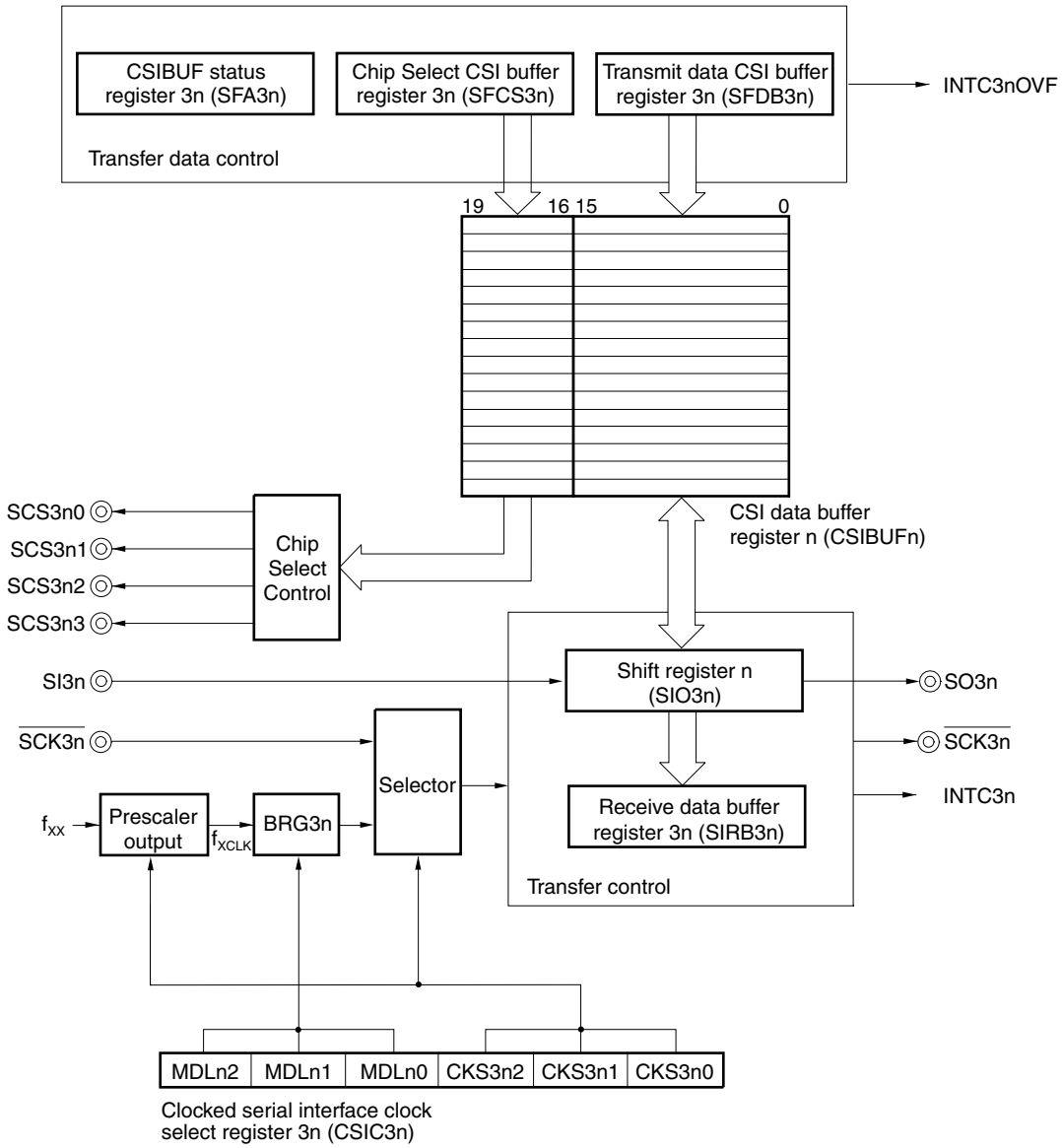
(9) Transfer data number specification registers 30, 31 (SFN30, SFN31)

The SFN3n register is an 8-bit register that sets the number of CSI3n transfer data in consecutive mode.

(10) CSI data buffer registers 0, 1 (CSIBUF0, CSIBUF1)

By consecutively writing transmit data to the SFDB3n register from where it is transferred, the data can be stored in the CSIBUFn register while the CSIBUFn pointer for writing is automatically incremented (CSIBUFn).
The CSIBUFn is a 16-bit buffer register.

Figure 17-1: Block Diagram of Clocked Serial Interface 3n (CSI3n)



Remarks: 1. n = 0, 1

2. f_{XX}: Main clock
f_{XCLK}: Basic clock selected by CKS3n2 to CKS3n0 bits of CSIC3n register

17.3 Control Registers

(1) Clocked serial interface mode registers 3n (CSIM3n)

The CSIM3n register controls the operation of CSI3n (n = 0, 1).
This registers can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Cautions: 1. Writing the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits is enabled only when CTXEn bit = 0 and CRXEn bit = 0.

2. To use CSI3n, be sure to set the external pins related to the CSI3n function to control the mode and set the CSIC3n register. Then set the CSICAEn bit to 1 before setting the other bits.

Figure 17-2: Clocked Serial Interface Mode Register 3n (CSIM3n) (1/2)

After reset: 00H R/W Address: CSIM30 FFFFFD40H, CSIM31 FFFFFD60H

CSIM3n (n = 0, 1)

7	6	5	4	3	2	1	0
CSICAEn	CTXEn	CRXEn	TRMDn	DIRn	CSITn	CSWEn	CSMDn

CSICAEn	CSI3n Operation Clock Control
0	Stops clock supply to CSI3n
1	Supplies clock to CSI3n
Cautions: 1.	**The CSI3n unit is reset when the CSICAEn bit = 0, and CSI3n is stopped. To operate CSI3n, first set the CSICAEn bit to 1.**
2.	**When rewriting the CSICAEn bit from 0 to 1 or from 1 to 0, simultaneously rewriting the bits other than the CSICAEn bit of the CSIM3n register is prohibited. When the CSICAEn bit = 0, rewriting the bits other than the CSICAEn bit of the CSIM3n register, and the SFDB3n, SFDB3nL, and SFA3n registers is prohibited.**

CTXEn	Transmission Operation Enable
0	Disables transmission
1	Enables transmission
Caution:	**The CTXEn bit is reset when the CSICAEn bit is cleared to 0.**

CRXEn	Reception Operation Enable
0	Disables reception
1	Enables reception
Caution:	**The CRXEn bit is reset when the CSICAEn bit is cleared to 0.**

Remark: n = 0, 1

Figure 17-2: Clocked Serial Interface Mode Register 3n (CSIM3n) (2/2)

TRMDn	Transfer Mode Specification
0	Single mode
1	Consecutive mode

DIRn	Transfer Direction Specification
0	MSB-first transfer
1	LSB-first transfer
Specifies the transfer direction when data is written from the SFDB3n register to the CSIBUFn register or read from the SIRB3n and CSIBUFn registers.	

CSITn	Transmission Completion Interrupt (INTC3n) Control
0	No delay
1	Delay mode (The interrupt request signal is delayed by half a cycle.)
Cautions: 1. The delay mode (CSIT bit = 1) is valid only in the master mode (CKS3n2 to CKS3n0 bits of the CSIC3n register other than 111B). In the slave mode (CKS3n2 to CKS3n0 bits = 111B), do not set the delay mode. If the delay mode is set, INTC3n is not affected by the CSITn bit. **2. If the CSITn bit is set to 1 in the consecutive mode (TRMDn bit = 1), the INTC3n interrupt is not output except when the last data set by the SFNn3 to SFNn0 bits of the SFN3n register is transferred, but a delay of half a clock can be inserted between each data transferred.**	

CSWEn	Transfer Wait Control
0	Disables transfer wait.
1	Enables transfer wait (1 wait cycle inserted on starting transfer).
Caution:	**Inserting a transfer wait cycle (CSWEn bit = 1) is valid only in the master mode (CKS3n2 to CKS3n0 bits of the CSIC3n register other than 111B). In the slave mode (CKS3n2 to CKS3n0 bits = 111B), do not insert a transfer wait cycle. If set, a transfer wait cycle is not inserted.**

CSMDn	Chip Select Mode Specification
0	Disables inactive level setting of chip select outputs (SCS3n0 to SCS3n3) during transfer wait.
1	Enables inactive level setting of chip select outputs (SCS3n0 to SCS3n3) during transfer wait.
Caution:	**The CSMDn bit setting is valid only when the transfer wait is enabled (CSWEn bit = 1) and the master mode is specified (CKS3n0 bits of the CSIC3n register other than 111B).** **In all other cases the CSMDn bit setting is invalid and no inactive level setting of chip select outputs between two consecutive transfers takes place.**

Remark: n = 0, 1

(2) Clocked serial interface clock select register 3n (CSIC3n)

The CSIC3n register is an 8-bit register that controls the operation clock and operating mode of CSI3n.

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

Reset input sets this register to 07H.

Caution: Data can be written to the CSIC3n register only when the CTXEn bit = 0 and CRXEn bit = 0 in the CSIM3n register.

Figure 17-3: Clocked Serial Interface Clock Select Register 3n (CSIC3n) (1/3)

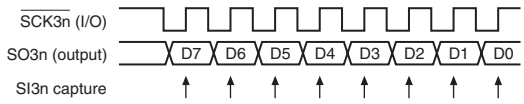
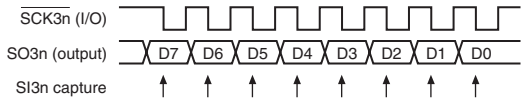
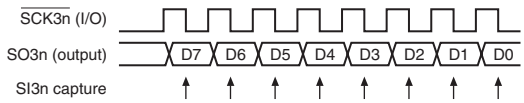
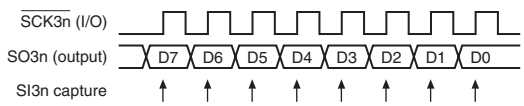
After reset: 07H R/W Address: CSIC30 FFFFFD41H, CSIC31 FFFFFD61H

CSIC3n (n = 0, 1)

7	6	5	4	3	2	1	0
MDLn2	MDLn1	MDLn0	CKPn	DAPn	CKS3n2	CKS3n1	CKS3n0

MDLn2	MDLn1	MDLn0	Set Value (N)	Transfer Clock (BRG3n Output Signal)
0	0	0	–	BRG3n stop mode (power save)
0	0	1	1	$f_{XCLK}/2$
0	1	0	2	$f_{XCLK}/4$
0	1	1	3	$f_{XCLK}/6$
1	0	0	4	$f_{XCLK}/8$
1	0	1	5	$f_{XCLK}/10$
1	1	0	6	$f_{XCLK}/12$
1	1	1	7	$f_{XCLK}/14$
Caution:	**In the slave mode (CKS3n2 to CKS3n0 bits = 111B), it is recommended to clear the MDLn2 to MDLn0 bits to 000 (BRG3n stop mode).**			

Figure 17-3: Clocked Serial Interface Clock Select Register 3n (CSIC3n) (2/3)

CKPn	DAPn	Specification of Data Transmission/Reception Timing in Relation to Clock Phase
0	0	
0	1	
1**Note**	0	
1**Note**	1	

Note: If the CKPn bit is set to 1 in the master mode (CKS3n2 to CKS3n0 bits are other than 111B), the $\overline{\text{SCK3n}}$ pin outputs a low level when it is inactive. If the CTXEn bit of the CSIM3n register is cleared to 0 (disabling transmission) and CRXEn bit is cleared to 0 (disabling reception), the $\overline{\text{SCK3n}}$ pin outputs a high level.

Therefore, take the following measures to fix the $\overline{\text{SCK3n}}$ pin to low level when CSI3n is not used.

[$\overline{\text{SCK3n}}$ pin]

<1> Clear the corresponding port bit (P82 of the P8 register for CSI30, or P92 of the P9 register for CSI31) to 0:
The port output level is set to low.

<2> Clear the corresponding bit in the port mode register (PM82 of the PM8 register for CSI30, or PM92 of the PM9 register for CSI31) to 0:
The pin is set into output mode.

<3> Clear the corresponding bit in the port mode control register (PMC82 of the PMC8 register for CSI30, or PMC92 of the PMC9 register for CSI31) to 0:
The pin is set into port mode (fixed to low-level output).

<4> Clear the CTXEn and CRXEn bits of the CSIM3n register to 0:
Transmission and reception are disabled.

<5> Set the CTXEn or CRXEn bit of the CSIM3n register to 1:
Transmission or reception is enabled (both transmission and reception can also be enabled).

<6> Set the corresponding bit in the port mode control register (PMC82 of the PMC8 register for CSI30, or PMC92 of the PMC9 register for CSI31) to 1:
The pin is set in the control mode ($\overline{\text{SCK3n}}$ pin output).

Because the register set values <1> and <2> are retained, control can be performed only by <3> to <6> once they have been set.

Remark: n = 0, 1

Figure 17-3: Clocked Serial Interface Clock Select Register 3n (CSIC3n) (3/3)

CKS3n2	CKS3n1	CKS3n0	Set Value (k)	Basic Clock (f_{XCLK})	Mode
0	0	0	0	f_{XX}	Master mode
0	0	1	1	$f_{XX}/2$	Master mode
0	1	0	2	$f_{XX}/4$	Master mode
0	1	1	3	$f_{XX}/8$	Master mode
1	0	0	4	$f_{XX}/16$	Master mode
1	0	1	5	$f_{XX}/32$	Master mode
1	1	0	6	$f_{XX}/64$	Master mode
1	1	1	–	External clock ($\overline{SCK3n}$)	Slave mode

Remarks: 1. f_{XX}: Main clock

2. n = 0, 1

(3) Receive data buffer register 3n (SIRB3n, SIRB3nL, SIRB3nH)

The SIRB3n register is a 16-bit buffer register that stores receive data. It is overlayed by an 8-bit buffer register SIRB3nL on the lower 8 bits, and an 8-bit buffer register SIRB3nH on the higher 8 bits.

By consecutively reading this register in the consecutive mode (TRMDn bit of the CSIM3n register = 1), the received data in the CSIBUFn register can be sequentially read while the CSIBUFn pointer for reading is incremented.

In the single mode (TRMDn bit of the CSIM3n register = 0), received data is read by reading the SIRB3n register and it is judged that the SIRB3n register has become empty.

The SIRB3n register is read-only, in 16-bit units.

The SIRB3nL and SIRB3nH registers are read-only, in 8-bit units.

Reset input clears the SIRB3n register to 0000H, and the SIRB3nL and SIRB3nH registers to 00H accordingly.

In addition to reset input, the SIRB3n as well as the SIRB3nL and SIRB3nH registers are initialized by clearing (to 0) the CSICAEn bit of the CSIM3n register.

Figure 17-4: Receive Data Buffer Register 3n (SIRB3n, SIRB3nL, SIRB3nH)

After reset: 0000H[Note] R Address: SIRB30 FFFFFD42H, SIRB31 FFFFFD62H

SIRB3n (n = 0, 1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRB n15	SIRB n14	SIRB n13	SIRB n12	SIRB n11	SIRB n10	SIRB n9	SIRB n8	SIRB n7	SIRB n6	SIRB n5	SIRB n4	SIRB n3	SIRB n2	SIRB n1	SIRB n0
SIRB3nH								SIRB3nL							

After reset: 00H[Note] R Address: SIRB30L FFFFFD42H, SIRB31L FFFFFD62H

SIRB3nL (n = 0, 1)

7	6	5	4	3	2	1	0
SIRBn7	SIRBn6	SIRBn5	SIRBn4	SIRBn3	SIRBn2	SIRBn1	SIRBn0

After reset: 00H[Note] R Address: SIRB30H FFFFFD43H, SIRB31H FFFFFD63H

SIRB3nH (n = 0, 1)

7	6	5	4	3	2	1	0
SIRBn15	SIRBn14	SIRBn13	SIRBn12	SIRBn11	SIRBn10	SIRBn9	SIRBn8

Note: In consecutive mode (TRMDn bit of the CSIM3n register = 1): Undefined

Remark: n = 0, 1

(4) Chip select CSI buffer register 3n (SFCS3n, SFCS3nL)

The SFCS3n register is a 16-bit buffer register that stores transmit data. It is overlayed by an 8-bit buffer register SFCS3nL on the lower 8 bits.

When chip select data is written to the SFCS3n (SFCS3nL) register, the data is stored in the CSIBUFn register following the CSIBUFn pointer for writing. The store operation is executed after next write of the transmit data CSI buffer register SFDB3n (SFDB3nL).

When the data of this register is read, the value of the transmit data written last is read.

The SFCS3n register can be read or written in 16-bit units.

The SFCS3nL register can be read or written, in 8-bit or 1-bit units.

Reset input clears the SFCS3n register to FFFFH, and the SFCS3nL register to FFH accordingly.

Figure 17-5: Chip Select CSI Buffer Register 3n (SFCS3n, SFCS3nL)

After reset: FFFFH R/W Address: SFCS30 FFFFFD44H, SFCS31 FFFFFD64H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFCS3n	SFCS n15	SFCS n14	SFCS n13	SFCS n12	SFCS n11	SFCS n10	SFCS n9	SFCS n8	SFCS n7	SFCS n6	SFCS n5	SFCS n4	SFCS n3	SFCS n2	SFCS n1	SFCS n0
(n = 0, 1)									SFCS3nL							

After reset: FFH RW Address: SFCS30L FFFFFD44H, SFCS31L FFFFFD64H

	7	6	5	4	3	2	1	0
SFCS3nL	SFCSn7	SFCSn6	SFCSn5	SFCSn4	SFCSn3	SFCSn2	SFCSn1	SFCSn0
(n = 0, 1)								

Remark: n = 0, 1

(5) Transmit data CSI buffer register 3n (SFDB3n, SFDB3nL, SFDB3nH)

The SFDB3n register is a 16-bit buffer register that stores transmit data. It is overlayed by an 8-bit buffer register SFDB3nL on the lower 8 bits, and an 8-bit buffer register SFDB3nH on the higher 8 bits.

When transmit data is written to the SFDB3n register, the data is sequentially stored in the CSIBUFn register while the CSIBUFn pointer for writing is incremented.

When the data of this register is read, the value of the transmit data written last is read.

The SFDB3n register can be read or written in 16-bit units.

The SFDB3nL and SFDB3nH registers can be read or written, in 8-bit or 1-bit units.

Reset input clears the SFDB3n register to 0000H, and the SFDB3nL and SFDB3nH registers to 00H accordingly.

Figure 17-6: Transmit Data CSI Buffer Register 3n (SFDB3n, SFDB3nL, SFDB3nH)

After reset: 0000H R/W Address: SFDB30 FFFFFD46H, SFDB31 FFFFFD66H

SFDB3n (n = 0, 1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFDBn15	SFDBn14	SFDBn13	SFDBn12	SFDBn11	SFDBn10	SFDBn9	SFDBn8	SFDBn7	SFDBn6	SFDBn5	SFDBn4	SFDBn3	SFDBn2	SFDBn1	SFDBn0
SFDB3nH								SFDB3nL							

After reset: 00H RW Address: SFDB30L FFFFFD46H, SFDB31L FFFFFD66H

SFDB3nL (n = 0, 1)

7	6	5	4	3	2	1	0
SFDBn7	SFDBn6	SFDBn5	SFDBn4	SFDBn3	SFDBn2	SFDBn1	SFDBn0

After reset: 00H R/W Address: SFDB30H FFFFFD47H, SFDB31H FFFFFD67H

SFDB3nH (n = 0, 1)

7	6	5	4	3	2	1	0
SFDBn15	SFDBn14	SFDBn13	SFDBn12	SFDBn11	SFDBn10	SFDBn9	SFDBn8

Remark: n = 0, 1

(6) CSIBUF status register 3n (SFA3n)

The SFA3n register indicates the status of the CSIBUFn register or the transfer status.
This register can be read or written in 8-bit or 1-bit units (however, bits 6 to 0 can only be read. They do not change even if they are written).
Reset input clears the register to 20H.

Cautions: 1. Reading the SFA3n register is prohibited when the CSICAEn bit of the CSIM3n register is cleared (0).

2. Because the values of the SFFULn, SFEMPn, CSOTn, and SFPn3 to SFPn0 bits may change at any time during transfer, their values during transfer may differ from the actual values. Especially, use the CSOTn bit independently (do not use this bit in relation with the other bits). To detect the end of transfer by the SFA3n register, check to see if the SFEMPn bit is 1 after the data to be transferred has been written to the CSIBUFn register.

3. If the SFA3n register is read immediately after data has been written to the SFDB3n and SFDB3nL registers, the values of the SFFULn, SFEMPn, and SFPn3 to SFPn0 bits do not change in time.

4. If the SFA3n register is read before the SFFULn bit is set to 1 and the 17th data is written, the CSIBUFn overflow interrupt (INTC3nOVF) is generated.

Figure 17-7: CSIBUF Status Register 3n (SFA3n)(1/3)

After reset: 00H R/W Address: SFA30 FFFFFD48H, SFA31 FFFFFD68H

SFA3n (n = 0, 1)

7	6	5	4	3	2	1	0
FPCLRn	SFFULn	SFEMPn	CSOTn	SFPn3	SFPn2	SFPn1	SFPn0

FPCLRn	CSIBUFn Pointer Clear Operation
0	No operation
1	Clear all CSIBUFn pointers

Cautions: 1. This bit is always 0 when it is read.

2. If 1 is written to the FPCLRn bit in the middle of transfer, transfer is aborted. Because all the CSIBUFn pointers are cleared to 0, the remaining data in the CSIBUFn register is ignored.
If 1 is written to the FPCLRn bit, be sure to read the SFA3n register to check to see if all the CSIBUFn pointers have been correctly cleared to 0 (SFFULn bit = 0, SFEMPn bit = 1, SFPn3 to SFPn0 bits = 0000B). Nothing happens even if 0 is written to the FPCLRn bit.

Remark: n = 0, 1

Figure 17-7: CSIBUF Status Register 3n (SFA3n)(2/3)

SFFULn	CSIBUFn Full Status Flag
0	CSIBUFn register has a vacancy
1	CSIBUFn is full

Cautions: 1. This bit is cleared to 0 when the CSICAEn bit of the CSIM3n register is cleared to 0 and the FPCLR bit is set to 1.

2. If transfer of 16 data is specified in the consecutive mode (TRMDn bit of the CSIM3n register = 1) (SFNn3 to SFNn0 bits of the SFN3n register = 0000B), the SFFULn bit is set to 1 in the same way as in the single mode (TRMDn bit of the CSIM3n register = 0) when 16 data are in the CSIBUFn register. If even one of the data has been completely transferred, the SFFULn bit is cleared to 0. However, this does not mean that the CSIBUFn register has a vacancy.

SFEMPn	CSIBUFn Empty Status Flag
0	Data is in CSIBUFn register
1	CSIBUFn is empty

Cautions: 1. This flag is cleared to 0 when the CSICAEn bit of the CSIM3n register is cleared to 0 and the FPCLR bit is set to 1.

2. If the data written to the CSIBUFn register has been transferred in the consecutive mode (TRMDn bit of the CSIM3n register = 1), the SFEMP bit is set to 1 in the same way as in the single mode (TRMDn bit of the CSIM3n register = 0) even if receive data is stored in the CSIBUF register.

CSOTn	Transfer Status Flag
0	Idle status
1	Transfer or transfer start processing in progress

Cautions: 1. This flag is cleared to 0 when the CSICAEn bit of the CSIM3n register is cleared to 0 and the FPCLRn bit is set to 1, or when the CTXEn and CRXEn bits of the CSIM3n register are cleared to 0.

2. This flag is set (1) from when transfer is started until there is no more transfer data in the CSIBUFn register in the single mode (TRMDn bit of the CSIM3n register = 0) or until the specified number of data has been transferred in the consecutive mode (TRMDn bit of the CSIM3n register = 1).

Remark: n = 0, 1

Figure 17-7: CSIBUF Status Register 3n (SFA3n)(3/3)

SFPn3	SFPn2	SFPn1	SFPn0	CSIBUFn Pointer Status
0H to FH (0 to 15)				• In the single mode (TRMDn bit of the CSIM3n register = 0), the “number of transfer data remaining in CSIBUFn register (CSIBUFn pointer value for writing – CSIBUFn pointer value for SIO3n loading)” can be read. • In the consecutive mode (TRMDn bit of the CSIM3n register = 1), the “number of data completely transferred (value of CSIBUFn pointer for SIO3n loading/storing)” can be read. If the SFPn3 to SFPn0 bits are 0H, however, the number of transferred data is as follows, depending on the setting of the SFEMPn bit. When SFEMPn bit = 0: Number of transferred data = 0 When SFEMPn bit = 1: Number of transferred data = 16 or status before starting transfer (before writing transfer data)
Caution: **These bits are cleared to 0 in synchronization with the operating clock when the FPCLRn bit = 1. However, the values of these bits are held until the CSICAEn bit of the CSIM3n register is cleared to 0 or the FPCLRn bit is set to 1.**				

Remark: n = 0, 1

(7) Transfer data length select register 3n (CSIL3n)

The CSIL3n register is used to select the transfer data length of CSI3n.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears the register to 00H.

Caution: The CSIL3n register may be transferring data when the CTXEn or CRXEn bit of the CSIM3n register is 1. Before writing data to the CSIL3n register, be sure to clear the CTXEn and CRXEn bits to 0.

Figure 17-8: Transfer Data Length Select Register 3n (CSIL3n)

After reset: 00H R/W Address: CSIL30 FFFFFD49H, CSIL31 FFFFFD69H

CSIL3n (n = 0, 1)

7	6	5	4	3	2	1	0
CSLVn3	CSLVn2	CSLVn1	CSLVn0	CCLn3	CCLn2	CCLn1	CCLn0

CSLVnm	Chip Select Output (SCS3nm) Level Setting (n = 0, 1; m = 0 to 3)
0	Active level of SCSnm output is low level
1	Active level of SCSnm output is high level

CCLn3	CCLn2	CCLn1	CCLn0	Transfer Data Length
0	0	0	0	16 bits
1	0	0	0	8 bits
1	0	0	1	9 bits
1	0	1	0	10 bits
1	0	1	1	11 bits
1	1	0	0	12 bits
1	1	0	1	13 bits
1	1	1	0	14 bits
1	1	1	1	15 bits
Other than above				Setting prohibited

Caution: If a transfer data length other than 16 bits is specified (CCLn3 to CCLn0 bits = 0000), an undefined value is read to the higher excess bits of the SIRB3n and CSIBUFn registers (see 10.3.5 (3) Data transfer direction specification function).

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

(8) Transfer data number specification register 3n (SFN3n)

The SFN3n register is used to set the number of transfer data of CSI3n in the consecutive mode (TRMDn bit of the CSIM3n register = 1).

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

Figure 17-9: Transfer Data Number Specification Register 3n (SFN3n)

After reset: 00H R/W Address: SFN30 FFFFFD49H, SFN31 FFFFFD69H

	7	6	5	4	3	2	1	0
SFN3n	0	0	0	0	SFNn3	SFNn2	SFNn1	SFNn0

(n = 0, 1)

SFNn3	SFNn2	SFNn1	SFNn0	Number of Transfer Data
0	0	0	0	16
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Caution: Writing data exceeding the value set by the SFNn3 to SFNn0 bits (number of CSI3n transfer data) to the CSIBUFn register is prohibited (data is ignored even if written).

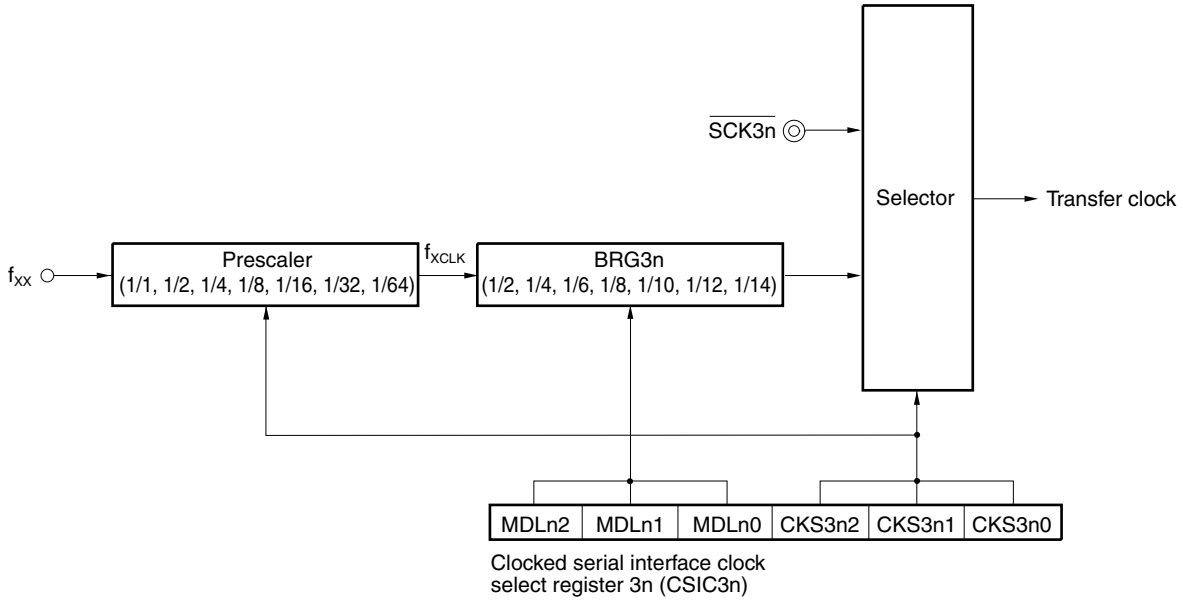
Remark: n = 0, 1

17.4 Dedicated Baud Rate Generator 3n (BRG3n)

The transfer clock of CSI3n can be selected from the output of a dedicated baud rate generator (BRG3n) or external clock (n = 0, 1).
The serial clock source is specified by the CSIC3n register.
In the master mode (CKS3n2 to CKS3n0 bits of the CSIC3n register other than 111B), BRG3n is selected as the clock source.

(1) Transfer clock

Figure 17-10: Transfer Clock of CSI3n



Remarks: 1. n = 0, 1

2. f_{XX}: Main clock
f_{XCLK}: Basic clock selected by CSIC3n register

(2) Baud rate

The baud rate is calculated by the following expression.

$$\text{Baud rate} = \frac{f_{XX}}{N \times 2^{(k+1)}} \text{ [bps]}$$

Remarks: 1. f_{XX}: Main clock

2. k: Value set by CKS3n2 to CKS3n0 bits of CSIC3n register ($0 \leq k \leq 6$)

3. N: Value set by MDLn2 to MDLn0 bits of CSIC3n register ($1 \leq N \leq 7$)

Cautions: 1. If the CKS3n2 to CKS3n0 bits of the CSIC3n register are cleared to 000B, setting the MDLn2 to MDLn0 bits of the CSIC3n register to 001B is prohibited.

2. Because the maximum transfer rate in the master mode (CKS3n2 to CKS3n0 bits other than 111B) is 8 Mbps, do not exceed this value.
Example: When the main clock f_{XX} is 64 MHz, the maximum transfer rate is set when the CKS3n2 to CKS3n0 bits = 000B and the MDLn2 to MDLn0 bits = 100B.

17.5 Operation

17.5.1 Operation modes

Table 17-1: Operation Modes

TRMDn Bit	CKS3n2 to CKS3n0 Bits	CTXEn and CRXEn Bits	DIRn Bit	CSITn Bit	CSWEn Bit	CSMDn Bit
Single mode	Master mode	Transmission/ reception/ transmission and reception	MSB/LSB first	INTC3n delay mode enabled/ disabled	Transfer wait disabled	Intermediate inactive level of chip select outputs disabled
					Transfer wait enabled	Intermediate inactive level of chip select outputs enabled
	Slave mode			–	–	–
Consecutive mode	Master mode			INTC3n delay mode enabled/ disabled	Transfer wait disabled	Intermediate inactive level of chip select outputs disabled
					Transfer wait enabled	Intermediate inactive level of chip select outputs enabled
	Slave mode			–	–	

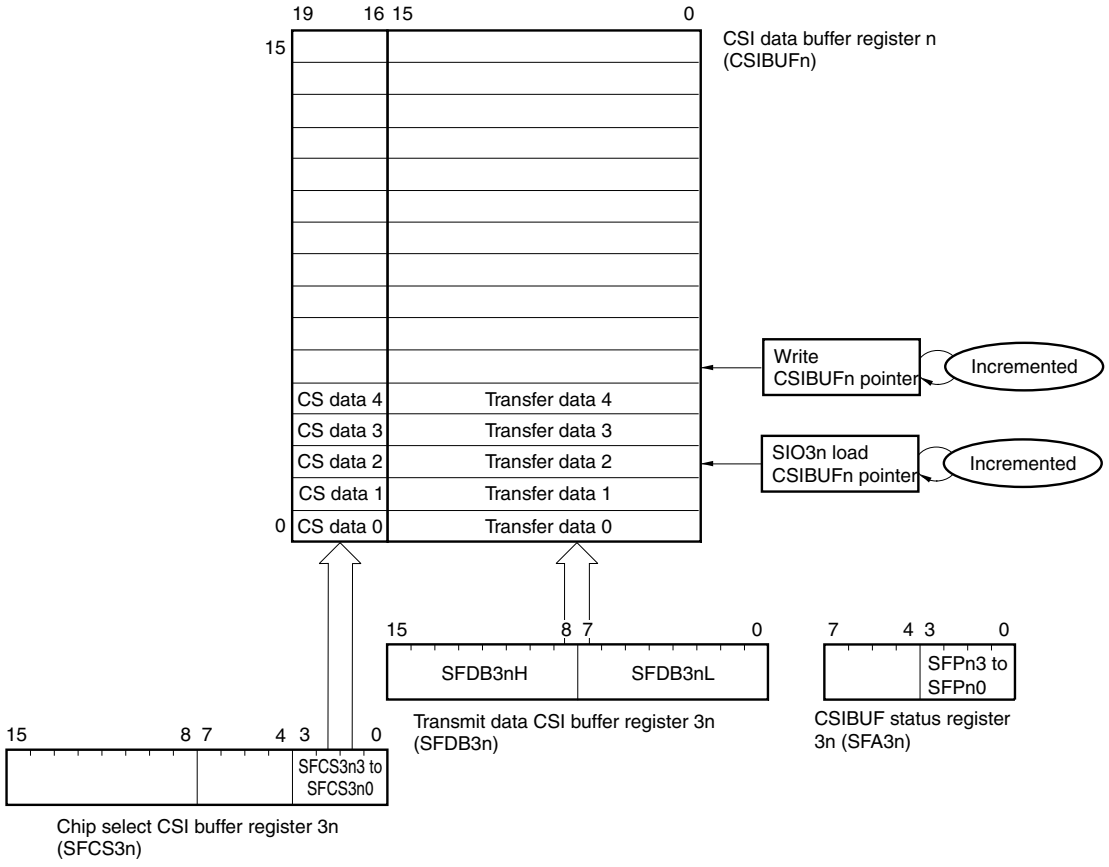
Remarks: 1. CTXEn bit: Bit 6 of CSIM3n register
CRXEn bit: Bit 5 of CSIM3n register
TRMDn bit: Bit 4 of CSIM3n register
DIRn bit: Bit 3 of CSIM3n register
CSITn bit: Bit 2 of CSIM3n register
CSWEn bit: Bit 1 of CSIM3n register
CSMDn bit: Bit 0 of CSIM3n register
CKS3n2 to CKS3n0 bits: Bits 2 to 0 of CSIC3n register

2. n = 0, 1

17.5.2 Function of CSI data buffer register (CSIBUFn)

By consecutively writing the transmit data to the SFCS3n register and the SFDB3n register from where it is transferred, the data can be stored in the CSIBUFn register while the CSIBUFn pointer for writing is automatically incremented (the CSIBUFn register size is 20 bits × 16) (n = 0, 1).
When the chip select outputs SCS3n0 to SCS3n3 are used, write SFCS3n register before the SFDB3n register. However, in slave mode the chip select outputs SCS3n0 to SCSS3n3 keep the inactive level and therefore writing to the SFCS3n register is not necessary.
The condition under which transfer is to be started (SFEMPn bit of the SFA3n register = 0) is satisfied when data is written to the lower 8 bits of the SFDB3n register (or SFDB3nL register). If a transfer data length of 9 bits or more is specified (CCLn3 to CCLn0 bits of the CSIL3n register = 0000B, or 1001B to 1111B), data must be written to the SFDB3n register in 16-bit units or to the SFDB3nH and SFDB3nL registers, in that order, in 8-bit units. If the transfer data length is set to 8 bits (CCLn3 to CCLn0 bits = 1000B), data must be written to the SFDB3nL register in 8-bit units or to the SFDB3n register in 16-bit units. If data is written to the SFDB3nL register in 16-bit units, however, the higher 8 bits of the data (of the SFDB3nH register) are ignored and not transferred.
The SFFULn bit of the SFA3n register is set to 1 when 16 data exist in the CSIBUFn register and outputs a CSIBUFn overflow interrupt (INTC3nOVF) when the SFFULn bit = 1 and when the 17th transfer data is written.
Sixteen data exist in the CSIBUFn register in the single mode (TRMDn bit of the CSIM3n register = 0) when “CSIBUFn pointer value for writing = CSIBUFn pointer value for SIO3n loading, and SFFULn bit = 1”. When the CSIBUFn pointer for SIO3n loading is incremented after completion of transfer, the CSIBUFn register has a vacancy of one data (in the consecutive mode (TRMDn bit = 1), the CSIBUFn register does not have a vacancy even if one data has been transferred).

Figure 17-11: Function of CSI Data Buffer Register n (CSIBUFn)



Remark: n = 0, 1

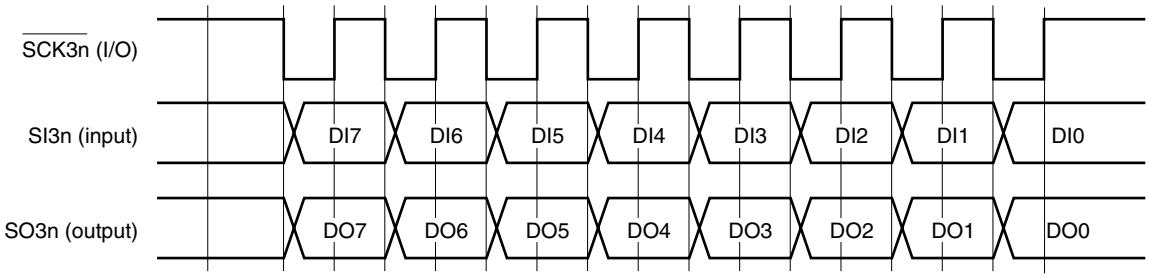
17.5.3 Data transfer direction specification function

The data transfer direction can be changed by using the DIRn bit of the CSIM3n register (n = 0, 1).

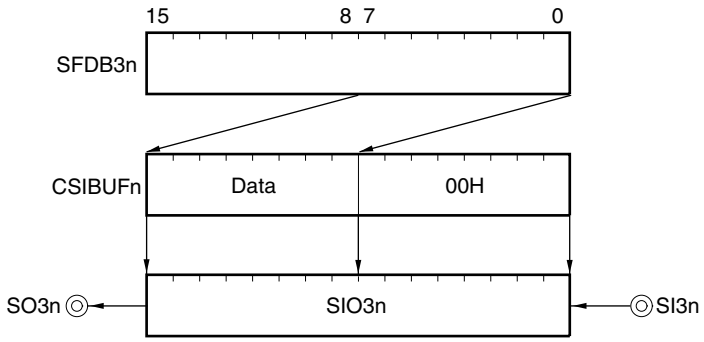
(1) MSB first (DIRn bit = 0)

Figure 17-12: Data Transfer Direction Specification (MSB first)

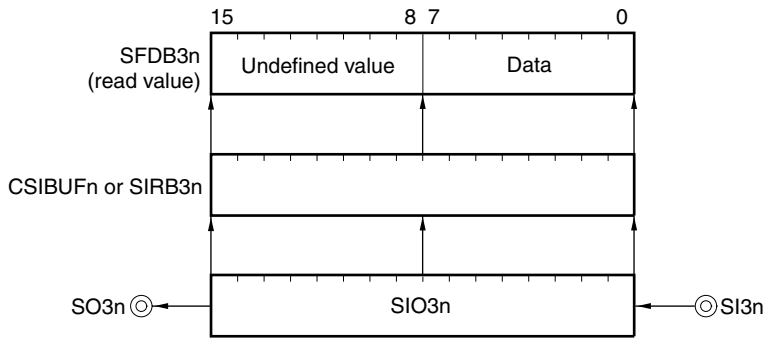
(a) Transfer direction: MSB first, Transfer data length: 8 Bits



(b) Writing from SFDB3n register to CSIBUFn register



(c) Reading from CSIBUFn register or SFDB3n register

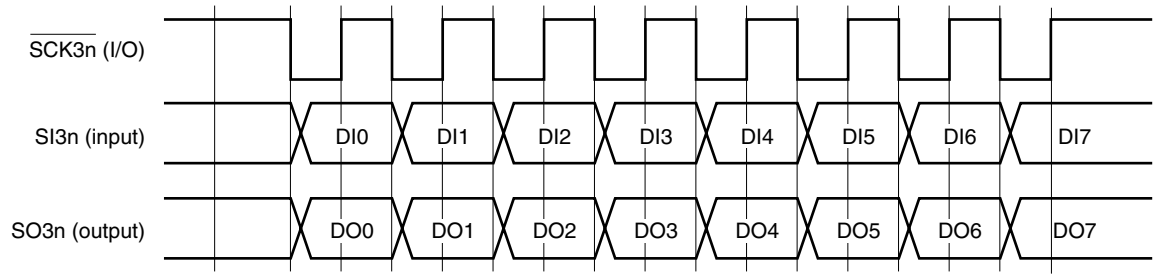


Remark: n = 0, 1

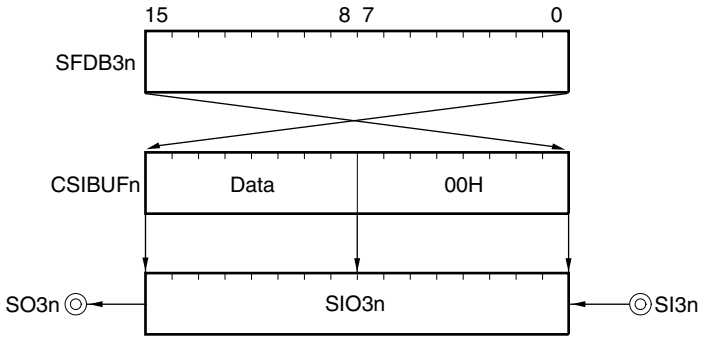
(2) LSB first (DIRn bit = 1)

Figure 17-13: Data Transfer Direction Specification (LSB first)

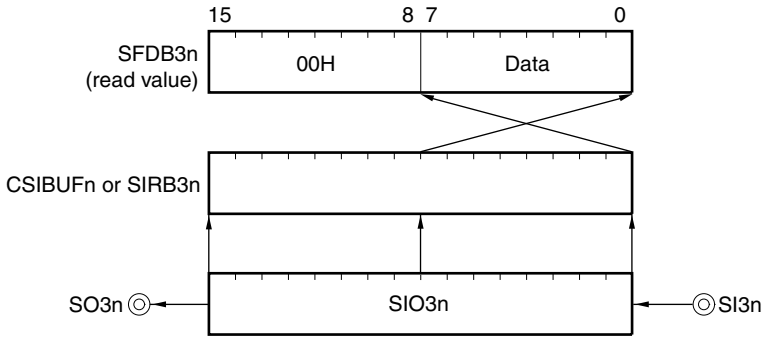
(a) Transfer direction: LSB first, Transfer data length: 8 Bits



(b) Writing from SFDB3n register to CSIBUFn register



(c) Reading from CSIBUFn register or SFDB3n register



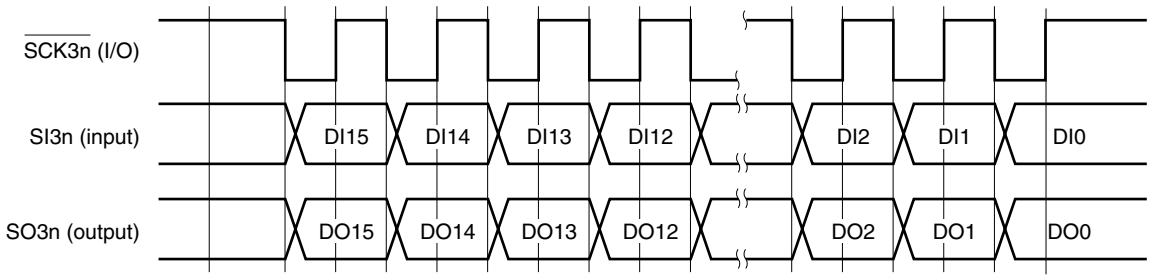
Remark: n = 0, 1

17.5.4 Transfer data length changing function

The transfer data length can be set from 8 to 16 bits in 1-bit units, by using the CCLn3 to CCLn0 bits of the CSIL3n register (n = 1, 0).

Figure 17-14: Transfer Data Length Changing Function

Transfer Data Length: 16 Bits (CCLn3 to CCLn0 Bits of CSIL3n Register = 0000B), Transfer Direction: MSB First (DIRn Bit of CSIM3n Register = 0)



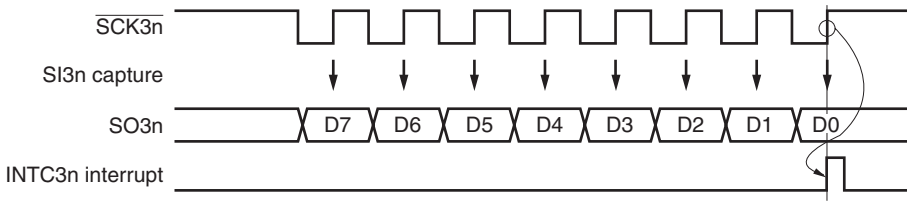
Remark: n = 0, 1

17.5.5 Function to select serial clock and data phase

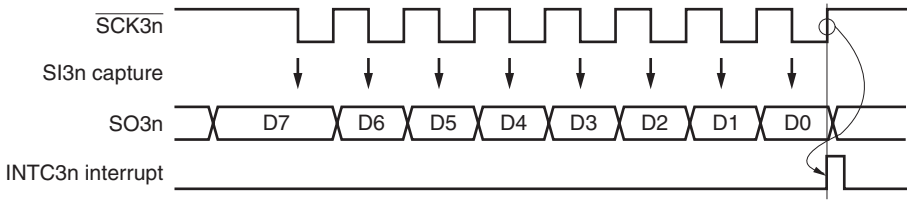
The serial clock and data phase can be changed by using the CKPn and DAPn bits of the CSIC3n register (n = 0, 1).

Figure 17-15: Clock Timing

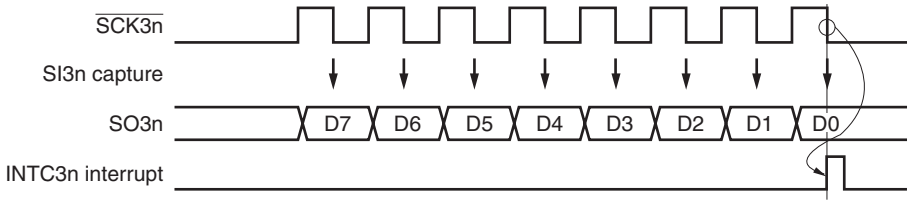
(a) When CKPn bit = 0, DAPn bit = 0



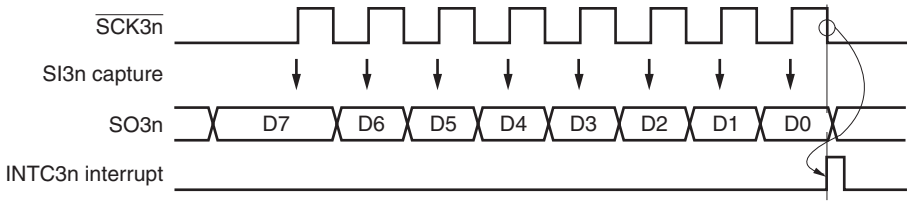
(b) When CKPn bit = 0, DAPn bit = 1



(c) When CKPn bit = 1, DAPn bit = 0



(d) When CKPn bit = 1, DAPn bit = 1



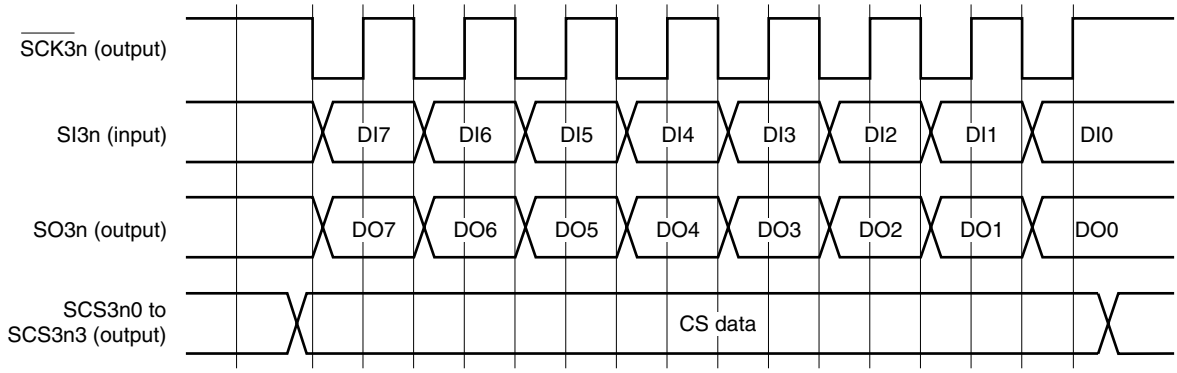
Remark: n = 0, 1

17.5.6 Master mode

The master mode is set and data is transferred with the transfer clock output to the $\overline{\text{SCK3n}}$ pin when the CKS3n2 to CKS3n0 bits of the CSIC3n register are set to a value other than 111B ($\overline{\text{SCK3n}}$ pin input is invalid) (n = 0, 1).
The default output level of the $\overline{\text{SCK3n}}$ pin is high when the CKPn bit of the CSIC3n register is 0, and low when the CKPn bit is 1.
In master mode the chip select outputs (SCS3n0 to SCS3n3) are effective.

Figure 17-16: Master Mode

CKPn and DAPn Bits of CSIC3n Register = 00B,
Active Level of CS Outputs: Low Level (CSLVn3 to CSLVn0 Bits of CSIL3n Register = 0000B)
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 Bits of CSIL3n Register = 1000B)



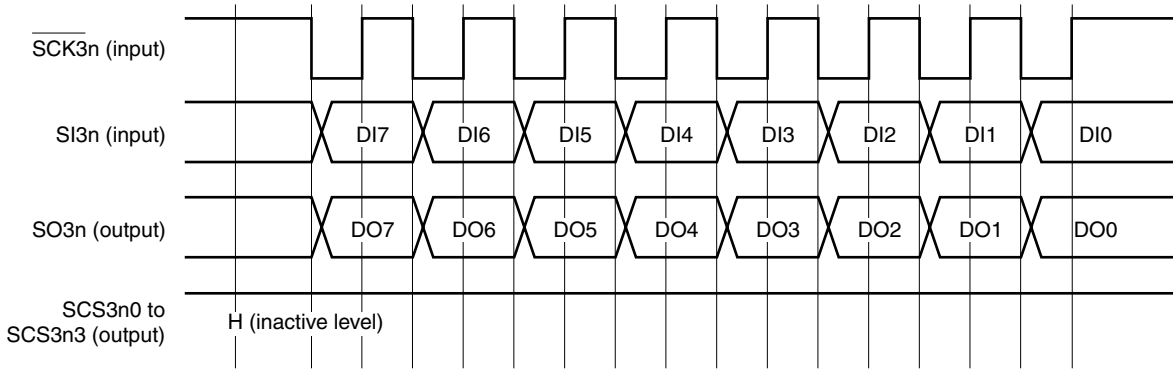
Remark: n = 0, 1

17.5.7 Slave mode

The slave mode is set when the CKS3n2 to CKS3n0 bits of the CSIC3n register are set to 111B, and data is transferred with the transfer clock input to the $\overline{\text{SCK3n}}$ pin (in the slave mode, it is recommended to set the MDLn2 to MDLn0 bits of the CSIC3n register to 000B and set the BRGn stop mode) (n = 0, 1).
The chip select outputs (SCS3n0 to SCS3n3) are ineffective in slave mode, the output levels are fixed to inactive level (chip select outputs are effective in master mode only).

Figure 17-17: Slave Mode

CKPn and DAPn Bits of CSIC3n Register = 00B,
Active Level of CS Outputs: Low Level (CSLVn3 to CSLVn0 Bits of CSIL3n Register = 0000B)
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 Bits of CSIL3n Register = 1000B)



The conditions under which data can be transferred in the slave mode are listed in the table below.

Table 17-2: Conditions Under Which Data Can Be Transferred in Slave Mode

Transfer Mode		CTXEn Bit	CRXEn Bit	CSIBUFn Register	SIRB3n Register and SIO3n Register
Single mode	Transmission mode	1	0	Data is in CSIBUFn register (SFEMPn bit = 0).	–
	Reception mode	0	1	Dummy data is in CSIBUFn register (SFEMPn bit = 0).	SIRB3n register or SIO3n register is empty.
	Transmission/ reception mode	1	1	Data is in CSIBUFn register (SFEMPn bit = 0).	
Consecutive mode	Transmission mode	1	0	Data is in CSIBUFn register (SFEMPn bit = 0).	–
	Reception mode	0	1	Dummy data is in CSIBUFn register (SFEMPn bit = 0).	–
	Transmission/ reception mode	1	1	Data is in CSIBUFn register (SFEMPn bit = 0).	–

Remarks: 1. CTXEn bit: Bit 6 of CSIM3n register
CRXEn bit: Bit 5 of CSIM3n register
SFEMPn bit: Bit 5 of SFA3n register

2. n = 0, 1

17.5.8 Transfer clock selection function

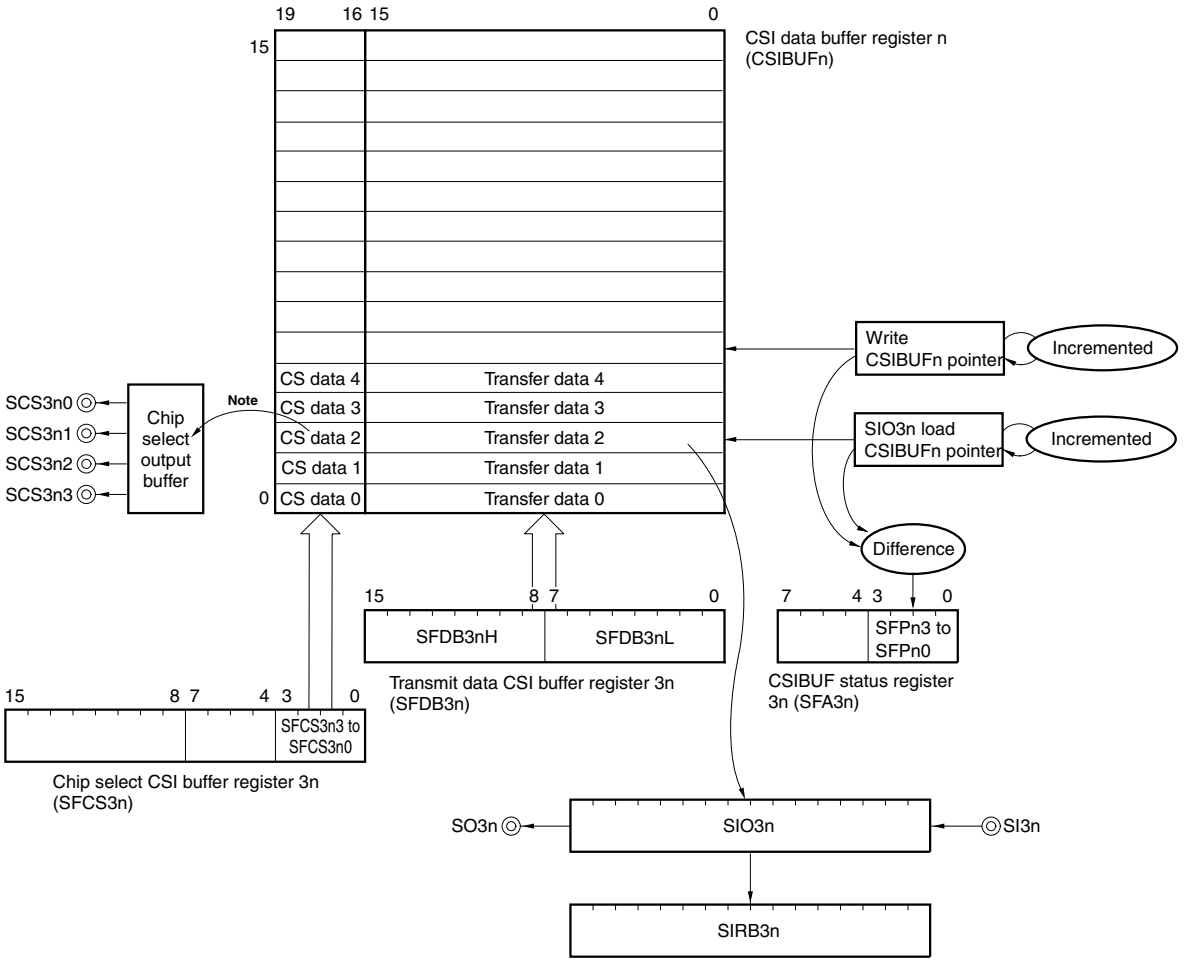
In the master mode (CKS3n2 to CKS3n0 bits of the CSIC3n register other than 111B), the bit transfer rate can be selected by setting the CKS3n2 to CKS3n0 and MDLn2 to MDLn0 bits of the CSIC3n register (ref. to **17.3 (2) Clocked serial interface clock select registers 30, 31 (CSIC30, CSIC31)**).

17.5.9 Single mode

The single mode is set when the TRMDn bit of the CSIM3n register is 0 (n = 0, 1).
In this mode, transfer is started when the CTXEn bit or CRXEn bit is set to 1 and when data is in the CSIBUFn register (SFEMPn bit = 0 in the SFA3n register).
If no data is in the CSIBUFn register (SFEMPn bit = 1), transfer is kept waiting until a given start condition is satisfied.
When data is written to the CSIBUFn register while the CTXEn or CRXEn bit is 1, the CSOTn bit of the SFA3n register (transfer status flag) is set to 1, and the chip select data (CS data) corresponding to SIO3n load CSIBUFn pointer is transferred to the chip select output buffer. However, in slave mode (CKS3n2 to CKS3n0 bits of the CSIC3n register = 111B) the chip select outputs (SCS3n0 to SCS3n3) keep always the inactive level.
If transfer is not in the wait status, the transfer data indicated by the SIO3n load CSIBUFn pointer is loaded from the CSIBUFn register to the SIO3n register, and transfer processing is started.
If the SIRB3n register is empty when one data has been transferred in the reception mode or transmission/reception mode, the received data is stored from the SIO3n register to the SIRB3n register, the transmission/reception completion interrupt (INTC3n) is output, and the SIO3n load CSIBUFn pointer is incremented. If the SIRB3n register is not empty, the next transfer processing is started. However, storing the receive data in the SIRB3n register, outputting the INTC3n interrupt, and incrementing the SIO3n load CSIBUFn pointer are held pending, until the previously received data is read from the SIRB3n register and the SIRB3n register becomes empty.
In the transmission mode, the INTC3n interrupt is output and the SIO3n load pointer is incremented when transfer processing of one data has been completed (the SIRB3n register is always empty because no data is stored from the SIO3n register to the SIRB3n register).
In all modes (transmission, reception, and transmission/reception modes), if the CSIBUFn register is empty (write CSIBUFn pointer value = SIO3n load CSIBUFn pointer value) when transfer processing of one data has been completed, the CSOTn bit is cleared to 0. The value of the “number of remaining data in the CSIBUFn register (write CSIBUFn pointer – SIO3n load pointer)” can always be read from the SFPn3 to SFPn0 bits of the SFA3n register.

Caution: When writing data to the SFDB3n register, be sure to confirm that the SFFULn bit of the SFA3n register is 0. Even if data is written to this register when SFFULn bit is 1, the CSIBUFn overflow interrupt (INTC3nOVF) is output, and the written data is ignored.

Figure 17-18: Single Mode



Note: Transfer of CS data will be performed in master mode only.

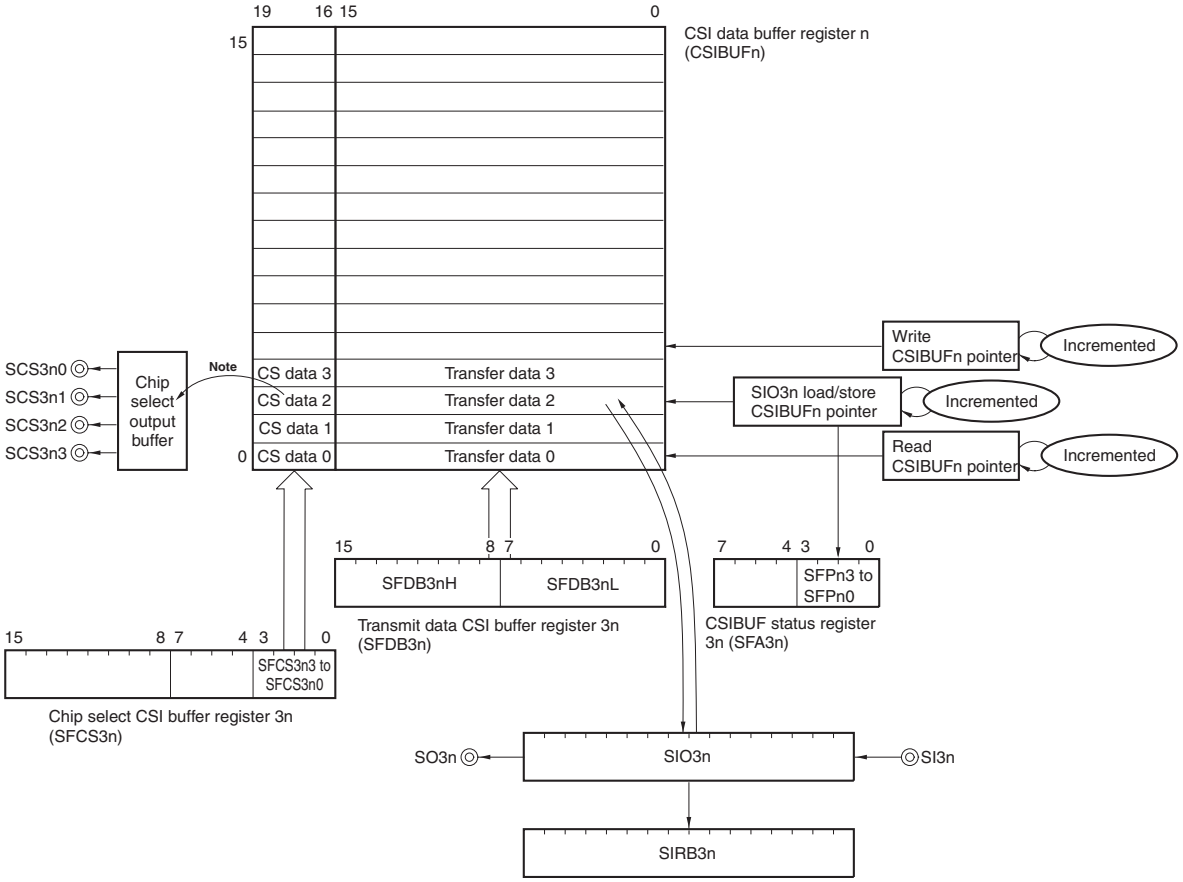
Remark: n = 0, 1

17.5.10 Consecutive mode

The consecutive mode is set when the TRMDn bit of the CSIM3n register is 1 (n = 0, 1).
In this mode, transfer is started when the CTXEn bit or CRXEn bit is 1 and when data is in the CSIBUFn register (SFEMPn bit of the SFA3n register = 0). At this time, set the number of transfer data in advance by using the SFNn3 to SFNn0 bits of the SFN3n register. Seventeen or more transfer data cannot be set. If 17 or more transfer data are written to the CSIBUFn register, the excess data are ignored and not transferred. Do not write data exceeding the number of transfer data specified by the SFNn3 to SFNn0 bits of the SFN3n register to the CSIBUFn register.
If no data is in the CSIBUFn register (SFEMPn bit = 1), transfer is kept waiting until a given start condition is satisfied.
If data is written to the CSIBUFn register when the CTXEn or CRXEn bit is 1, the CSOTn bit (transfer status flag) of the SFA3n register is set to 1 and the chip select data (CS data) according to the SIO3n load/store CSIBUFn pointer is transferred to the chip select output buffer. However, in slave mode (CKS3n2 to CKS3n0 bits of the CSIC3n register = 111B) the chip select outputs (SCS3n0 to SCS3n3) keep always the inactive level.
If transfer is not in the wait status, the transfer data indicated by the SIO3n load/store CSIBUFn pointer is loaded from the CSIBUFn register to SIO3n register. Then transfer processing is started.
When transfer processing of one data is completed in the reception mode or transmission/reception mode, the received data is overwritten from the SIO3n register to the transfer data in the CSIBUFn register indicated by the SIO3n load/store CSIBUFn pointer, and then the pointer is incremented. By consecutively reading the transfer data from the SIRB3n register after all data in the CSIBUFn register have been transferred (when the INTC3n interrupt has occurred), the receive data can be sequentially read while the read CSIBUFn pointer is incremented.
In the transmission mode, the SIO3n load/store CSIBUFn pointer is incremented when transfer processing of one data has been completed.
In all modes (transmission, reception, and transmission/reception modes), when data has been transferred by the value set by the SFNn3 to SFNn0 bits of the SFN3n register, the CSOTn bit is cleared to 0 and the transmission/reception completion interrupt (INTC3n) is output.
To transfer the next data, be sure to write 1 to the FPCLRn bit of the SFA3n register and clear all the CSIBUFn pointers to 0.
The “number of transferred data (SIO3n load/store CSIBUFn pointer value)” can always be read from the SFPn3 to SFPn0 bits of the SFA3n register.

Caution: The SFA3n register is in the same status when transfer data is written (before start of transfer) after the CSIBUFn pointer is cleared (FPCLRn bit of the SFA3n register = 1) and when 16 data have been transferred (SFFULn bit = 0, SFEMPn bit = 1, SFPn3 to SFPn0 bits = 0000B).

Figure 17-19: Consecutive Mode



Note: Transfer of CS data will be performed in master mode only.

Remark: n = 0, 1

17.5.11 Transmission mode

The transmission mode is set when the CTXEn bit of the CSIM3n register is set to 1 and the CRXEn bit is cleared to 0. In this mode, transmission is started by a trigger that writes transmit data to the SFDB3n register or sets the CTXEn bit to 1 when transmit data is in the SFDB3n register (n = 0, 1). Even in the single mode (TRMDn bit of the CSIM3n register = 0), whether the SIRB3n or SIO3n register is empty has nothing to do with starting transmission. The value input to the SI3n pin during transmission is latched in the shift register (SIO3n) but is not transferred to the SIRB3n and CSIBUFn registers at the end of transmission.
The transmission/reception completion interrupt (INTC3n) occurs immediately after data is sent out from the SIO3n register.

17.5.12 Reception mode

The reception mode is set when the CTXEn bit of the CSIM3n register is cleared to 0 and CRXEn bit is set to 1. In this mode, reception is started by using the processing of writing dummy data to the SFDB3n register as a trigger (n = 0, 1). In the single mode (TRMDn bit of the CSIM3n register = 0), however, the condition of starting reception includes that the SIRB3n or SIO3n register is empty. (If reception to the SIO3n register is completed when the previously received data is held in the SIRB3n register without being read, the previously received data is read from the SIRB3n register and the wait status continues until the SIRB3n register becomes empty.)
The SO3n pin outputs a low level.
The transmission/reception completion interrupt (INTC3n) occurs immediately after receive data is transferred from the SIO3n register to the SIRB3n register.

17.5.13 Transmission/reception mode

The transmission/reception mode is set when both the CTXEn and CRXEn bits of the CSIM3n register are set to 1. In this mode, transmission/reception is started by using the processing to write transmit data to the SFDB3n register as a trigger (n = 0, 1). In the single mode (TRMDn bit of the CSIM3n register = 0), however, the condition of starting transmission/reception includes that the SIRB3n or SIO3n register is empty. (If reception to the SIO3n register is completed when the previously received data is held in the SIRB3n register without being read, the previously received data is read from the SIRB3n register and the wait status continues until the SIRB3n register becomes empty.)

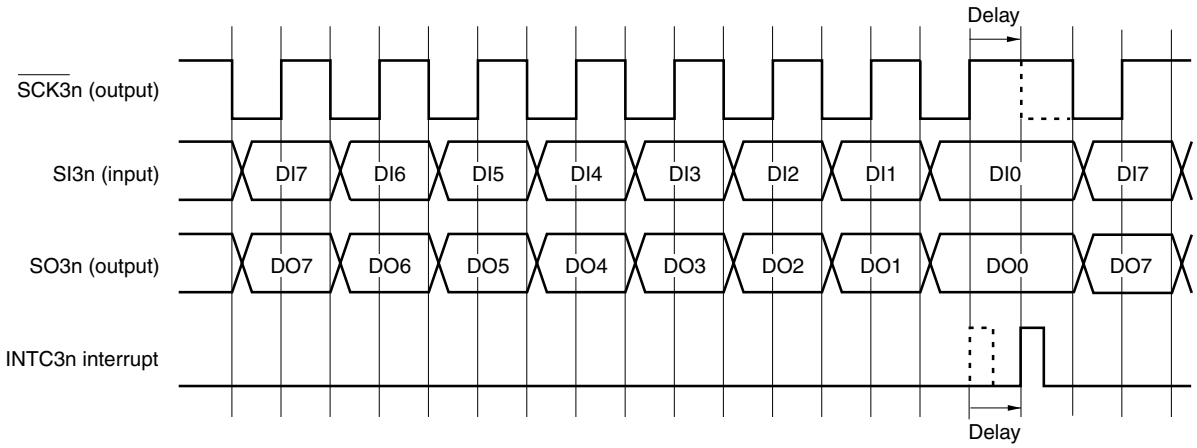
17.5.14 Delay control of transmission/reception completion interrupt (INTC3n)

In the master mode (CKS3n2 to CKS3n0 bits of the CSIC3n register other than 111B), occurrence of the transmission/reception completion interrupt (INTC3n) can be delayed by half a clock (1/2 serial clock), depending on the setting of the CSITn bit of the CSIM3n register (CSITn bit = 1). The CSITn bit is valid only in the master mode. In the slave mode (CKS3n2 to CKS3n0 bits = 111B), setting the CSITn bit to 1 is prohibited (even if set, the INTC3n interrupt is not affected).

Caution: If the CSITn bit of the CSIM3n register is set to 1 in the consecutive mode (TRMDn bit of the CSIM3n register = 1), the INTC3n interrupt is not output at the end of data other than the last data set by the SFNn3 to SFNn0 bits of the SFN3n register, but a delay of half a clock can be inserted between each data transfer.

Figure 17-20: Delay Control of Transmission/Reception Completion Interrupt (INTC3n):

CSITn Bit of the CSIM3n Register = 1,
CSWEn Bit of the CSIM3n Register = 0,
CKPn and DAPn Bits of the CSIC3n Register = 00B,
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 Bits of the CSIL3n Register = 1000B)



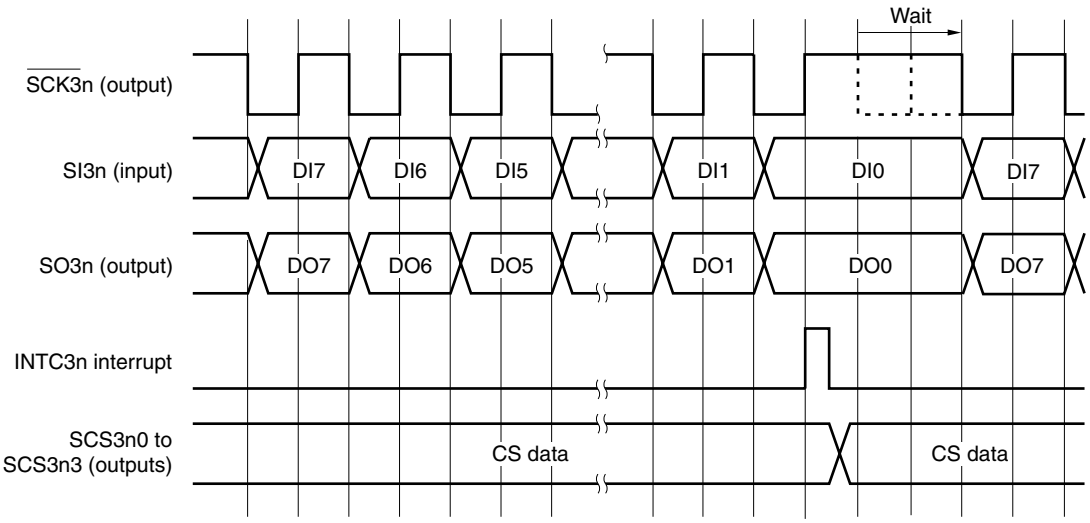
Remark: n = 0, 1

17.5.15 Transfer wait function

In the master mode (CKS3n2 to CKS3n0 bits of the CSIC3n register other than 111B), starting transfer can be delayed by one clock, depending on the setting of the CSWEn bit of the CSIM3n register (CSWEn bit = 1). The CSWEn bit is valid only in the master mode. In the slave mode (CKS3n2 to CKS3n0 bits = 111B), setting the CSWEn bit to 1 is prohibited (even if set, transfer wait is not inserted).
When the transfer wait function is enabled (CSWEn bit = 1), the chip select outputs can be
During transfer wait (CSWE bit = 1) the chip select outputs (SCS3n0 to SCS3n3) can be configured for an intermediate inactive level output of half a clock period by setting the CSMDn bit of the CSIM3n register to 1.

Figure 17-21: Transfer Wait Function (1/3)

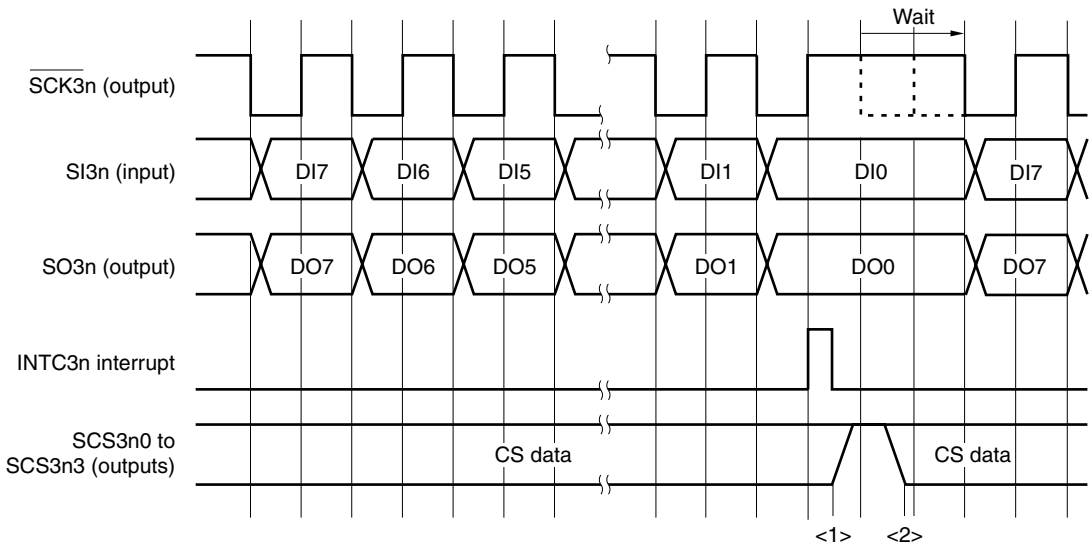
(a) Transfer Wait Enabled (CSWEn Bit = 1),
INTC3n Delay Disabled (CSITn Bit = 0),
CKPn and DAPn Bits = 00B,
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
Intermediate Inactive Chip Select Level Disabled (CSMDn = 0)



Remark: n = 0, 1

Figure 17-21: Transfer Wait Function (2/3)

(b) Transfer Wait Enabled (CSWEn Bit = 1), INTC3n Delay Disabled (CSITn Bit = 0), CKPn and DAPn Bits = 00B, Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B), Intermediate Inactive Chip Select Level Enabled (CSMDn = 1)

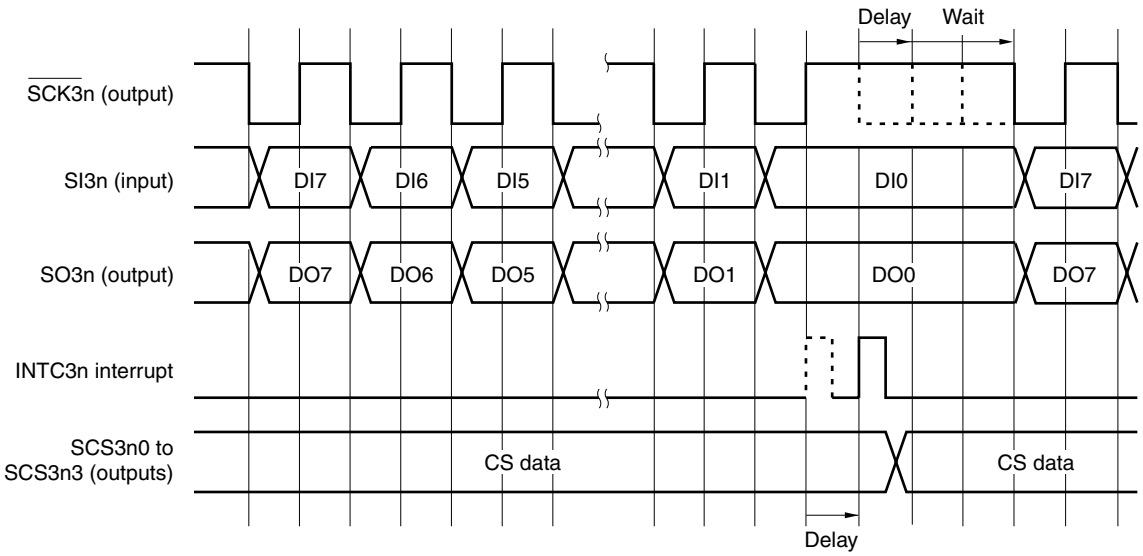


Remarks: 1. When the CSIBUFn register is empty at the time of <1>, the chip select pins output an inactive level and maintain it.
When the CSIBUFn register is not empty at the time of <1>, the chip select pins output an inactive level up to the time of <2>, and output subsequently the succeeding chip select data
Moreover, in single mode (TRMDn bit of the CSIM3n register = 0) the chip select pins output an inactive level from the time <1> and held it pending until the previously receive data is read from the SIRB3n register and the SIRB3n register becomes empty.

2. n = 0, 1

Figure 17-21: Transfer Wait Function (3/3)

(c) Transfer Wait Enabled (CSWEn Bit = 1), INTC3n Delay Enabled (CSITn Bit = 1), CKPn and DAPn Bits = 00B, Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B), Intermediate Inactive Chip Select Level Disabled (CSMDn = 0)



Remark: n = 0, 1

17.5.16 Output pins

(1) $\overline{\text{SCK3n}}$ pin

The $\overline{\text{SCK3n}}$ pin outputs a high level when both the CTXEn and CRXEn bits of the CSIM3n register are 0 (n = 0, 1).

In the master mode (CKS3n2 to CKS3n0 bits = other than 111 in the CSIC3n register), this pin outputs the default level when the FPCLRn bit of the SFA3n register is set to 1.

In the slave mode (CKS3n2 to CKS3n0 bits = 111 in the CSIC3n register), the default output level of the $\overline{\text{SCK3n}}$ pin is fixed to the high level.

Table 17-3: Default Output Level of $\overline{\text{SCK3n}}$ Pin

CKPn Bit	CKS3n2 to CKS3n0 Bits	Default Output Level of $\overline{\text{SCK3n}}$ Pin
0	111B (slave mode)	High level[Note]
	Other than 111B (master mode)	High level
1	111B (slave mode)	High level
	Other than 111B (master mode)	Low level

Note: Default value after reset, or value when CSICAEn bit of the CSIM3n register is cleared to 0.

Remarks: 1. The output of the $\overline{\text{SCK3n}}$ pin changes if the CKPn bit is rewritten in the master mode.

2. n = 0, 1

(2) SO3n pin

The SO3n pin outputs a low level when both the CTXEn and CRXEn bits of the CSIM3n register are 0 (n = 0, 1).

This pin outputs a low level when the FPCLRn bit of the SFA3n register is set to 1 (the previous value is retained only in the slave mode (CKS3n2 to CKS3n0 bits of the CSIC3n register = 111B) and when the DAPn bit of the CSIC3n register is 0).

Table 17-4: Default Output Level of SO3n Pin

Default Output Level of SO3n Pin
Low level[Note]

Note: Default value after reset, or value when CSICAEn bit of the CSIM3n register is cleared to 0

Remark: n = 0, 1

(3) SCS3n0 to SCS3n3 pins

The SCS3n0 to SCS3n3 pins output the default level when both the CTXEn and CRXEn bits of the CSIM3n register are 0, or when the CSICAEn bit of the CSIM3n register is cleared to 0 (n = 0, 1). These pins output the default level when the FPCLRn bit of the SFA3n register is set to 1. In slave mode these pins output always the default level (inactive level).

Table 17-5: Default Output Level of SCS3n0 to SCS3n3 Pins

CSLVn Bit	Default Output Level of SCS3n0 to SCS3n3 Pins
0	High level**Note**
1	Low level

Note: Default value after reset.

Remark: n = 0, 1

17.5.17 CSIBUFn overflow interrupt signal (INTC3nOVF)

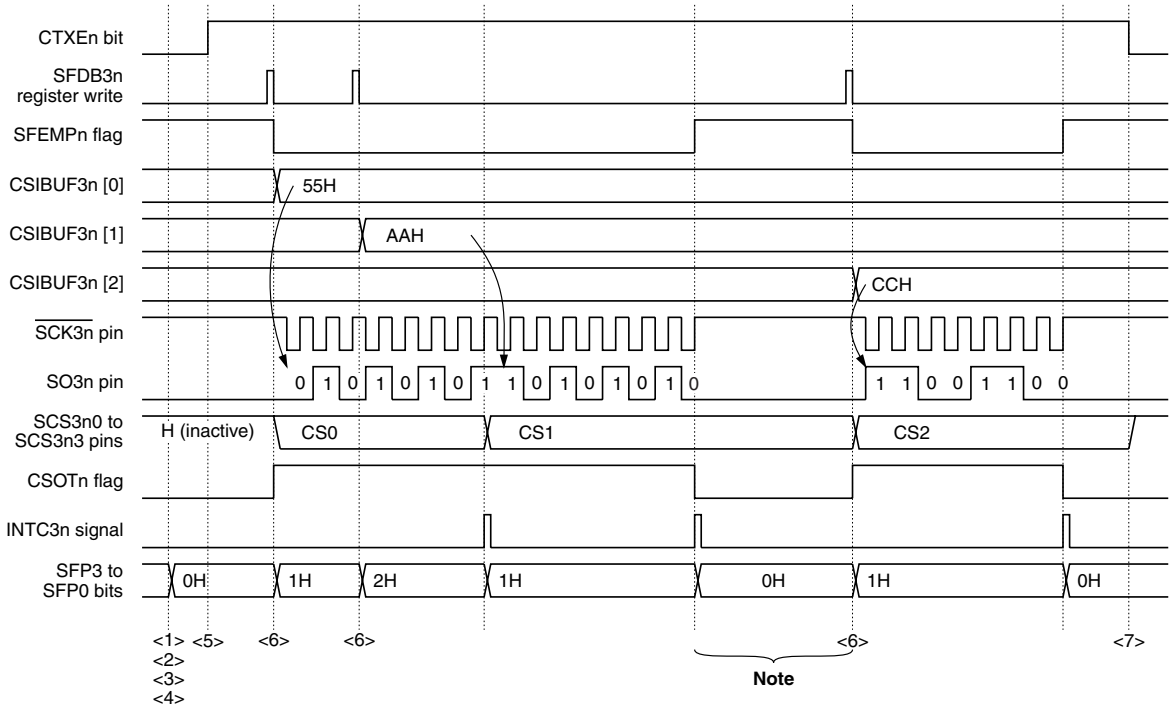
The INTC3nOVF interrupt is output when 16 data exist in the CSIBUFn register and when the 17th data is written (to the SFDB3n or SFDB3nL register). The 17th data is not written but ignored.
In the single mode (TRMDn bit of the CSIM3n register = 0), 16 data exist in the CSIBUFn register when "write CSIBUFn pointer value = SIO3n load CSIBUFn pointer value" and SFFULn bit of the SFA3n register = 1. When transfer is completed and the SIO3n load CSIBUFn pointer is incremented, the CSIBUFn register has one vacancy (the CSIBUFn register has no vacancy even when transfer of one data has been completed in the consecutive mode (TRMDn bit = 1)).

17.6 Operating Procedures

17.6.1 Single mode (master mode, transmission mode)

Figure 17-22: Single Mode (Master Mode, Transmission Mode)

MSB First (DIR bit = 0), CKP bit = 0, DAP bit = 0
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
INTC3n Interrupt Not Delayed (CSIT bit = 0),
Transfer Wait: Disabled (CSWE bit = 0),
Chip Select Active Level: L-Level (CSLVn3 to CSLVn0 bits = 0000B)



Note: During this period a reception from the slave is put on hold until at least one transmit data has been loaded to the CSIBUFn register by writing the SFDB3n register (SFEMPn flag of SFA3n register = 0) in order to start the transfer.

Remark: n = 0, 1

<1> When the CSICAEn bit of the CSIM3n register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission by setting the CTXEn bit to 1.
<6> Confirm that the SFFULn bit of the SFA3n register is 0, and then write first CS data to the SFCS3n register and subsequently write transfer data to the SFDB3n register.
If it is clearly known that the SFFULn bit is 0 because transfer data is written to that bit by the interrupt servicing routine of INTC3n, it is not always necessary to confirm that the SFFULn bit is 0.
<7> Confirm that the INTC3n interrupt has occurred and the SFEMPn bit of the SFA3n register is 1, and disable transmission by clearing the CTXEn bit of the CSIM3n register to 0 (end of transmission).

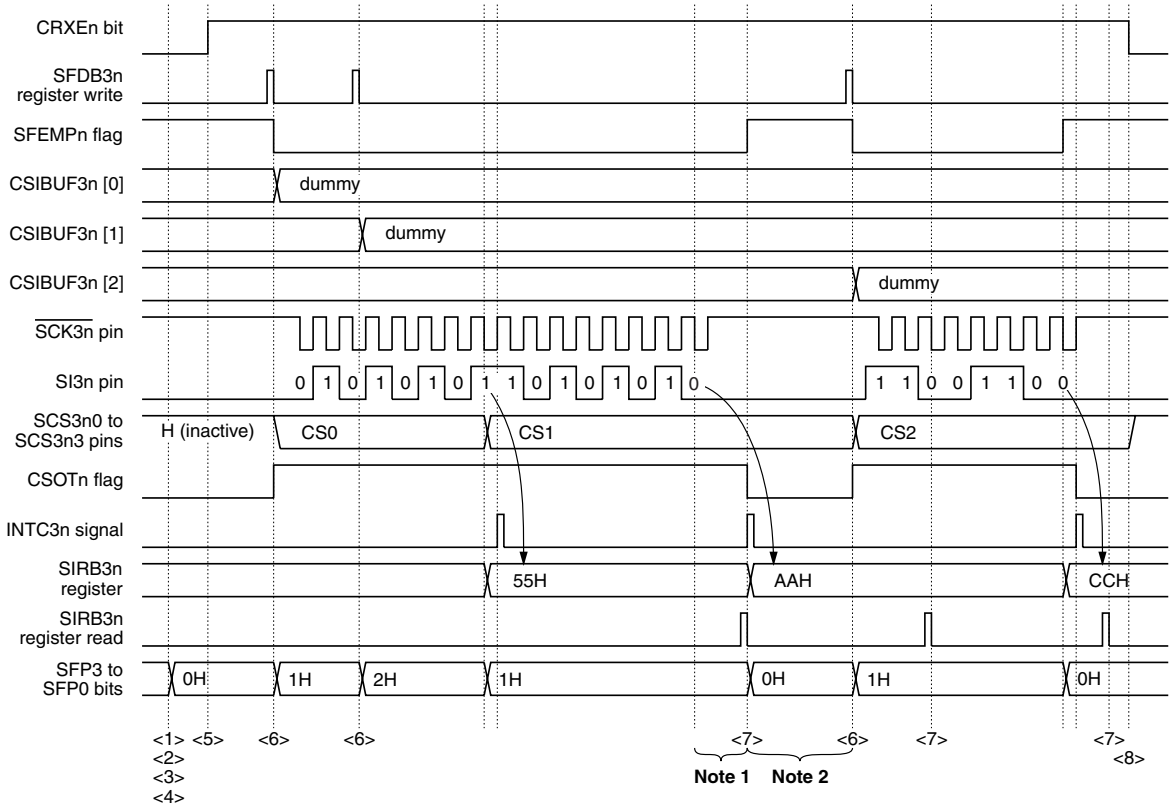
Remarks: 1. To execute a further transfer, repeat <6> before <7>.

2. n = 0, 1

17.6.2 Single mode (master mode, reception mode)

Figure 17-23: Single Mode (Master Mode, Reception Mode)

MSB First (DIR bit = 0), CKP bit = 1, DAP bit = 1
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
INTC3n Interrupt Not Delayed (CSIT bit = 0),
Transfer Wait: Disabled (CSWE bit = 0),
Chip Select Active Level: L-Level (CSLVn3 to CSLVn0 bits = 0000B)



Notes: 1. While the SIRB3n register is full a new transfer start of reception from the slave is put on hold until the SIRB3n register is read.

2. During this period a reception from the slave is put on hold until at least one dummy transmit data has been loaded to the CSIBUFn register by writing the SFDB3n register (SFEMPn flag of SFA3n register = 0) in order to start the transfer.

Remark: n = 0, 1

<1> When the CSICAEn bit of the CSIM3n register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable reception by setting the CRXEn bit to 1.
<6> Confirm that the SFFULn bit of the SFA3n register is 0, and then write first CS data to the SFCS3n register and subsequently write dummy transfer data to the SFDB3n register (reception start trigger).
If it is clearly known that the SFFULn bit is 0 because dummy transfer data is written to that bit by the interrupt servicing routine of INTC3n, it is not always necessary to confirm that the SFFULn bit is 0.
<7> Confirm that the INTC3n interrupt has occurred, and then read the SIRB3n register.
<8> Confirm that the INTC3n interrupt has occurred and the SFEMPn bit is 1, and disable reception by clearing the CRXEn bit of the CSIM3n register to 0 (end of reception).

Remarks: 1. To execute a further transfer, repeat <6> and <7> before <8>.
Perform writing dummy transfer data in <6> every time transfer is executed.

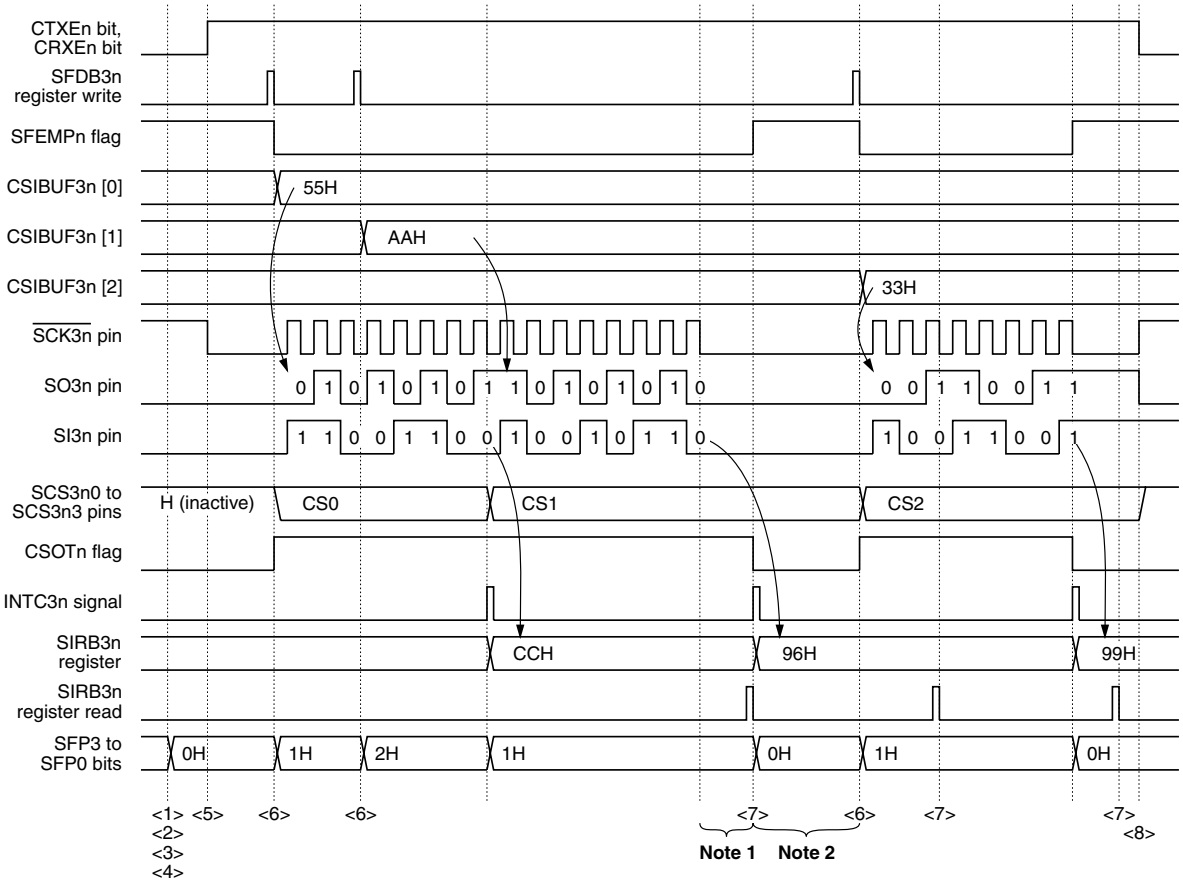
2. The SO3n pin output is fixed to low level (default value).

3. n = 0, 1

17.6.3 Single mode (master mode, transmission/reception mode)

Figure 17-24: Single Mode (Master Mode, Transmission/Reception Mode)

MSB First (DIR bit = 0), CKP bit = 1, DAP bit = 0
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
INTC3n Interrupt Not Delayed (CSIT bit = 0),
Transfer Wait: Disabled (CSWE bit = 0),
Chip Select Active Level: L-Level (CSLVn3 to CSLVn0 bits = 0000B)



Notes: 1. While the SIRB3n register is full a new transfer start of reception from the slave is put on hold until the SIRB3n register is read.

2. During this period a reception from the slave is put on hold until at least one transmit data has been loaded to the CSIBUFn register by writing the SFDB3n register (SFEMPn flag of SFA3n register = 0) in order to start the transfer.

Remark: n = 0, 1

<1> When the CSICAEn bit of the CSIM3n register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission/reception by setting the CTXEn and CRXEn bits to 1.
<6> Confirm that the SFFULn bit of the SFA3n register is 0, and then write first CS data to the SFCS3n register and subsequently write transfer data to the SFDB3n register.
If it is clearly known that the SFFULn bit is 0 because transfer data is written to that bit by the interrupt servicing routine of INTC3n, it is not always necessary to confirm that the SFFULn bit is 0.
<7> Confirm that the INTC3n interrupt has occurred, and then read the SIRB3n register.
<8> Confirm that the INTC3n interrupt has occurred and the SFEMPn bit is 1, and disable transmission/reception by clearing the CTXEn and CRXEn bits of the CSIM3n register to 0 (end of transmission/reception).

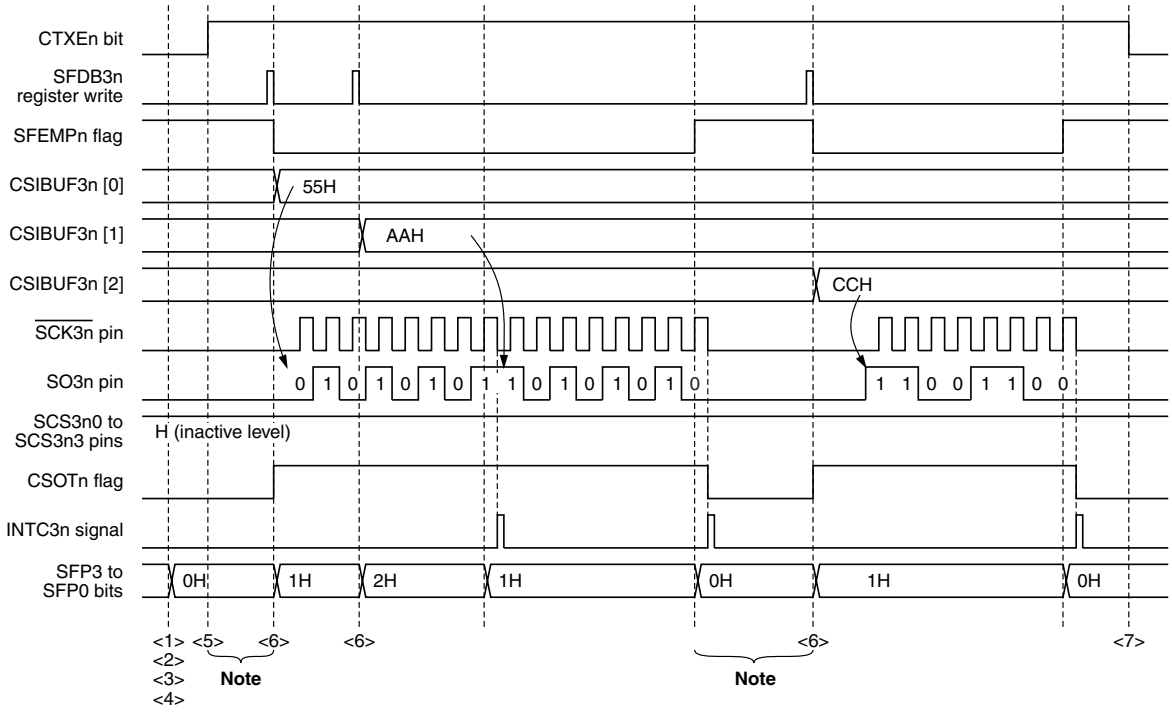
Remarks: 1. To execute a further transfer, repeat <6> before <7>.

2. n = 0, 1

17.6.4 Single mode (slave mode, transmission mode)

Figure 17-25: Single Mode (Slave Mode, Transmission Mode)

MSB First (DIR bit = 0), CKP bit = 1, DAP bit = 1
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
INTC3n Interrupt Not Delayed (CSIT bit = 0),
Transfer Wait: Disabled (CSWE bit = 0),
Chip Select Active Level: L-Level (CSLVn3 to CSLVn0 bits = 0000B)



Note: During this period a transmission to the master will be ignored until at least one transmit data is loaded to the CSIBUFn register by writing the SFDB3n register (SFEMPn flag of SFA3n register = 0).

Remark: n = 0, 1

<1> When the CSICAEn bit of the CSIM3n register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission by setting the CTXEn bit to 1.
<6> Confirm that the SFFULn bit of the SFA3n register is 0, and then write transfer data to the SFDB3n register.
Since the chip select outputs (SCS3n0 to SCS3n3) are ineffective in the slave mode and always output the inactive level, writing of CS data to the SFCS3n register is not necessary.
If it is clearly known that the SFFULn bit is 0 because transfer data is written to that bit by the interrupt servicing routine of INTC3n, it is not always necessary to confirm that the SFFULn bit is 0.
<7> Confirm that the INTC3n interrupt has occurred and the SFEMPn bit is 1, and disable transmission by clearing the CTXEn bit of the CSIM3n register to 0 (end of transmission).

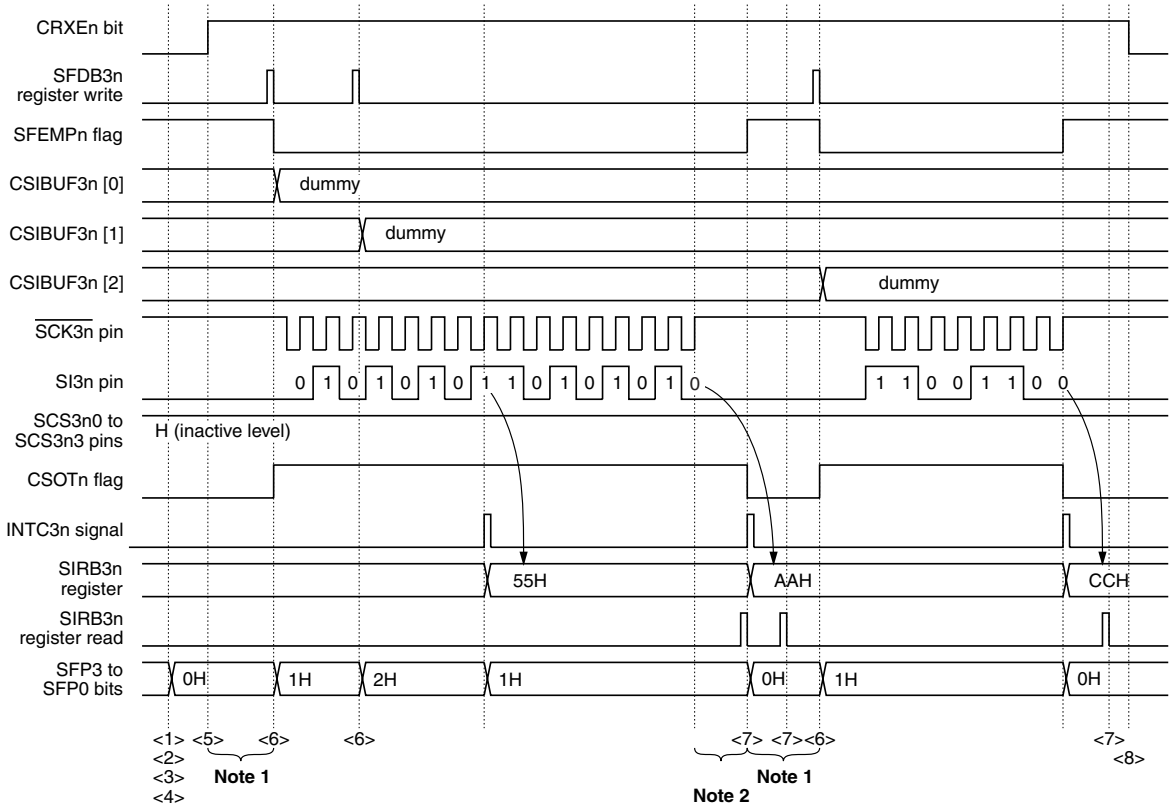
Remarks: 1. To execute a further transfer, repeat <6> before <7>.

2. n = 0, 1

17.6.5 Single mode (slave mode, reception mode)

Figure 17-26: Single Mode (Slave Mode, Reception Mode)

MSB First (DIR bit = 0), CKP bit = 0, DAP bit = 0
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
INTC3n Interrupt Not Delayed (CSIT bit = 0),
Transfer Wait: Disabled (CSWE bit = 0),
Chip Select Active Level: L-Level (CSLVn3 to CSLVn0 bits = 0000B)



Notes: 1. During this period a transmission/reception from the master will be ignored until at least one dummy transmit data is loaded to the CSIBUFn register by writing the SFDB3n register (SFEMPn flag of SFA3n register = 0).

2. While the SIRB3n register is full a new reception from the master will be ignored until the SIRB3n register is read.

Remark: n = 0, 1

<1> When the CSICAEn bit of the CSIM3n register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable reception by setting the CRXEn bit to 1.
<6> Confirm that the SFFULn bit of the SFA3n register is 0, and then write dummy transfer data to the SFDB3n register (reception start trigger).
Since the chip select outputs (SCS3n0 to SCS3n3) are ineffective in the slave mode and always output the inactive level, writing of CS data to the SFCS3n register is not necessary.
If it is clearly known that the SFFULn bit is 0 because dummy transfer data is written to that bit by the interrupt servicing routine of INTC3n, it is not always necessary to confirm that the SFFULn bit is 0.
<7> Confirm that the INTC3n interrupt has occurred, and then read the SIRB3n register.
<8> Confirm that the INTC3n interrupt has occurred and the SFEMPn bit is 1, and disable reception by clearing the CRXEn bit of the CSIM3n register to 0 (end of reception).

Remarks: 1. To execute a further transfer, repeat <6> and <7> before <8>.
Perform writing dummy transfer data in <6> every time transfer is executed.

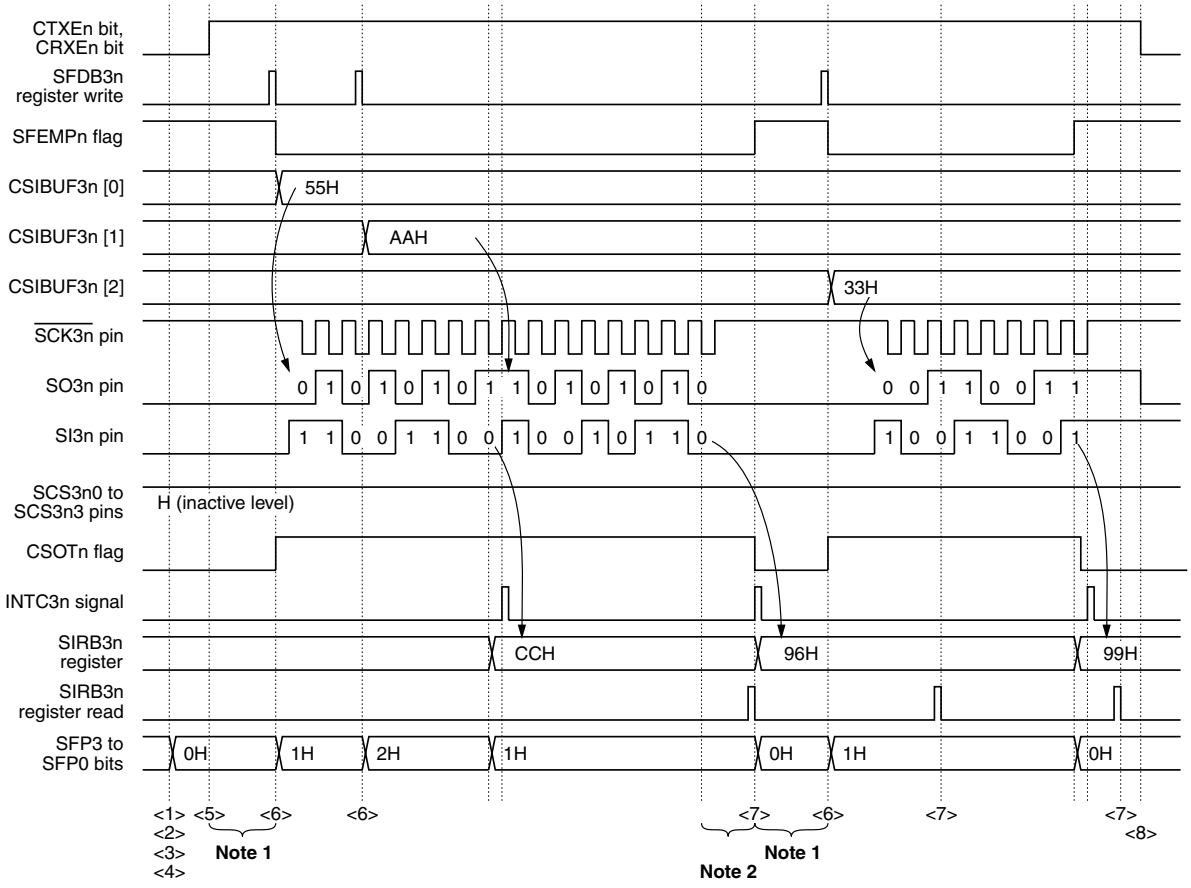
2. The SO3n pin output is fixed to low level (default value).

3. n = 0, 1

17.6.6 Single mode (slave mode, transmission/reception mode)

Figure 17-27: Single Mode (Slave Mode, Transmission/Reception Mode)

MSB First (DIR bit = 0), CKP bit = 0, DAP bit = 1
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
INTC3n Interrupt Not Delayed (CSIT bit = 0),
Transfer Wait: Disabled (CSWE bit = 0),
Chip Select Active Level: L-Level (CSLVn3 to CSLVn0 bits = 0000B)



Notes: 1. During this period a transmission/reception from the master will be ignored until at least one transmit data is loaded to the CSIBUFn register by writing the SFDB3n register (SFEMPn flag of SFA3n register = 0).

2. While the SIRB3n register is full a new transmission/reception from the master will be ignored until the SIRB3n register is read.

Remark: n = 0, 1

<1> When the CSICAEn bit of the CSIM3n register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission/reception by setting the CTXEn and CRXEn bits to 1.
<6> Confirm that the SFFULn bit of the SFA3n register is 0, and then write transfer data to the SFDB3n register.
Since the chip select outputs (SCS3n0 to SCS3n3) are ineffective in the slave mode and always output the inactive level, writing of CS data to the SFCS3n register is not necessary.
If it is clearly known that the SFFULn bit is 0 because transfer data is written to that bit by the interrupt servicing routine of INTC3n, it is not always necessary to confirm that the SFFULn bit is 0.
<7> Confirm that the INTC3n interrupt has occurred, and then read the SIRB3n register.
<8> Confirm that the INTC3n interrupt has occurred and the SFEMPn bit is 1, and disable transmission/reception by clearing the CTXEn and CRXEn bits of the CSIM3n register to 0 (end of transmission/reception).

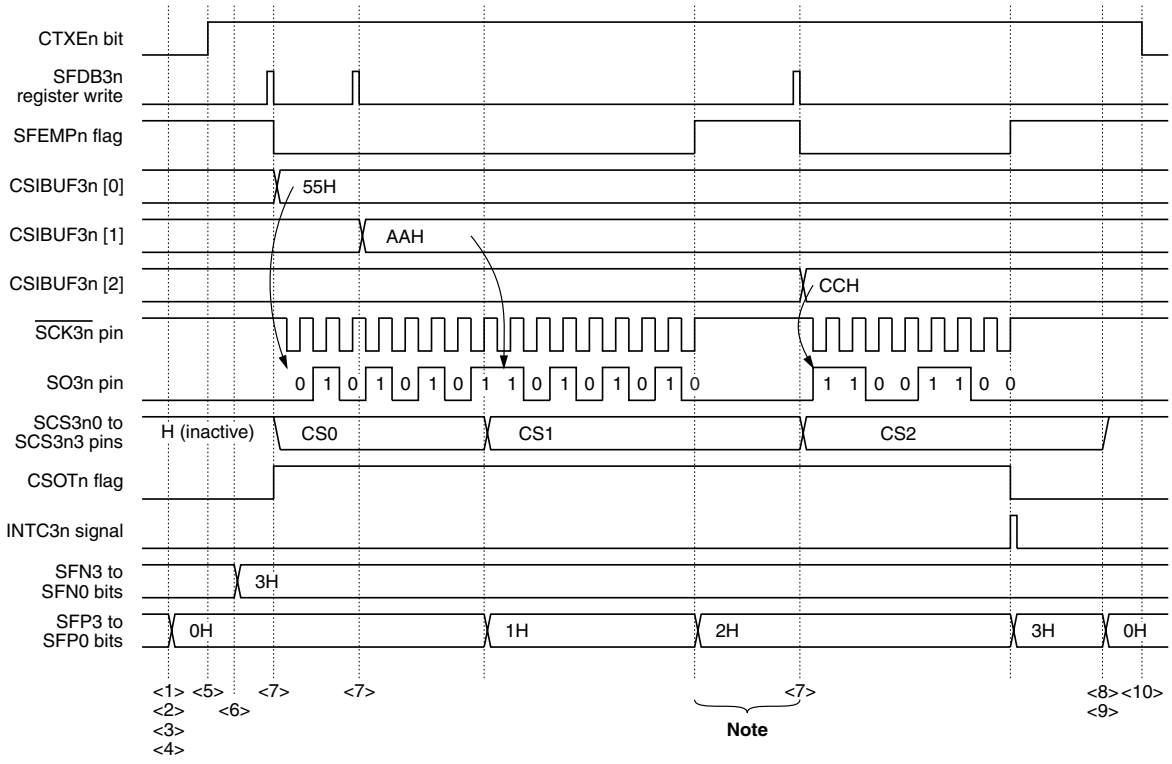
Remarks: 1. To execute a further transfer, repeat <6> and <7> before <8>.

2. n = 0, 1

17.6.7 Consecutive mode (master mode, transmission mode)

Figure 17-28: Consecutive Mode (Master Mode, Transmission Mode)

MSB First (DIR bit = 0), CKP bit = 0, DAP bit = 0
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
INTC3n Interrupt Not Delayed (CSIT bit = 0),
Transfer Wait: Disabled (CSWE bit = 0),
Chip Select Active Level: L-Level (CSLVn3 to CSLVn0 bits = 0000B)



Note: During this period a reception from the slave is put on hold until at least one transmit data has been loaded to the CSIBUFn register by writing the SFDB3n register (SFEMPn flag of SFA3n register = 0) in order to start the transfer.

<1> When the CSICAEn bit of the CSIM3n register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission by setting the CTXEn bit to 1.
<6> Set the number of data to be transmitted by using the SFNn3 to SFNn0 bits of the SFN3n register.
<7> Write first CS data to the SFCS3n register and subsequently write transfer data to the SFDB3n register. Writing data exceeding the set value of the SFN3n register is prohibited.
<8> Confirm that the INTC3n interrupt has occurred and the SFEMPn bit is 1. Then write 1 to the FPCLRn bit of the SFA3n register, and clear all the CSIBUFn pointers to 0 in preparation for the next transfer.
<9> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<10> Disable transmission by clearing the CTXEn bit of the CSIM3n register to 0 (end of transmission).

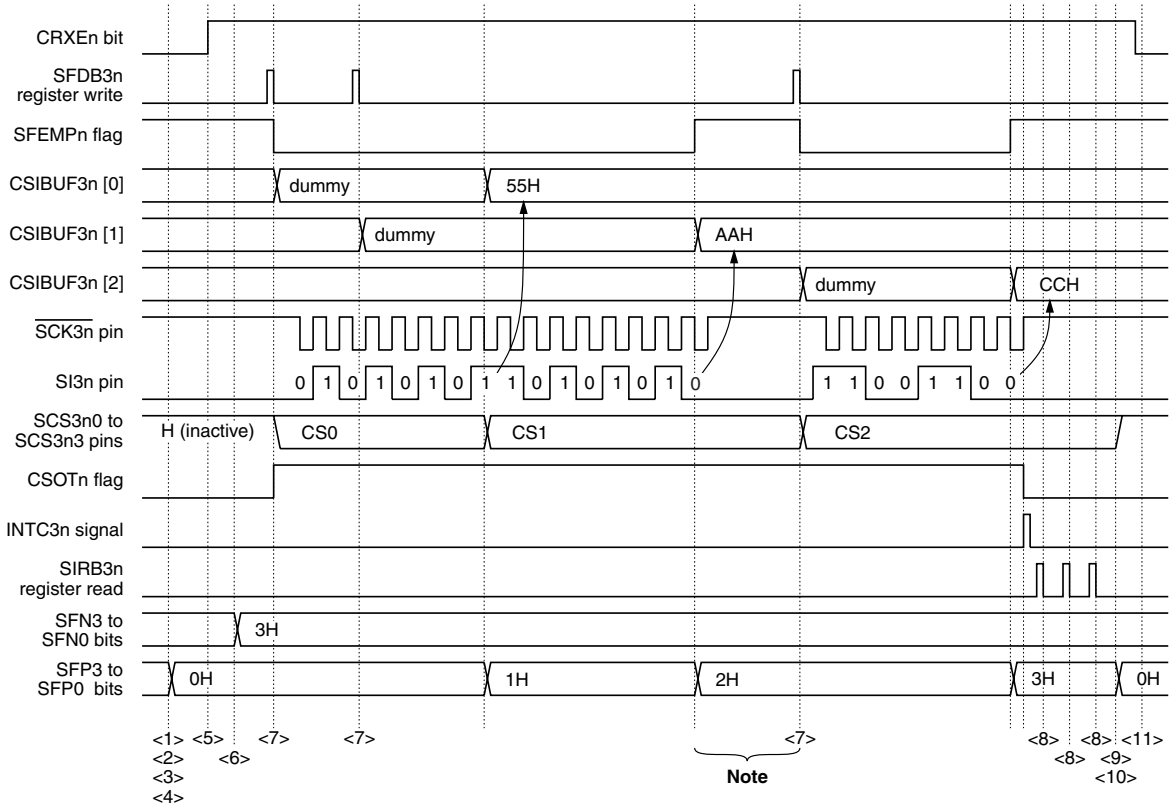
Remarks: 1. To execute a further transfer, repeat <6> to <9> before <10>.

2. n = 0, 1

17.6.8 Consecutive mode (master mode, reception mode)

Figure 17-29: Consecutive Mode (Master Mode, Reception Mode)

MSB First (DIR bit = 0), CKP bit = 0, DAP bit = 1
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
INTC3n Interrupt Not Delayed (CSIT bit = 0),
Transfer Wait: Disabled (CSWE bit = 0),
Chip Select Active Level: L-Level (CSLVn3 to CSLVn0 bits = 0000B)



Note: During this period a reception from the slave is put on hold until at least one dummy transmit data has been loaded to the CSIBUFn register by writing the SFDB3n register (SFEMPn flag of SFA3n register = 0) in order to start the transfer.

Remark: n = 0, 1

<1> When the CSICAEn bit of the CSIM3n register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable reception by setting the CRXEn bit to 1.
<6> Set the number of data to be received by using the SFNn3 to SFNn0 bits of the SFN3n register.
<7> Write first CS data to the SFCS3n register and subsequently write dummy transfer data to the SFDB3n register (reception start trigger). Writing dummy data exceeding the set value of the SFN3n register is prohibited.
<8> Confirm that the INTC3n interrupt has occurred and the SFEMPn bit is 1. Then read the SIRB3n register (sequentially read the receive data stored in the CSIBUFn register).
<9> Write 1 to the FPCLRn bit of the SFA3n register, and clear all the CSIBUFn pointers to 0 in preparation for the next transfer.
<10> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<11> Disable reception by clearing the CRXEn bit of the CSIM3n register to 0 (end of reception).

Remarks: 1. To execute a further transfer, repeat <6> to <10> before <11>.
Perform writing dummy transfer data in <7> every time transfer is executed.

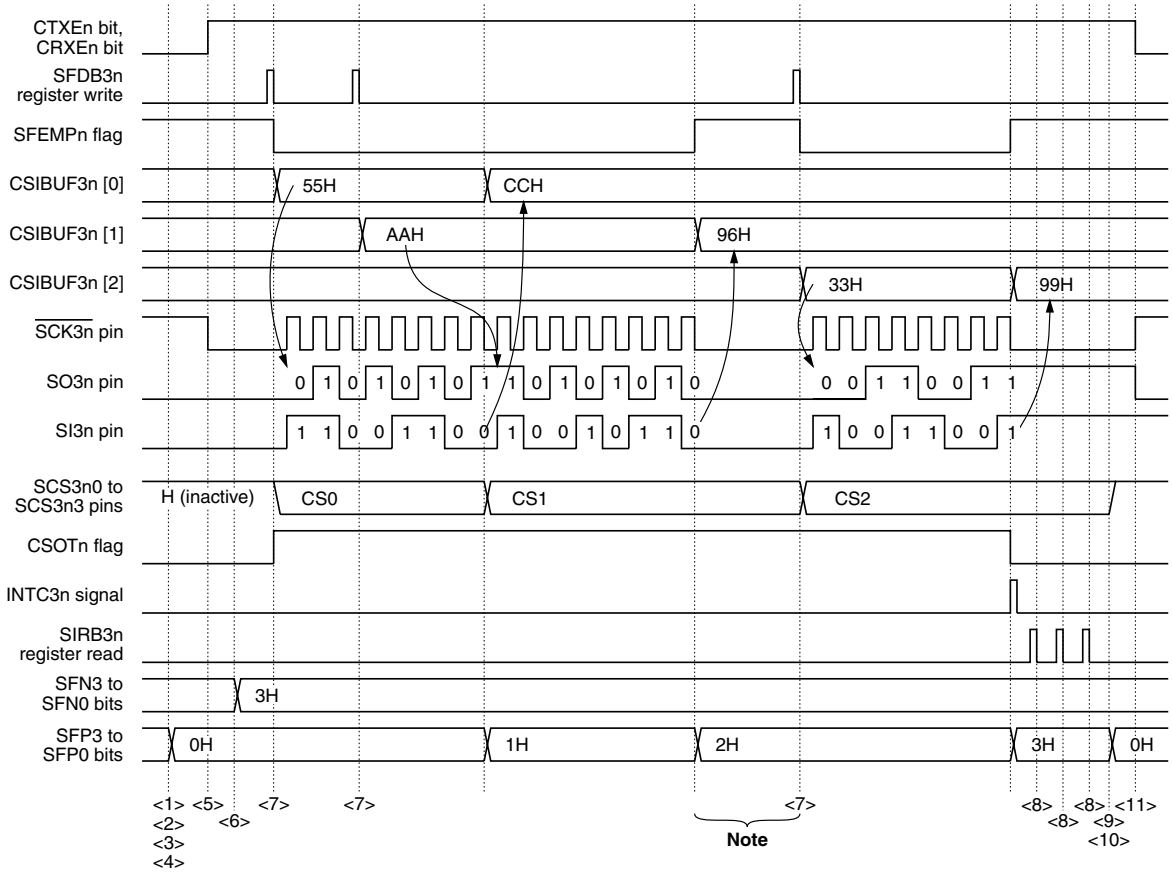
2. The SO3n pin output is fixed to low level (default value).

3. n = 0, 1

17.6.9 Consecutive mode (master mode, transmission/reception mode)

Figure 17-30: Consecutive Mode (Master Mode, Transmission/Reception Mode)

MSB First (DIR bit = 0), CKP bit = 0, DAP bit = 1
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
INTC3n Interrupt Not Delayed (CSIT bit = 0),
Transfer Wait: Disabled (CSWE bit = 0),
Chip Select Active Level: L-Level (CSLVn3 to CSLVn0 bits = 0000B)



Note: During this period a reception from the slave is put on hold until at least one transmit data has been loaded to the CSIBUFn register by writing the SFDB3n register (SFEMPn flag of SFA3n register = 0) in order to start the transfer.

Remark: n = 0, 1

<1> When the CSICAEn bit of the CSIM3n register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission/reception by setting both the CTXEn and CRXEn bits to 1.
<6> Set the number of data to be transmitted/received by using the SFNn3 to SFNn0 bits of the SFN3n register.
<7> Write first CS data to the SFCS3n register and subsequently write transfer data to the SFDB3n register. Writing data exceeding the set value of the SFN3n register is prohibited.
<8> Confirm that the INTC3n interrupt has occurred and the SFEMPn bit is 1. Then read the SIRB3n register (sequentially read the receive data stored in the CSIBUFn register).
<9> Write 1 to the FPCLRn bit of the SFA3n register, and clear all the CSIBUFn pointers to 0 in preparation for the next transfer.
<10> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<11> Disable transmission/reception by clearing the CTXEn and CRXEn bits of the CSIM3n register to 0 (end of transmission/reception).

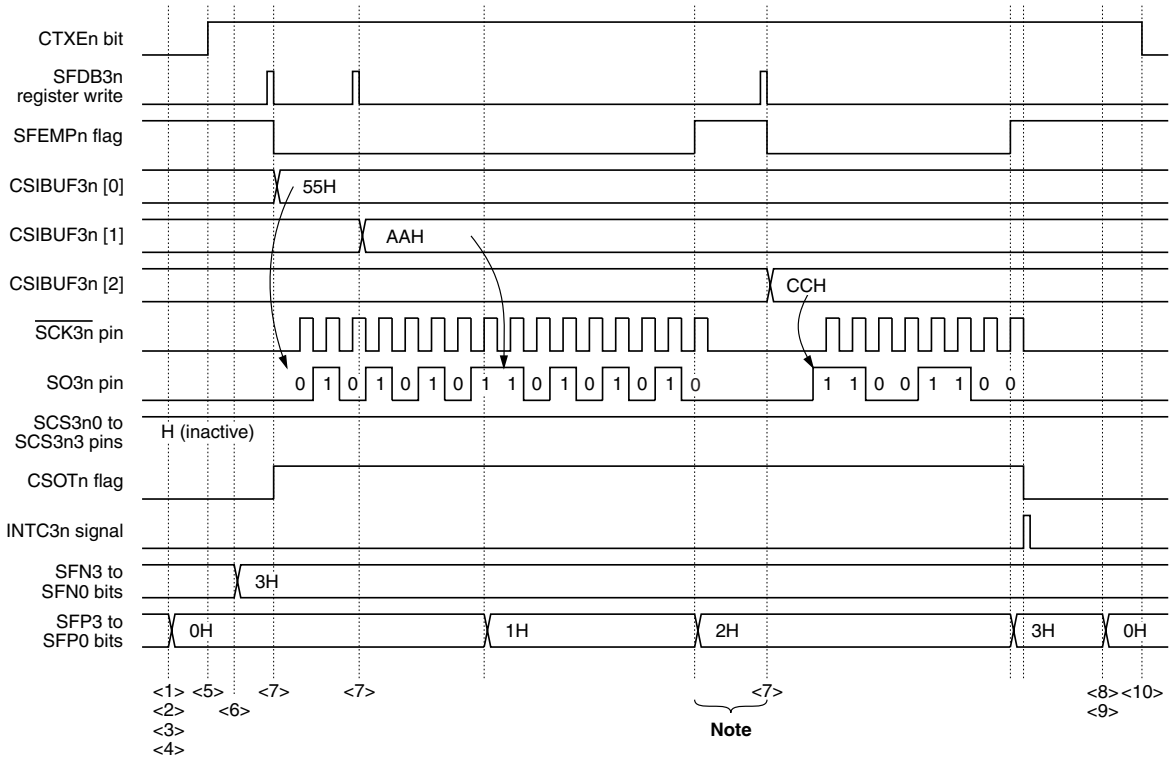
Remarks: 1. To execute a further transfer, repeat <6> to <10> before <11>.

2. n = 0, 1

17.6.10 Consecutive mode (slave mode, transmission mode)

Figure 17-31: Consecutive Mode (Slave Mode, Transmission Mode)

MSB First (DIR bit = 0), CKP bit = 1, DAP bit = 1
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
INTC3n Interrupt Not Delayed (CSIT bit = 0),
Transfer Wait: Disabled (CSWE bit = 0),
Chip Select Active Level: L-Level (CSLVn3 to CSLVn0 bits = 0000B)



Note: During this period a reception request from the master will be ignored until at least one transmit data is loaded to the CSIBUFn register by writing the SFDB3n register (SFEMPn flag of SFA3n register = 0).

Remark: n = 0, 1

<1> When the CSICAEn bit of the CSIM3n register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission by setting the CTXEn bit to 1.
<6> Set the number of data to be transmitted by using the SFNn3 to SFNn0 bits of the SFN3n register.
<7> Write transfer data to the SFDB3n register. Writing data exceeding the set value of the SFN3n register is prohibited.
Since the chip select outputs (SCS3n0 to SCS3n3) are ineffective in the slave mode and always output the inactive level, writing of CS data to the SFCS3n register is not necessary.
<8> Confirm that the INTC3n interrupt has occurred and the SFEMPn bit is 1. Then write 1 to the FPCLRn bit of the SFA3n register, and clear all the CSIBUFn pointers to 0 in preparation for the next transfer.
<9> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<10> Disable transmission by clearing the CTXEn bit of the CSIM3n register to 0 (end of transmission).

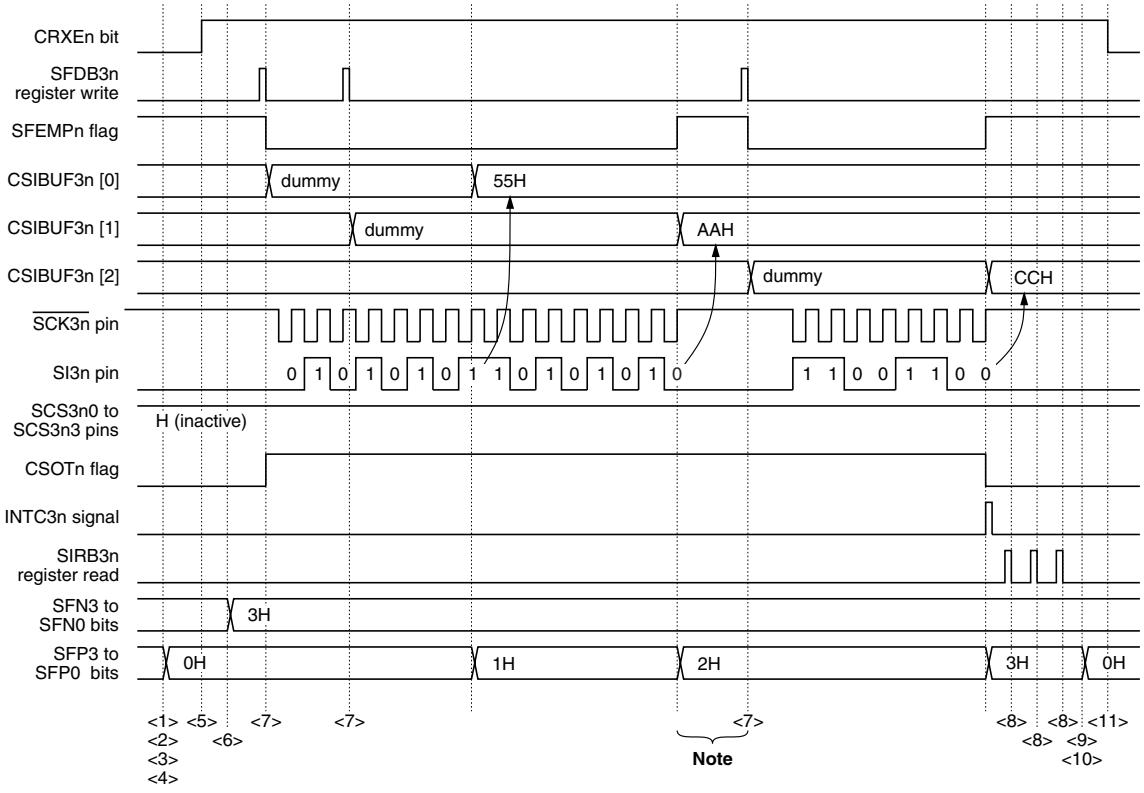
Remarks: 1. To execute a further transfer, repeat <6> to <9> before <10>.

2. n = 0, 1

17.6.11 Consecutive mode (slave mode, reception mode)

Figure 17-32: Consecutive Mode (Slave Mode, Reception Mode)

MSB First (DIR bit = 0), CKP bit = 0, DAP bit = 0
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
INTC3n Interrupt Not Delayed (CSIT bit = 0),
Transfer Wait: Disabled (CSWE bit = 0),
Chip Select Active Level: L-Level (CSLVn3 to CSLVn0 bits = 0000B)



Note: During this period a transmission from the master will be ignored until at least one transmit data is loaded to the CSIBUFn register by writing the SFDB3n register (SFEMPn flag of SFA3n register = 0).

Remark: n = 0, 1

<1> When the CSICAEn bit of the CSIM3n register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable reception by setting the CRXEn bit to 1.
<6> Set the number of data to be received by using the SFNn3 to SFNn0 bits of the SFN3n register.
<7> Write dummy transfer data to the SFDB3n register (reception start trigger). Writing dummy data exceeding the set value of the SFN3n register is prohibited.
Since the chip select outputs (SCS3n0 to SCS3n3) are ineffective in the slave mode and always output the inactive level, writing of CS data to the SFCS3n register is not necessary.
<8> Confirm that the INTC3n interrupt has occurred and the SFEMPn bit is 1. Then read the SIRB3n register (sequentially read the receive data stored in the CSIBUFn register).
<9> Write 1 to the FPCLRn bit of the SFA3n register, and clear all the CSIBUFn pointers to 0 in preparation for the next transfer.
<10> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<11> Disable reception by clearing the CRXEn bit of the CSIM3n register to 0 (end of reception).

Remarks: 1. To execute a further transfer, repeat <6> to <10> before <11>.
Perform writing dummy transfer data in <7> every time transfer is executed.

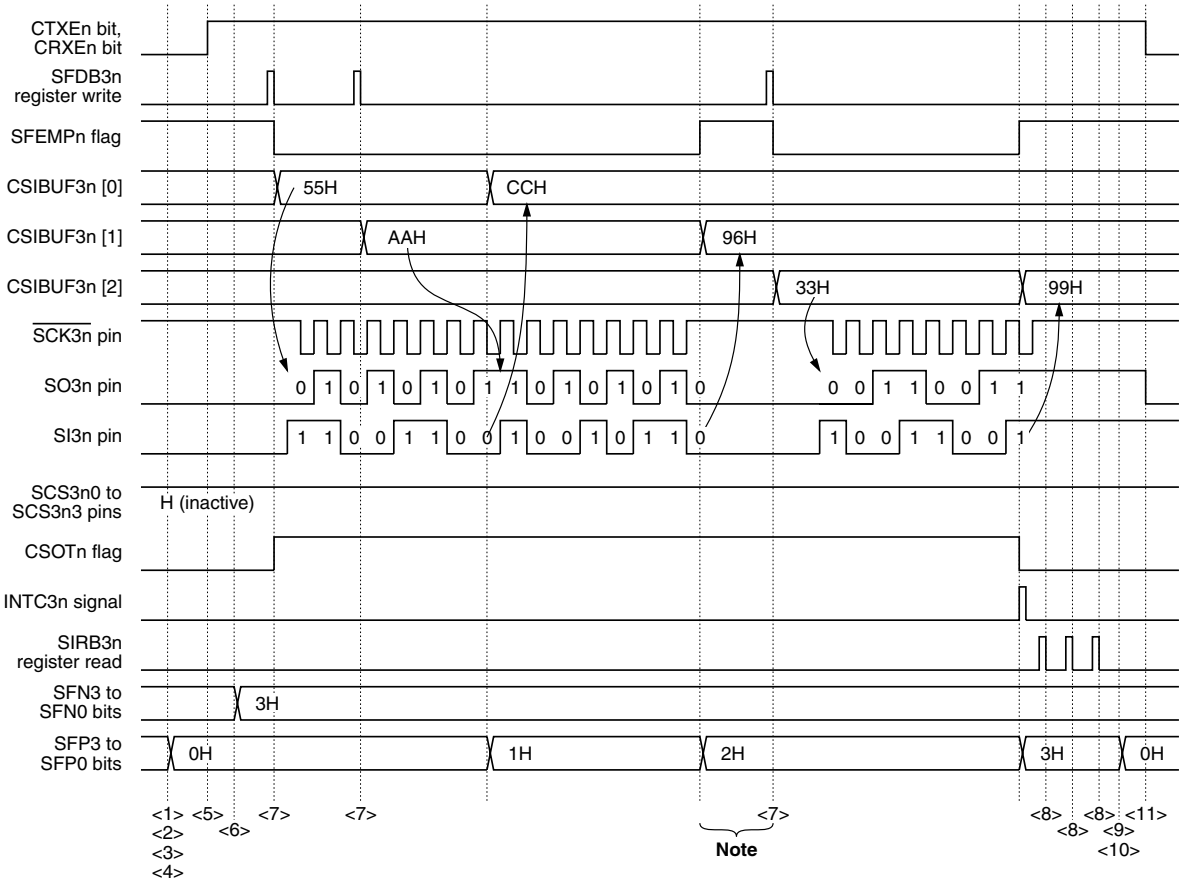
2. The SO3n pin output is fixed to low level (default value).

3. n = 0, 1

17.6.12 Consecutive mode (in slave mode and transmission/reception mode)

Figure 17-33: Consecutive Mode (Slave Mode, Transmission/Reception Mode)

MSB First (DIR bit = 0), CKP bit = 0, DAP bit = 1
Transfer Data Length: 8 Bits (CCLn3 to CCLn0 bits = 1000B)
INTC3n Interrupt Not Delayed (CSIT bit = 0),
Transfer Wait: Disabled (CSWE bit = 0),
Chip Select Active Level: L-Level (CSLVn3 to CSLVn0 bits = 0000B)



Note: During this period a transmission/reception from the master will be ignored until at least one transmit data is loaded to the CSIBUFn register by writing the SFDB3n register (SFEMPn flag of SFA3n register = 0).

Remark: n = 0, 1

<1> When the CSICAEn bit of the CSIM3n register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission/reception by setting both the CTXEn and CRXEn bits to 1.
<6> Set the number of data to be transmitted/received by using the SFNn3 to SFNn0 bits of the SFN3n register.
<7> Write transfer data to the SFDB3n register. Writing data exceeding the set value of the SFN3n register is prohibited.
Since the chip select outputs (SCS3n0 to SCS3n3) are ineffective in the slave mode and always output the inactive level, writing of CS data to the SFCS3n register is not necessary.
<8> Confirm that the INTC3n interrupt has occurred and the SFEMPn bit is 1. Then read the SIRB3n register (sequentially read the receive data stored in the CSIBUFn register).
<9> Write 1 to the FPCLRn bit of the SFA3n register, and clear all the CSIBUFn pointers to 0 in preparation for the next transfer.
<10> Confirm that the SFFULn bit = 0, SFEMPn bit = 1, and SFPn3 to SFPn0 bits = 0000 in the SFA3n register.
<11> Disable transmission/reception by clearing the CTXEn and CRXEn bits of the CSIM3n register to 0 (end of transmission/reception).

Remarks: 1. To execute a further transfer, repeat <6> to <10> before <11>.

2. n = 0, 1

17.7 Cautions

The following points must be observed when using CSI3n (n = 0, 1).

Cautions: 1. The CSI3n unit is reset and CSI3n is stopped when the CSICAEn bit of the CSIM3n register is cleared to 0. To operate CSI3n, first set the CSICAEn bit to 1. Usually, before clearing the CSICAEn bit to 0, clear both the CTXEn and CRXEn bits to 0 (after the end of transfer).

2. Be sure to write 1 to the FPCLRn bit of the SFA3n register to clear all the CSIBUFn pointers to 0 before enabling transfer by setting the CTXEn or CRXEn bit of the CSIM3n register to 1. If the CTXEn or CRXEn bit is set to 1 without clearing the pointers, and if the previously transferred data remains in the CSIBUFn register, transferring that data is immediately started.
If transfer data is set to the CSIBUFn register before transfer is enabled, transfer is started as soon as the CTXEn or CRXEn bit is set to 1.

3. If the SFA3n register is read immediately after data has been written to the SFDB3n and SFDB3nL registers, the SFFULn, SFEMPn, and SFPn3 to SFPn0 bits of the SFA3n register may not change their values in time.
If the SFA3n register is read before the SFFULn bit is set to 1 and a 17th data is written, the CSIBUFn overflow interrupt (INTC3nOVF) occurs.

4. When using CSI3n in configuration with DMA transfer, observe that only single mode is permitted (TRMDn bit of CSIM3n register = 0), and chip select CSI registers (SFCS3n, SFCS3nL) are not supported.

Chapter 18 AFCAN Controller

18.1 Outline Description

This product features an on-chip 2-channel CAN (Controller Area Network) controller that complies with CAN protocol as standardized in ISO 11898.

18.1.1 Features

- Compliant with ISO 11898 and tested according to ISO/DIS 16845 (CAN conformance test)
- Standard frame and extended frame transmission/reception enabled
- Transfer rate: 1 Mbps max.
- 32 message buffers/2 channels
- Receive/transmit history list function
- Automatic block transmission function
- Multi-buffer receive block function
- Mask setting of four patterns is possible for each channel

18.1.2 Implementation

Implemented in V850E/PH2 are 2 AFCAN modules. The actual mapping of the SFR depends on the content of the BPC register (please refer to **3.4.7 ”Programmable peripheral I/O area” on page 109**). For a complete list refer to **Table 3-6, “Programmable Peripheral I/O Registers,” on page 111**.

Each module has it’s own channel offset to this base address. These offsets are as follows:

Table 18-1: CAN Channel Offsets

CAN Channel	Channel Offset	Type
CAN0	0x0000	AFCAN
CAN1	0x0600	

18.1.3 Overview of functions

Table 18-2 presents an overview of the CAN controller functions.

Table 18-2: Overview of Functions

Function	Details
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Maximum 1 Mbps
Data storage	Storing messages in the CAN RAM
Number of messages	- 32 message buffers/2 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 number 0 to 7 specified as the transmit message buffer 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 message reception when a 16-bit timer is used in combination. Time stamp capture trigger can be selected (SOF or EOF in a CAN message frame can be detected.). - The time stamp function can be set for a transmit message. - A specific byte in a data field can be replaced by a capture time stamp[Note]
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
Forced release from bus-off state	- Default mode can be set while bus is off, so that bus can be forcibly released from the bus-off state.
Power save mode	- CAN sleep mode (can be woken up by CAN bus) - CAN stop mode (cannot be woken up by CAN bus)

Note: This function is valid only with a macro having an advanced time stamp function.

18.1.4 Configuration

The CAN controller is composed of the following four blocks.

(1) NPB interface

This functional block provides an NPB (NEC peripheral I/O bus) interface and means of transmitting and receiving signals between the CAN module and the host CPU.

(2) MCM (Message Control Module)

This functional block controls access to the CAN protocol layer and to the CAN RAM within the CAN module.

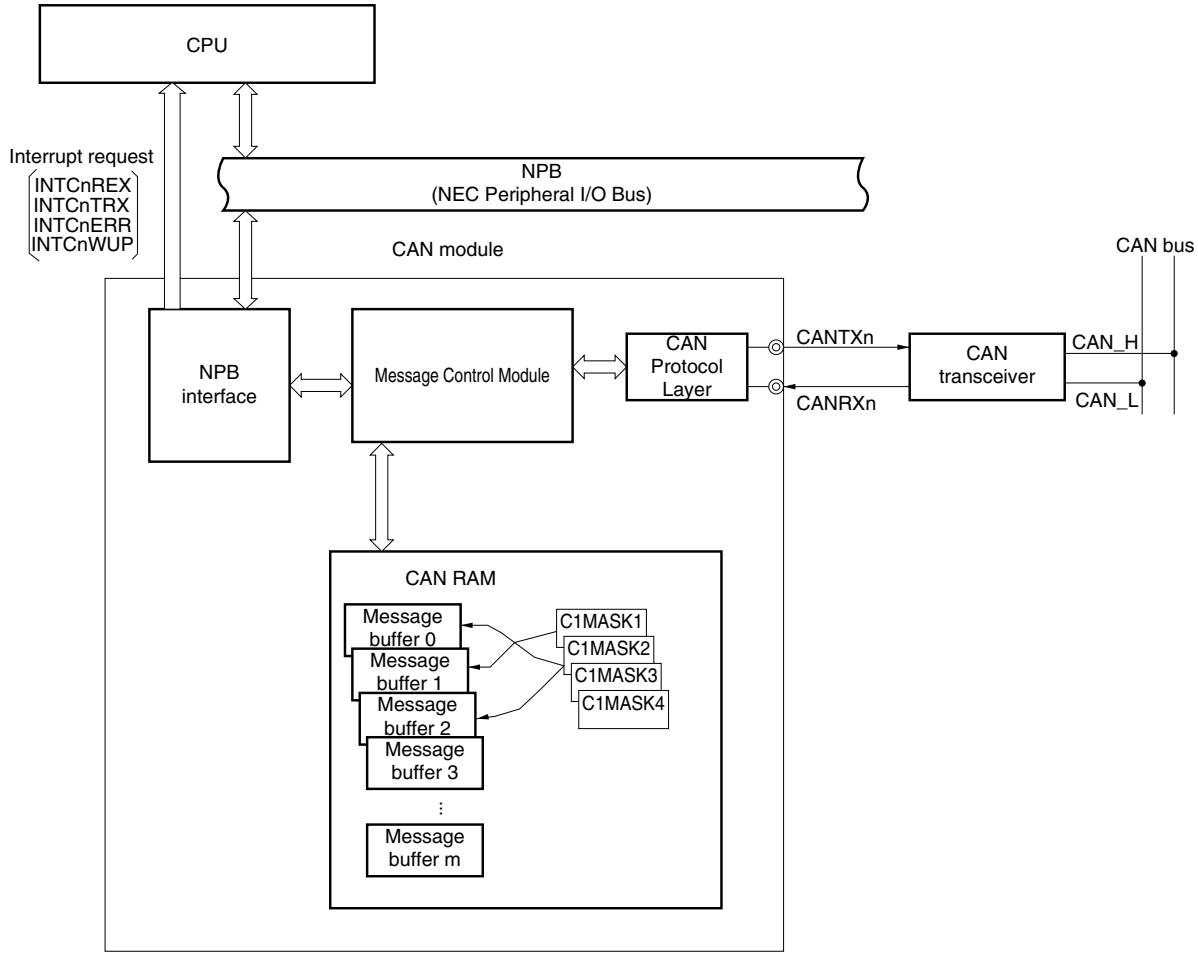
(3) CAN protocol layer

This functional block is involved in the operation of the CAN protocol and its related settings.

(4) CAN RAM

This is the CAN memory functional block, which is used to store message IDs, message data, etc.

Figure 18-1: Block Diagram of CAN Module

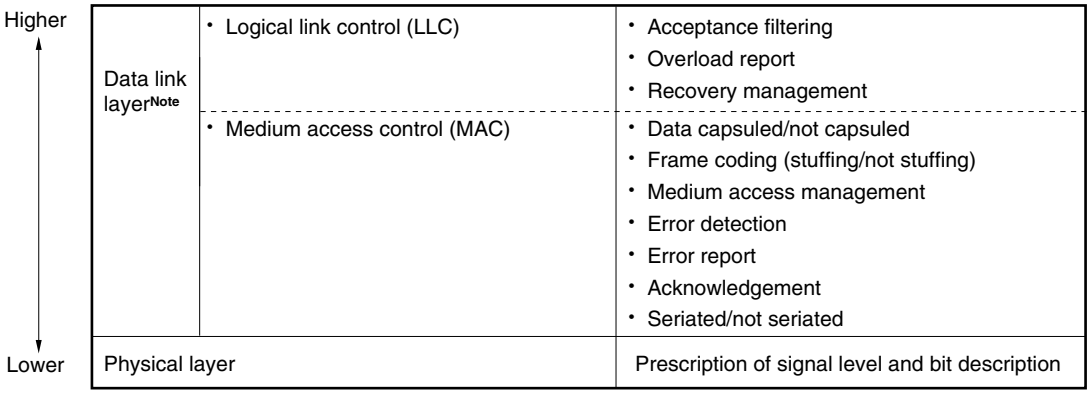


Remark: n = 0, 1,
m = 0 to 31

18.2 CAN Protocol

CAN (Controller Area Network) is a high-speed multiplex communication protocol for real-time communication in automotive applications (class C). CAN is prescribed by ISO 11898. For details, refer to the ISO 11898 specifications.
The CAN specification is generally divided into two layers: a physical layer and a data link layer. In turn, the data link layer includes logical link and medium access control. The composition of these layers is illustrated below.

Figure 18-2: Composition of Layers



Note: CAN controller specification

18.2.1 Frame format

(1) Standard format frame

- The standard format frame uses 11-bit identifiers, which means that it can handle up to 2048 messages.

(2) Extended format frame

- The extended format frame uses 29-bit (11 bits + 18 bits) identifiers which increase the number of messages that can be handled to 2048×2^{18} messages.

- Extended format frame is set when “recessive level” (CMOS level equals “1”) is set for both the SRR and IDE bits in the arbitration field.

18.2.2 Frame types

The following four types of frames are used in the CAN protocol.

Table 18-3: Frame Types

Frame Type	Description
Data frame	Frame used to transmit data
Remote frame	Frame used to request a data frame
Error frame	Frame used to report error detection
Overload frame	Frame used to delay the next data frame or remote frame

(1) Bus value

The bus values are divided into dominant and recessive.

- Dominant level is indicated by logical 0.
- Recessive level is indicated by logical 1.

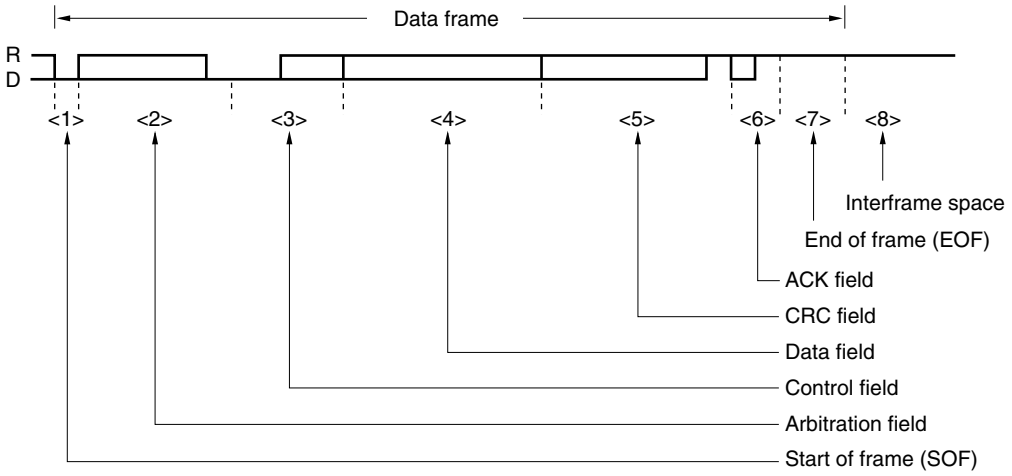
When a dominant level and a recessive level are transmitted simultaneously, the bus value becomes dominant level.

18.2.3 Data frame and remote frame

(1) Data frame

A data frame is composed of seven fields.

Figure 18-3: Data Frame

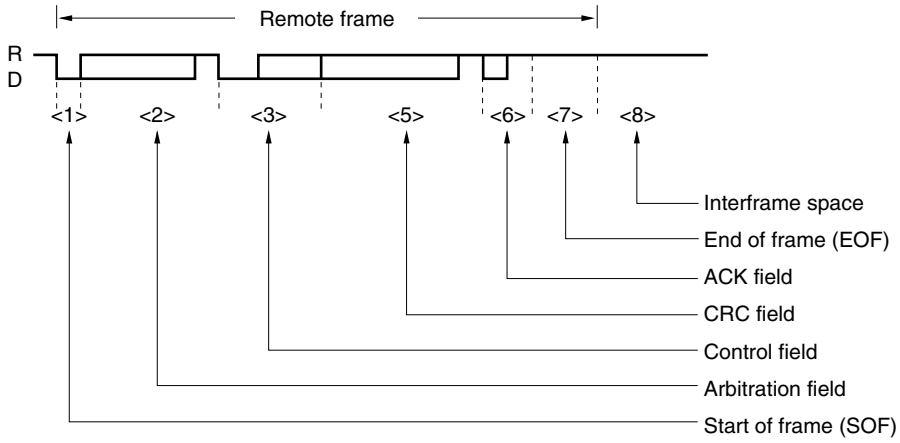


Remark: D: Dominant = 0
R: Recessive = 1

(2) Remote frame

A remote frame is composed of six fields.

Figure 18-4: Remote Frame



Remarks: 1. The data field is not transferred even if the control field's data length code is not "0000B".

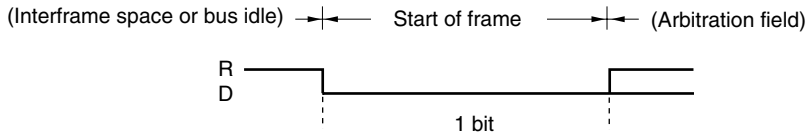
2. D: Dominant = 0
R: Recessive = 1

(3) Description of fields

<1> Start of frame (SOF)

The start of frame field is located at the start of a data frame or remote frame.

Figure 18-5: Start of Frame (SOF)



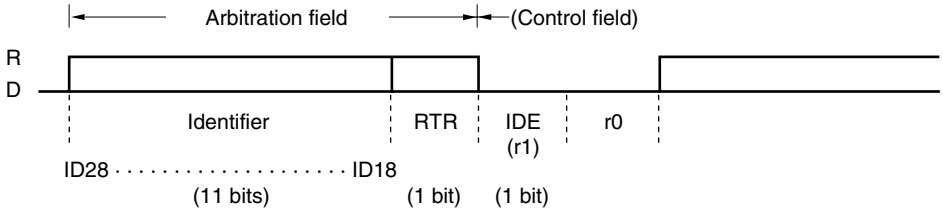
Remark: D: Dominant = 0
R: Recessive = 1

- If dominant level is detected in the bus idle state, a hard-synchronization is performed (the current TQ is assigned to be the SYNC segment).
- If dominant level is sampled at the sample point following such a hard-synchronization, the bit is assigned to be a SOF. If recessive level is detected, the protocol layer returns to the bus idle state and regards the preceding dominant pulse as a disturbance only. No error frame is generated in such case.

<2> Arbitration field

The arbitration field is used to set the priority, data frame/remote frame, and frame format.

Figure 18-6: Arbitration Field (in Standard Format Mode)

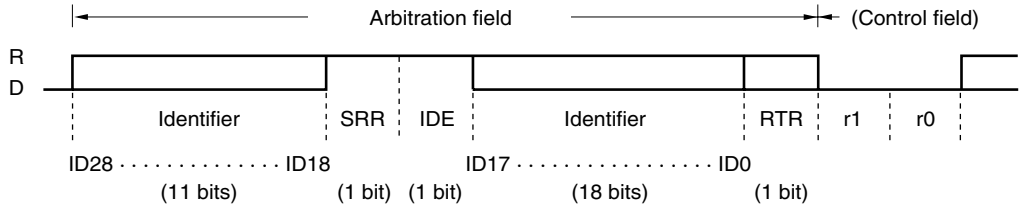


Cautions: 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Remark: D: Dominant = 0
R: Recessive = 1

Figure 18-7: Arbitration Field (in Extended Format Mode)



Cautions: 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Remark: D: Dominant = 0
R: Recessive = 1

Table 18-4: RTR Frame Settings

Frame Type	RTR Bit
Data frame	0 (D)
Remote frame	1 (R)

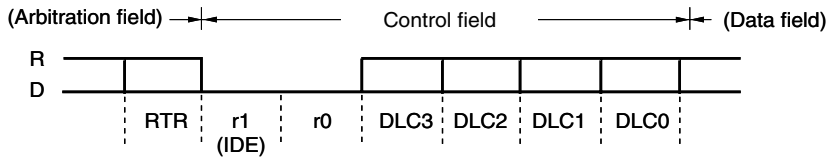
Table 18-5: Frame Format Setting (IDE Bit) and Number of Identifier (ID) Bits

Frame Format	SRR Bit	IDE Bit	Number. of Bits
Standard format mode	None	0 (D)	11 bits
Extended format mode	1 (R)	1 (R)	29 bits

<3> Control field

The control field sets “N” as the number of data bytes in the data field (N = 0 to 8).

Figure 18-8: Control Field



Remark: D: Dominant = 0
R: Recessive = 1

In a standard format frame, the control field’s IDE bit is the same as the r1 bit.

Table 18-6: Data Length Setting

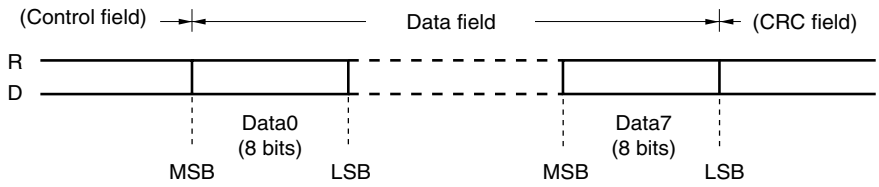
Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than above				8 bytes regardless of the value of DLC3 to DLC0

Caution: In the remote frame, there is no data field even if the data length code is not 0000B.

<4> Data field

The data field contains the amount of data (byte units) set by the control field. Up to 8 units of data can be set.

Figure 18-9: Data Field

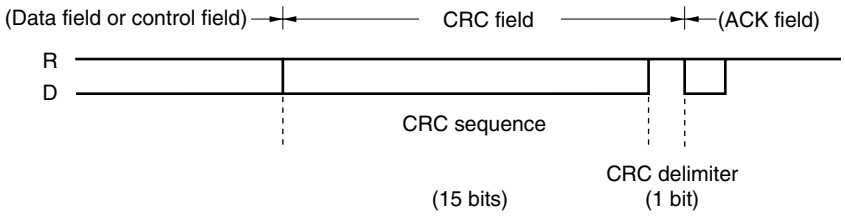


Remark: D: Dominant = 0
R: Recessive = 1

<5> CRC field

The CRC field is a 16-bit field that is used to check for errors in transmit data.

Figure 18-10: CRC Field



Remark: D: Dominant = 0
R: Recessive = 1

- The polynomial P(X) used to generate the 15-bit CRC sequence is expressed as follows.

  $P(X) = X15 + X14 + X10 + X8 + X7 + X4 + X3 + 1$

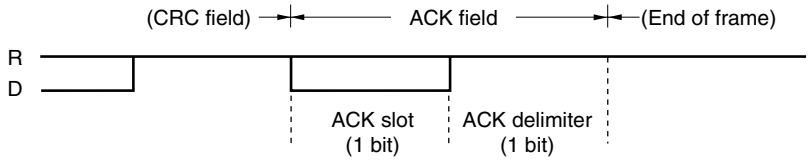
- Transmitting node: Transmits the CRC sequence calculated from the data (before bit stuffing) in the start of frame, arbitration field, control field, and data field.

- Receiving node: Compares the CRC sequence calculated using data bits that exclude the stuffing bits in the receive data with the CRC sequence in the CRC field. If the two CRC sequences do not match, the node issues an error frame.

<6> ACK field

The ACK field is used to acknowledge normal reception.

Figure 18-11: ACK Field



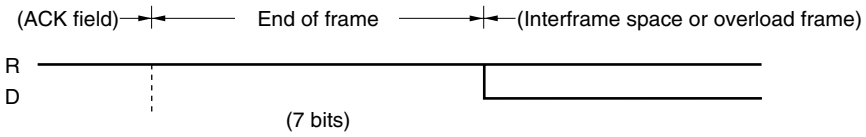
Remark: D: Dominant = 0
R: Recessive = 1

- If no CRC error is detected, the receiving node sets the ACK slot to the dominant level.
- The transmitting node outputs two recessive-level bits.

<7> End of frame (EOF)

The end of frame field indicates the end of data frame/remote frame.

Figure 18-12: End of Frame (EOF)



Remark: D: Dominant = 0
R: Recessive = 1

<8> Interframe space

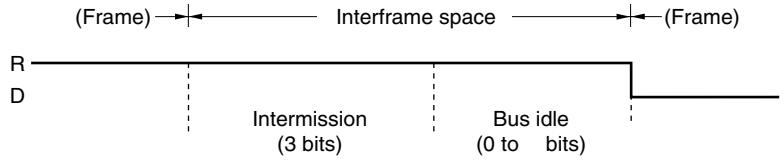
The interframe space is inserted after a data frame, remote frame, error frame, or overload frame to separate one frame from the next.

- The bus state differs depending on the error status.

(a) Error active node

The interframe space consists of a 3-bit intermission field and a bus idle field.

Figure 18-13: Interframe Space (Error Active Node)



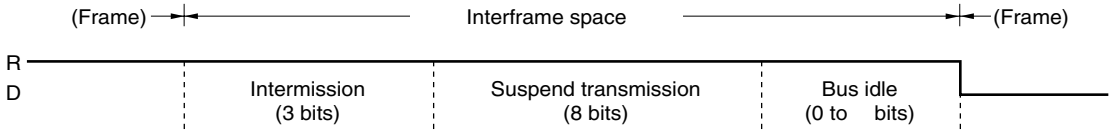
Remarks: 1. Bus idle: State in which the bus is not used by any node.

2. D: Dominant = 0
R: Recessive = 1

(b) Error passive node

The interframe space consists of an intermission field, a suspend transmission field, and a bus idle field.

Figure 18-14: Interframe Space (Error Passive Node)



Remarks: 1. Bus idle: State in which the bus is not used by any node.
Suspend transmission: Sequence of 8 recessive-level bits transmitted from the node in the error passive status.

2. D: Dominant = 0
R: Recessive = 1

Usually, the intermission field is 3 bits. If the transmitting node detects a dominant level at the third bit of the intermission field, however, it executes transmission.

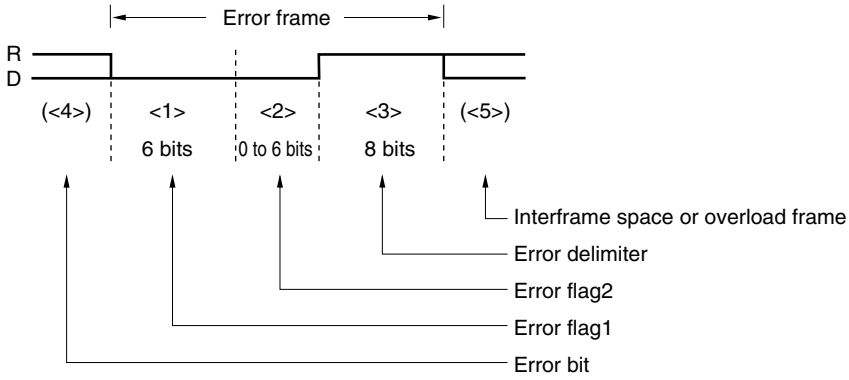
Table 18-7: Operation in Error Status

Error Status	Operation
Error active	A node in this status can transmit immediately after a 3-bit intermission.
Error passive	A node in this status can transmit 8 bits after the intermission.

18.2.4 Error frame

An error frame is output by a node that has detected an error.

Figure 18-15: Error Frame



Remark: D: Dominant = 0
R: Recessive = 1

Table 18-8: Definition Error Frame Fields

No.	Name	Bit Count	Definition
<1>	Error flag1	6	Error active node: Outputs 6 dominant-level bits consecutively. Error passive node: Outputs 6 recessive-level bits consecutively. If another node outputs a dominant level while one node is outputting a passive error flag, the passive error flag is not cleared until the same level is detected 6 bits in a row.
<2>	Error flag2	0 to 6	Nodes receiving error flag 1 detect bit stuff errors and issues this error flag.
<3>	Error delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Error bit	–	The bit at which the error was detected. The error flag is output from the bit next to the error bit. In the case of a CRC error, this bit is output following the ACK delimiter.
<5>	Interframe space/overload frame	–	An interframe space or overload frame starts from here.

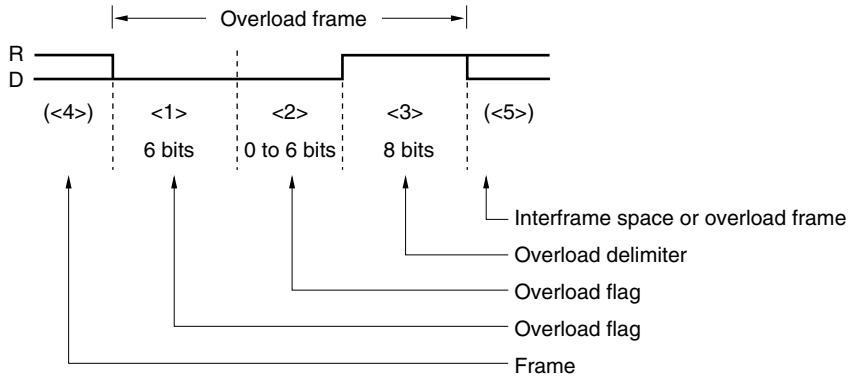
18.2.5 Overload frame

An overload frame is transmitted under the following conditions.

- When the receiving node has not completed the reception operation[Note]
- If a dominant level is detected at the first two bits during intermission
- If a dominant level is detected at the last bit (7th bit) of the end of frame or at the last bit (8th bit) of the error
- delimiter/overload delimiter

Note: The CAN is internally fast enough to process all received frames not generating overload frames.

Figure 18-16: Overload Frame



Remark: D: Dominant = 0
R: Recessive = 1

Table 18-9: Definition of Overload Frame Fields

No	Name	Bit Count	Definition
<1>	Overload flag	6	Outputs 6 dominant-level bits consecutively.
<2>	Overload flag from other node	0 to 6	The node that received an overload flag in the interframe space outputs an overload flag.
<3>	Overload delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Frame	–	Output following an end of frame, error delimiter, or overload delimiter.
<5>	Interframe space/ overload frame	–	An interframe space or overload frame starts from here.

18.3 Functions

18.3.1 Determining bus priority

(1) When a node starts transmission:

- During bus idle, the node that output data first transmits the data.

(2) When more than one node starts transmission:

- The node that outputs the dominant level for the longest consecutively from the first bit of the arbitration field
acquires the bus priority (if a dominant level and a recessive level are simultaneously transmitted, the dominant level is taken as the bus value).

- The transmitting node compares its output arbitration field and the data level on the bus.

Table 18-10: Determining Bus Priority

Level match	Continuous transmission
Level mismatch	Continuous transmission

(3) Priority of data frame and remote frame

- When a data frame and a remote frame are on the bus, the data frame has priority because its RTR bit, the last bit in the arbitration field, carries a dominant level.

Remark: If the extended-format data frame and the standard-format remote frame conflict on the bus (if ID28 to ID18 of both of them are the same), the standard-format remote frames takes priority.

18.3.2 Bit stuffing

Bit stuffing is used to establish synchronization by appending 1-bit inverted data if the same level continues for 5 bits, in order to prevent a burst error.

Table 18-11: Bit Stuffing

Transmission	During the transmission of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, 1 inverted-level bit of data is inserted before the following bit.
Reception	During the reception of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, reception is continued after deleting the next bit.

18.3.3 Multi masters

As the bus priority (a node acquiring transmit functions) is determined by the identifier, any node can be the bus master.

18.3.4 Multi cast

Although there is one transmitting node, two or more nodes can receive the same data at the same time because the same identifier can be set to two or more nodes.

18.3.5 CAN sleep mode/CAN stop mode function

The CAN sleep mode/CAN stop mode function puts the CAN controller in waiting mode to achieve low power consumption.
The controller is woken up from the CAN sleep mode by bus operation but it is not woken up from the CAN stop mode by bus operation (the CAN stop mode is controlled by CPU access).

18.3.6 Error control function

(1) Error types

Table 18-12: Error Types

Type	Description of Error		Detection State	
	Detection Method	Detection Condition	Transmission/ Reception	Field/Frame
Bit error	Comparison of output level and level on the bus (except stuff bit)	Mismatch of levels	Transmitting/ receiving node	Bit that outputting data on the bus at the start of frame to end of frame, error frame and overload frame.
Stuff error	Check the receive data at the stuff bit	6 consecutive bits of the same output level	Receiving node	Start of frame to CRC sequence
CRC error	Comparison of the CRC sequence generated from the receive data and the received CRC sequence	Mismatch of CRC	Receiving node	CRC field
Form error	Field/frame check of the fixed format	Detection of fixed format violation	Receiving node	- CRC delimiter - ACK field - End of frame - Error frame - Overload frame
ACK error	Check of the ACK slot by the transmitting node	Detection of recessive level in ACK slot	Transmitting node	ACK slot

(2) Output timing of error frame

Table 18-13: Output Timing of Error Frame

Type	Output Timing
Bit error, stuff error, form error, ACK error	Error frame output is started at the timing of the bit following the detected error.
CRC error	Error frame output is started at the timing of the bit following the ACK delimiter.

(3) Processing in case of error

The transmission node re-transmits the data frame or remote frame after the error frame (however, it does not re-transmit the frame in the single-shot mode.).

(4) Error state

(a) Types of error states

The following three types of error states are defined by the CAN specification.
- Error active
- Error passive
- Bus-off

These types of error states are classified by the values of the TEC7 to TEC0 bits (transmission error counter bits) and the REC6 to REC0 bits (reception error counter bits) of the CAN error counter register as shown in Table 18-14, “Types of Error States,” on page 741.
The present error state is indicated by the CAN module information register (CnINFO).
When each error counter value becomes equal to or greater than the error warning level (96), the TECS0 or RECS0 bit of the CnINFO register is set to 1. In this case, the bus state must be tested because it is considered that the bus has a serious fault. An error counter value of 128 or more indicates an error passive state and the TECS1 or RECS1 bit of the CnINFO register is set to 1.

- If the value of the transmission error counter is greater than or equal to 256 (actually, the transmission error counter does not indicate a value greater than or equal to 256), the bus-off state is reached and the BOFF bit of the CnINFO register is set to 1.

- If only one node is active on the bus at startup (i.e., a particular case such as when the bus is connected only to the local station), ACK is not returned even if data is transmitted. Consequently, re-transmission of the error frame and data is repeated. In the error passive state, however, the transmission error counter is not incremented and the bus-off state is not reached.

Table 18-14: Types of Error States

Type	Operation	Value of Error Counter	Indication of CnINFO Register	Operation specific to Given Error State
Error active	Transmission	0-95	TECS1, 0 = 00	• Outputs an active error flag (6 consecutive dominant-level bits) on detection of the error.
	Reception	0-95	RECS1, 0 = 00	
	Transmission	96-127	TECS1, 0 = 01	
	Reception	96-127	RECS1,0 = 01	
Error passive	Transmission	128-255	TECS1,0 = 11	• Outputs a passive error flag (6 consecutive recessive-level bits) on detection of the error. • Transmits 8 recessive-level bits, in between transmissions, following an intermission (suspend transmission)
	Reception	128 or more	RECS1,0 = 11	
Bus-off	Transmission	256 or more (not indicated) **Note**	BOFF = 1, TECS = 11	• Communication is not possible. Messages are not stored when receiving frames, however, the following operations of <1>, <2>, and <3> are done. <1> TSOUT toggles. <2> REC is incremented/decremented. <3> VALID bit is set. • If the initialization mode is set and then 11 recessive-level bits are generated 128 times in a row in an operation mode other than the initialization mode, the error counter is reset to 0 and the error active state can be restored.

Note: The value of the transmission error counter (TEC) is invalid when the BOFF bit is set to 1. If an error that increments the value of the transmission error counter by +8 while the counter value is in a range of 248 to 255, the counter is not incremented and the bus-off state is assumed.

(b) Error counter

The error counter counts up when an error has occurred, and counts down upon successful transmission and reception. The error counter is updated during the first bit of the error delimiter.

Table 18-15: Error Counter

State	Transmission Error Counter (TEC7 to TEC0)	Reception Error Counter (REC6 to REC0)
Receiving node detects an error (except bit error in the active error flag or overload flag).	No change	+1 (when REPS = 0)
Receiving node detects dominant level following error flag of error frame.	No change	+8 (when REPS = 0)
Transmitting node transmits an error flag. [As exceptions, the error counter does not change in the following cases.] <1> ACK error is detected in error passive state and dominant level is not detected while the passive error flag is being output. <2> A stuff error is detected in an arbitration field that transmitted a recessive level as a stuff bit, but a dominant level is detected.	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active transmitting node)	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active receiving node)	No change	+8(when REPS = 0)
When the node detects 14 consecutive dominant-level bits from the beginning of the active error flag or overload flag, and then subsequently detects 8 consecutive dominant-level bits. When the node detects 8 consecutive dominant levels after a passive error flag	+8 (during transmission)	+8 (during reception, when REPS = 0)
When the transmitting node has completed transmission without error (±0 if error counter = 0)	–1	No change
When the receiving node has completed reception without error	No change	–1 (1 ≤REC6 to REC0 ≤ 127, when REPS = 0) ±0 (REC6 to REC0 = 0, when REPS = 0) Value of 119 to 127 is set (when REPS = 1)

(c) Occurrence of bit error in intermission

An overload frame is generated.

Caution: If an error occurs, it is controlled according to the contents of the transmission error counter and reception error counter before the error occurred. The value of the error counter is incremented after the error flag has been output.

(5) Recovery from bus-off state

When the CAN module is in the bus-off state, the transmission pins (CTXDn) cut off from the CAN bus always output the recessive level.
The CAN module recovers from the bus-off state in the following bus-off recovery sequence.

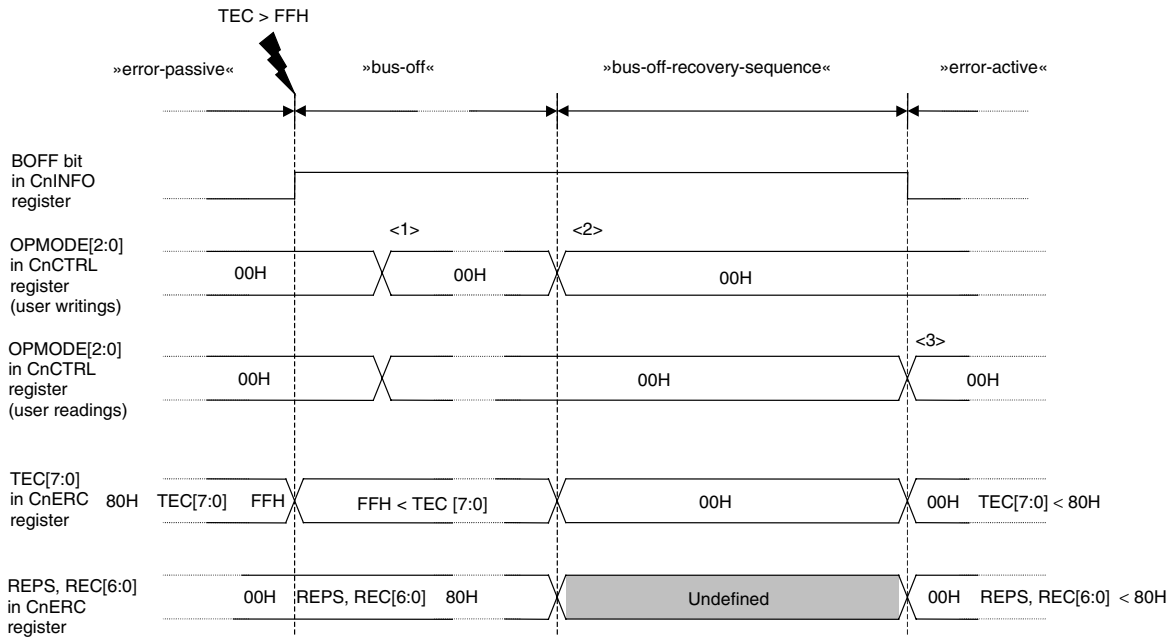
<1> A request to enter the CAN initialization mode
<2> A request to enter a CAN operation mode
(a) Recovery operation through normal recovery sequence
(b) Forced recovery operation that skips recovery sequence

(a) Recovery operation from bus-off state through normal recovery sequence

The CAN module first issues a request to enter the initialization mode (refer to timing <1> in "Recovery Operation from Bus-off State through Normal Recovery Sequence" on page 744). This request will be immediately acknowledged, and the OPMODE bits of the CnCTRL register are cleared to 000B. Processing such as analyzing the fault that has caused the bus-off state, re-defining the CAN module and message buffer using application software, or stopping the operation of the CAN module can be performed by clearing the GOM bit to 0.
Next, the user requests to change the mode from the initialization mode to an operation mode (refer to timing <2> in Figure 18-17, "Recovery Operation from Bus-off State through Normal Recovery Sequence," on page 744). This starts an operation to recover the CAN module from the bus-off state. The conditions under which the module can recover from the bus-off state are defined by the CAN protocol ISO 11898, and it is necessary to detect 11 consecutive recessive-level bits 128 times. At this time, the request to change the mode to an operation mode is held pending until the recovery conditions are satisfied. When the recovery conditions are satisfied (refer to timing <3> in "Recovery Operation from Bus-off State through Normal Recovery Sequence" on page 744), the CAN module can enter the operation mode it has requested. Until the CAN module enters this operation mode, it stays in the initialization mode. Whether the CAN module has completely entered the operation mode can be confirmed by reading the OPMODE bits of the CnCTRL register.
During the bus-off period and bus-off recovery sequence, the BOFF bit of the CnINFO register stays set (to 1). In the bus-off recovery sequence, the reception error counter (bits REC0 to REC6) counts the number of times 11 consecutive recessive-level bits have been detected on the bus. Therefore, the recovery state can be checked by reading REC0 to REC6 bits.

Caution: In the bus-off recovery sequence, the reception error counter (bits REC0 to REC6) counts up (+1) each time 11 consecutive recessive-level bits have been detected. Even during the bus-off period, the CAN module can enter the CAN sleep mode or CAN stop mode. In this case, the bus-off recovery sequence starts not entering a CAN initialization mode at the same time when the CAN sleep mode is released. Note that the bus-off recovery sequence will start when the CAN module will be woken up by the detection of the dominant edge other than clearing PSMODE by software.

Figure 18-17: Recovery Operation from Bus-off State through Normal Recovery Sequence



(b) Forced recovery operation that skips bus-off recovery sequence

The CAN module can be forcibly released from the bus-off state, regardless of the bus state, by skipping the bus-off recovery sequence. Here is the procedure.
First, the CAN module requests to enter the initialization mode. For the operation and points to be noted at this time, refer to **18.3.6 (5) (a)"Recovery operation from bus-off state through normal recovery sequence" on page 743**.
Next, the module requests to enter an operation mode. At the same time, the CCERC bit of the CnCTRL register must be set to 1.
As a result, the bus-off recovery sequence defined by the CAN protocol ISO 11898 is skipped, and the module immediately enters the operation mode. In this case, the module is connected to the CAN bus after it has monitored 11 consecutive recessive-level bits. For details, refer to the processing in **Figure 18-78, "Setting CAN sleep Mode/Stop Mode," on page 854**.

Caution: This function is not defined by the CAN protocol ISO 11898. When using this function, thoroughly evaluate its effect on the network system.

(6) Initializing CAN module error counter register (CnERC) in initialization mode

If it is necessary to initialize the CAN module error counter register (CnERC) and CAN module information register (CnINFO) for debugging or evaluating a program, they can be initialized to the default value by setting the CCERC bit of the CnCTRL register in the initialization mode. When initialization has been completed, the CCERC bit is automatically cleared to 0.

Cautions: 1. This function is enabled only in the initialization mode. Even if the CCERC bit is set to 1 in a CAN operation mode, the CnERC and CnINFO registers are not initialized.

2. The CCERC bit can be set at the same time as the request to enter a CAN operation mode.

Remark: n = 0, 1

18.3.7 Baud rate control function

(1) Prescaler

The CAN controller has a prescaler that divides the clock (f_{CAN}) supplied to CAN. This prescaler generates a CAN protocol layer basic clock (f_{TQ}) that is the CAN module system clock (f_{CANMOD}) divided by 1 to 256 (refer to **18.6.13 and Figure 18-38, "CAN Module n Bit Rate Prescaler Register (CnBRP)," on page 781**).

(2) Data bit time (8 to 25 time quanta)

One data bit time is defined as shown in Figure 18-18.
The CAN controller sets time segment 1, time segment 2, and re-Synchronization Jump Width (SJW) as the parameter of data bit time, as shown in Figure 18-18. Time segment 1 is equivalent to the total of the propagation (prop) segment and phase segment 1 that are defined by the CAN protocol specification. Time segment 2 is equivalent to phase segment 2.

Figure 18-18: Segment Setting

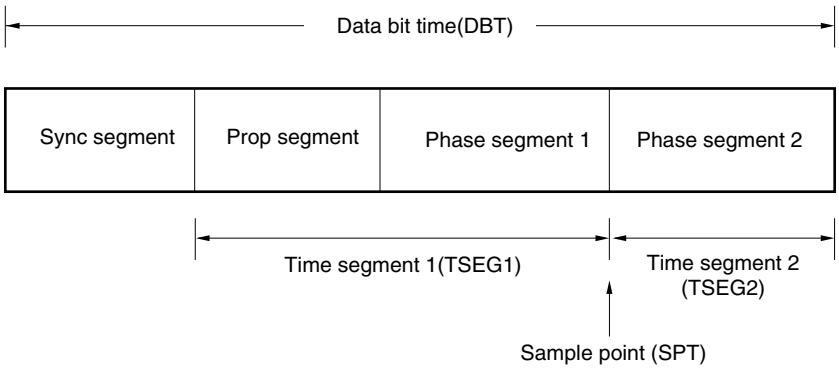


Table 18-16: Segment Setting

Segment Name	Settable Range	Notes on Setting to Confirm to CAN Specification
Time Segment 1 (TSEG1)	2TQ to 16TQ	—
Time Segment 2 (TSEG2)	1TQ to 8TQ	IPT**Note** of the CAN controller is 0TQ. To conform to the CAN protocol specification, therefore, a length equal to phase segment 1 must be set here. This means that the length of time segment 1 minus 1TQ is the settable upper limit of time segment 2.
Re-synchronization jump width (SJW)	1TQ to 4TQ	The length of time segment 1 minus 1TQ or 4 TQ, whichever is smaller.

Note: IPT: Information Processing Time

Remark: Reference: The CAN standard ISO 11898 specification defines the segments constituting the data bit time as shown in Figure 18-19 on the next page.

Figure 18-19: Configuration of Data Bit Time Defined by CAN Specification

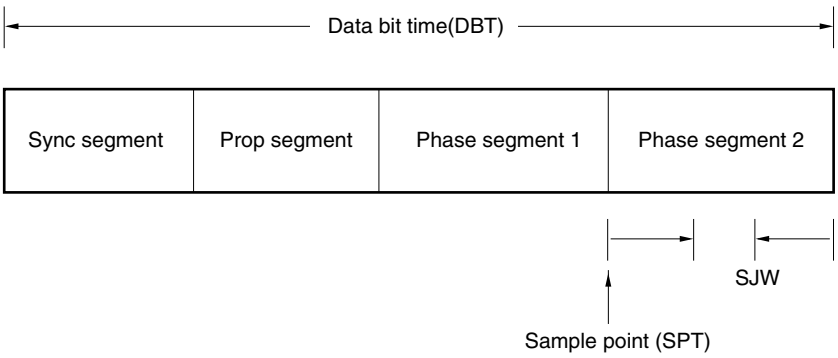


Table 18-17: Configuration of Data Bit Time Defined by CAN Specification

Segment Name	Segment Length	Description
Sync Segment (Synchronization Segment)	1	This segment starts at the edge where the level changes from recessive to dominant when hard-synchronization is established.
Prop Segment	Programmable to 1 to 8 or more	This segment absorbs the delay of the output buffer, CAN bus, and input buffer. The length of this segment is set so that ACK is returned before the start of phase segment 1. Time of prop segment ≥ (Delay of output buffer) + 2 × (Delay of CAN bus) + (Delay of input buffer) This segment compensates for an error of data bit time. The longer this segment, the wider the permissible range but the slower the communication speed.
Phase Segment 1	Programmable to 1 to 8	
Phase Segment 2	Phase Segment 1 or IPT**Note**, whichever greater	
SJW	Programmable from 1TQ to length of segment 1 or 4TQ, whichever is smaller	This width sets the upper limit of expansion or contraction of the phase segment during re-synchronization

Note: IPT: Information Processing Time

(3) Synchronizing data bit

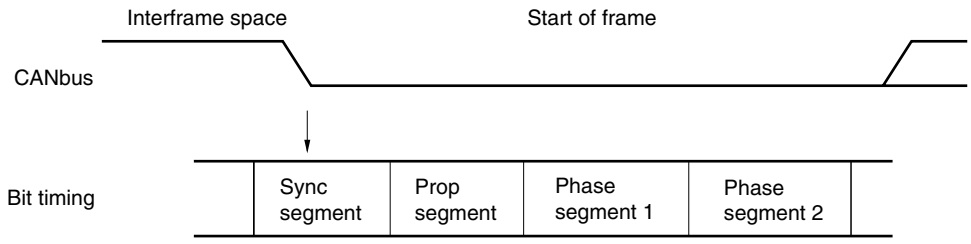
- The receiving node establishes synchronization by a level change on the bus because it does not have a sync signal.
- The transmitting node transmits data in synchronization with the bit timing of the transmitting node.

(a) Hard-synchronization

This synchronization is established when the receiving node detects the start of frame in the interframe space.

- When a falling edge is detected on the bus, that TQ means the sync segment and the next segment is the prop segment. In this case, synchronization is established regardless of SJW.

Figure 18-20: Hard-synchronization at Recognition of Dominant Level during Bus Idle



(b) Re-synchronization

Synchronization is established again if a level change is detected on the bus during reception (only if a recessive level was sampled previously).

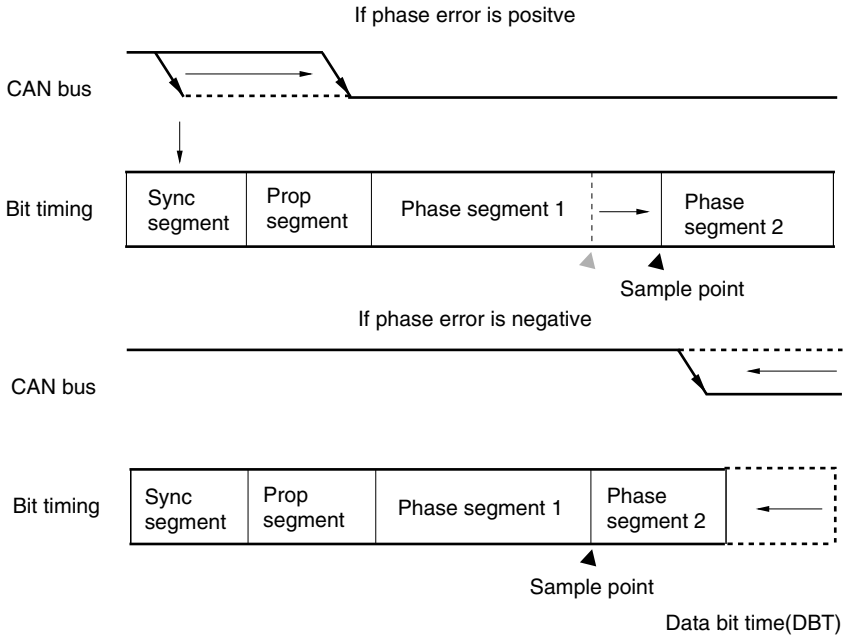
- The phase error of the edge is given by the relative position of the detected edge and sync segment.
- **<Sign of phase error>**

0: If the edge is within the sync segment
Positive: If the edge is before the sample point (phase error)
Negative: If the edge is after the sample point (phase error)

If phase error is positive: Phase segment 1 is longer by specified SJW.
If phase error is negative: Phase segment 2 is shorter by specified SJW.

- The sample point of the data of the receiving node moves relatively due to the “discrepancy” in baud rate between the transmitting node and receiving node.

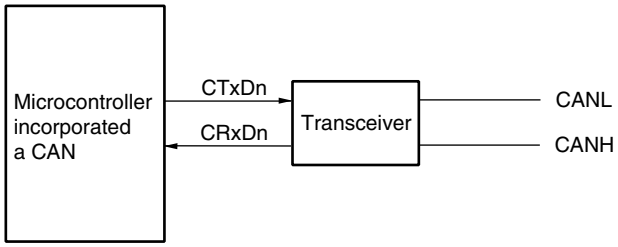
Figure 18-21: Re-synchronization



18.4 Connection With Target System

The microcontroller incorporated a CAN has to be connected to the CAN bus using an external transceiver.

Figure 18-22: Connection to CAN Bus



Remark: n = 0, 1

18.5 Internal Registers of CAN Controller

18.5.1 CAN controller configuration

Table 18-18: List of CAN Controller Registers (1/2)

Item	Register Name
CAN global registers	CAN global control register (CnGMCTRL)
	CAN global clock selection register (CnGMCS)
	CAN global automatic block transmission control register (CnGMABT)
	CAN global configuration register (CnGMCONF)
	CAN global automatic block transmission delay register (CnGMABTD)
CAN module registers	CAN module mask 1 register (CnMASK1L, CnMASK1H)
	CAN module mask 2 register (CnMASK2L, CnMASK2H)
	CAN module mask3 register (CnMASK3L, CnMASK3H)
	CAN module mask 4 registers (CnMASK4L, CnMASK4H)
	CAN module control register (CnCTRL)
	CAN module last error code register (CnLEC)
	CAN module information register (CnINFO)
	CAN module error counter register (CnERC)
	CAN module interrupt enable register (CnIE)
	CAN module interrupt status register (CnINTS)
	CAN module bit rate prescaler register (CnBRP)
	CAN module bit rate register (CnBTR)
	CAN module last in-pointer register (CnLIPT)
	CAN module receive history list register (CnRGPT)
	CAN module last out-pointer register (CnLOPT)
	CAN module transmit history list register (CnTGPT)
	CAN module time stamp register (CnTS)
Message buffer registers	CAN message data byte 01 register m (CnMDATA01m)
	CAN message data byte 0 register m (CnMDATA0m)
	CAN message data byte 1 register m (CnMDATA1m)
	CAN message data byte 23 register m (CnMDATA23m)
	CAN message data byte 2 register m (CnMDATA2m)
	CAN message data byte 3 Register m (CnMDATA3m)
	CAN message data byte 45 Register m (CnMDATA45m)
	CAN message data byte 4 Register m (CnMDATA4m)
	CAN message data byte 5 Register m (CnMDATA5m)
	CAN message data byte 67 Register m (CnMDATA67m)
	CAN message data byte 6 register m (CnMDATA6m)
	CAN message data byte 7 register m (CnMDATA7m)
	CAN message data length register m (CnMDLCm)

Table 18-18: List of CAN Controller Registers (2/2)

Item	Register Name
Message buffer registers	CAN message register m (CnMCONFm)
	CAN message ID register m (CnMIDLm, CnMIDHm)
	CAN message control register m (CnMCTRLm)

Remarks: 1. CAN global registers are identified by CnGM <register function>.
CAN module registers are identified by Cn <register function>.
Message buffer registers are identified by CnM <register function>.

2. n = 0, 1 (number of channel)
m = 0 to 31 (number of buffer)

18.5.2 Register access type

Table 18-19: CAN Global Register Access Types

Offset Address Note	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
00H	CAN global control register	CnGMCTRL	R/W	–	–	×	0000H
02H	CAN global clock selection register	CnGMCS	R/W	–	×	–	0FH
04H	CAN global configuration register	CnGMCONF	R	–	×	×	19H
06H	CAN global automatic block transmission control register	CnGMABT	R/W	–	–	×	0000H
08H	CAN global automatic block transmission delay register	CnGMABTD	R/W	–	×	–	00H

Note: Offset of Global Register Area is 000H.
The actual register address is calculated as follows:

Register Address = Peripheral programmable area address (BPC)
(01FFC000H, when recommended BPC value is set)
+ Channel offset (refer to **Table 18-1**)
+ Global register area offset (= 000H)
+ Offset address as listed in **Table 18-19** above

Remark: n = 0, 1 (number of channel)

Table 18-20: CAN Module Register Access Types

Offset Address **Note**	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
00H	CAN module mask 1 register	CnMASK1L	R/W	–	–	×	undefined
02H		CnMASK1H					
04H	CAN module mask 2 register	CnMASK2L	R/W	–	–	×	undefined
06H		CnMASK2H					
08H	CAN module mask 3 register	CnMASK3L	R/W	–	–	×	undefined
0AH		CnMASK3H					
0CH	CAN module mask 4 register	CnMASK4L	R/W	–	–	×	undefined
0EH		CnMASK4H					
10H	CAN module control register	CnCTRL	R/W	–	–	×	0000H
12H	CAN module last error code register	CnLEC	R/W	–	×	–	00H
13H	CAN module information register	CnINFO	R	–	×	–	00H
14H	CAN module error counter register	CnERC	R	–	–	×	0000H
16H	CAN module interrupt enable register	CnIE	R/W	–	–	×	0000H
18H	CAN module interrupt status register	CnINTS	R/W	–	–	×	0000H
1AH	CAN module bit rate prescaler register	CnBRP	R/W	–	×	–	FFH
1CH	CAN module bit rate register	CnBTR	R/W	–	–	×	370FH
1EH	CAN module last in-pointer register	CnLIPT	R	–	×	–	undefined
20H	CAN module receive history list register	CnRGPT	R/W	–	–	×	xx02H
22H	CAN module last out-pointer register	CnLOPT	R	–	×	–	undefined
24H	CAN module transmit history list register	CnTGPT	R/W	–	–	×	xx02H
26H	CAN module time stamp register	CnTS	R/W	–	–	×	0000H

Note: Offset of CAN Module Register Area is 040H.
The actual register address is calculated as follows:

Register Address = Peripheral programmable area address (BPC)
(01FFC000H, when recommended BPC value is set)
+ Channel offset (ref. to **Table 18-1**)
+ CAN module register area offset (= 040H)
+ Offset address as listed in **Table 18-20** above

Remark: n = 0, 1 (number of channel)

Table 18-21: Message Buffer Access Types

Offset Address **Note**	Register Name	Symbol	R/W	Bit Manipulation Units			Default value
				1	8	16	
00H	CAN message data byte 01 register m	CnMDATA01m	R/W	–	–	×	undefined
00H	CAN message data byte 0 register m	CnMDATA0m	R/W	–	×	–	undefined
01H	CAN message data byte 1 register m	CnMDATA1m	R/W	–	×	–	undefined
02H	CAN message data byte 23 register m	CnMDATA23m	R/W	–	–	×	undefined
02H	CAN message data byte 2 register m	CnMDATA2m	R/W	–	×	–	undefined
03H	CAN message data byte 3 register m	CnMDATA3m	R/W	–	×	–	undefined
04H	CAN message data byte 45 register m	CnMDATA45m	R/W	–	–	×	undefined
04H	CAN message data byte 4 register m	CnMDATA4m	R/W	–	×	–	undefined
05H	CAN message data byte 5 register m	CnMDATA5m	R/W	–	×	–	undefined
06H	CAN message data byte 67 register m	CnMDATA67m	R/W	–	–	×	undefined
06H	CAN message data byte 6 register m	CnMDATA6m	R/W	–	×	–	undefined
07H	CAN message data byte 7 register m	CnMDATA7m	R/W	–	×	–	undefined
08H	CAN message data length register m	CnMDLCm	R/W	–	×	–	0000xxxxB
09H	CAN message configuration register m	CnMCONFm	R/W	–	×	–	undefined
0AH	CAN message ID register m	CnMIDLm	R/W	–	–	×	undefined
0CH		CnMIDHm					
0EH	CAN message control register m	CnMCTRLm	R/W	–	–	×	00x00000 000xx000B

Note: Offset of message buffer area is 100H.
The actual register address is calculated as follows:

Register address = Peripheral programmable area address (BPC)
(01FFC000H, when recommended BPC value is set)
+ Channel offset (ref. to **Table 18-1**)
+ Message buffer area offset (= 100H)
+ Message buffer offset (= m × 020H)
+ Offset address as listed in **Table 18-21** above

Remark: n = 0, 1 (number of channel)
m = 0 to 31 (number of buffer)

18.5.3 Register bit configuration

Table 18-22: Bit Configuration of CAN Global Registers

Offset Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
00H	CnGMCTRL (W)	0	0	0	0	0	0	0	Clear GOM
01H		0	0	0	0	0	0	Set EFSD	Set GOM
00H	CnGMCTRL (R)	0	0	0	0	0	0	EFSD	GOM
01H		MBON	0	0	0	0	0	0	0
02H	CnGMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0
04H	CnGMCONF	Undefined (reserved for future use)		GCONF5	GCONF4	GCONF3	GCONF2	GCONF1	GCONF0
06H	CnGMABT (W)	0	0	0	0	0	0	0	Clear ABTTRG
07H		0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
06H	CnGMABT (R)	0	0	0	0	0	0	ABTCLR	ABTTRG
07H		0	0	0	0	0	0	0	0
08H	CnGMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

Remark: n = 0, 1 (number of channel)
m = 0 to 31 (number of buffer)

Table 18-23: Bit Configuration of CAN Module Registers (1/2)

Offset Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
00H	CnMASK1L	CM1ID [7:0]							
01H		CM1ID [15:8]							
02H	CnMASK1H	CM1ID [23:16]							
03H		0	0	0	CM1ID [28:24]				
04H	CnMASK2L	CM2ID [7:0]							
05H		CM2ID [15:8]							
06H	CnMASK2H	CM2ID [23:16]							
07H		0	0	0	CM2ID [28:24]				
08H	CnMASK3L	CM3ID [7:0]							
09H		CM3ID [15:8]							
0AH	CnMASK3H	CM3ID [23:16]							
0BH		0	0	0	CM3ID [28:24]				
0CH	CnMASK4L	CM4ID [7:0]							
0DH		CM4ID [15:8]							
0EH	CnMASK4H	CM4ID [23:16]							
0FH		0	0	0	CM4ID [23:16]				
10H	CnCTRL (W)	0	Clear AL	Clear VALID	Clear PSMODE1	Clear PSMODE0	Clear OPMODE2	Clear OPMODE1	Clear OPMODE0
11H		Set CCERC	Set AL	0	Set PSMODE1	Set PSMODE0	Set OPMODE2	Set OPMODE1	Set OPMODE0
10H	CnCTRL (R)	CCERC	AL	VALID	PSMODE1	PSMODE0	OPMODE2	OPMODE1	OPMODE0
11H		0	0	0	0	0	0	RSTAT	TSTAT
12H	CnLEC (W)	0	0	0	0	0	0	0	0
12H	CnLEC (R)	0	0	0	0	0	LEC2	LEC1	LEC0
13H	CnINFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0
14H	CnERC	TEC[7:0]							
15H		REPS	REC[6:0]						
16H	CnIE (W)	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0
17H		0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
16H	CnIE (R)	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
17H		0	0	0	0	0	0	0	0
18H	CnINTS (W)	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0
19H		0	0	0	0	0	0	0	0
18H	CnINTS (R)	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0
19H		0	0	0	0	0	0	0	0
1AH	CnBRP	TQPRS[7:0]							
1BH	–	Access prohibited (reserved for future use)							
1CH	CnBTR	0	0	0	0	TSEG1[3:0]			
1DH		0	0	SJW[1:0]		0	TSEG2[2:0]		

Table 18-23: Bit Configuration of CAN Module Registers (2/2)

Offset Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
1EH	CnLIPT	LIPT[7:0]							
1FH	–	Access prohibited (reserved for future use)							
20H	CnRGPT (W)	0	0	0	0	0	0	0	Clear ROVF
21H		0	0	0	0	0	0	0	0
20H	CnRGPT (R)	0	0	0	0	0	0	RHPM	ROVF
21H		RGPT[7:0]							
22H	CnLOPT	LOPT[7:0]							
23H	–	Access prohibited (reserved for future use)							
24H	CnTGPT (W)	0	0	0	0	0	0	0	Clear TOVF
25H		0	0	0	0	0	0	0	0
24H	CnTGPT (R)	0	0	0	0	0	0	THPM	TOVF
25H		TGPT[7:0]							
26H	CnTS (W)	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN
27H		0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
26H	CnTS (R)	0	0	0	0	0	TSLOCK	TSSEL	TSEN
27H		0	0	0	0	0	0	0	0
28H-3FH	–	Access prohibited (reserved for future use)							

Remark: n = 0, 1 (number of channel)

Table 18-24: Bit Configuration of Message Buffer Registers

Offset Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
00H	CnMDATA01m	Message data (byte0)							
01H		Message data (byte1)							
00H	CnMDATA0m	Message data (byte 0)							
01H	CnMDATA1m	Message data (byte 1)							
02H	CnMDATA23m	Message data (byte 2)							
03H		Message data (byte 3)							
02H	CnMDATA2m	Message data (byte 2)							
03H	CnMDATA3m	Message data (byte 3)							
04H	CnMDATA45m	Message data (byte 4)							
05H		Message data (byte 5)							
04H	CnMDATA4m	Message data (byte 4)							
05H	CnMDATA5m	Message data (byte 5)							
06H	CnMDATA67m	Message data (byte 6)							
07H		Message data (byte 7)							
06H	CnMDATA6m	Message data (byte 6)							
07H	CnMDATA7m	Message data (byte 7)							
08H	CnMDLCm	0				MDLC3	MDLC2	MDLC1	MDLC0
09H	CnMCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0
0AH	CnMIDLm	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
0BH		ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
0CH	CnMIDHm	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
0DH		IDE	0	0	ID28	ID27	ID26	ID25	ID24
0EH	CnMCTRLm (W)	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY
0FH		0	0	0	0	Set IE	0	Set TRQ	Set RDY
0EH	CnMCTRLm (R)	0	0	0	MOW	IE	DN	TRQ	RDY
0FH		0	0	MUC	0	0	0	0	0
10-1FH	–	Access prohibited (reserved for future)							

Remark: n = 0, 1 (number of channel)
m = 0 to 31 (number of buffer)

18.6 Control Registers

18.6.1 CAN global control register (CnGMCTRL)

The CnGMCTRL register is a 16-bit register used to control the operation of the CAN module.
This register can be read in 16-bit, 8-bit or 1-bit units. Write operation can be performed by a special bit set/clear function as described in **18.7 Bit Set/Clear Function**.
Reset input clears this register to 0000H.

Figure 18-23: CAN Global Control Register (CnGMCTRL) (1/2)

(a) Read Operation

After reset: 0000H Read Offset address: 000H

	15	14	13	12	11	10	9	8
CnGMCTRL (n = 0, 1)	MBON	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	EFSD	GOM

MBON	Bit Enabling Access to Message Buffer Register, Transmit/Receive History List Registers
0	Write access and read access to the message buffer register and the transmit/receive history list registers is disabled.
1	Write access and read access to the message buffer register and the transmit/receive history list registers is enabled.
• MBON bit is cleared (to 0) when the CAN module enters CAN sleep mode/CAN stop mode or GOM bit is cleared (to 0). • MBON bit is set (to 1) when the CAN sleep mode/the CAN stop mode is released or GOM bit is set (to 1).	

Cautions: 1. While the MBON bit is cleared (to 0), software access to the message buffers (CnMDATA0m to CnMDATA7m, CnMDATA01m, CnMDATA23m, CnMDATA45m, CnMDATA67m, CnMDLCm, CnMCONFm, CnMIDLm, CnMIDHm, and CnMCTRLm), or registers related to transmit history or receive history (CnLOPT, CnTGPT, CnLIPT, and CnRGPT) is disabled.

2. The MBON bit is read-only. Even if 1 is written to MBON while it is 0, the value of MBON does not change, and access to the message buffer registers, or registers related to transmit history or receive history remains disabled.

EFSD	Bit Enabling Forced Shut Down
0	Forced shut down by GOM = 0 disabled.
1	Forced shut down by GOM = 0 enabled.

Caution: To request forced shut down, the GOM bit must be cleared to 0 immediately after the EFSD bit has been set to 1. If access to another register (including reading the CnGMCTRL register) is executed without clearing the GOM bit immediately after the EFSD bit has been set to 1, the EFSD bit is forcibly cleared to 0, and the forced shut down request is invalid.

Figure 18-23: CAN Global Control Register (CnGMCTRL) (2/2)

GOM	Global Operation Mode Bit
0	CAN module is disabled from operating.
1	CAN module is enabled to operate.

Caution: The GOM bit can be cleared only in the initialization mode or immediately after EFSD bit is set (to 1).

(b) Write Operation

After reset: – Write Offset address: 000H

CnGMCTRL (n = 0, 1)

15	14	13	12	11	10	9	8
0	0	0	0	0	0	Set EFSD	Set GOM
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	Clear GOM

Set EFSD	EFSD Bit Setting
0	No change in ESFD bit
1	EFSD bit set to 1

Set GOM	Clear GOM	GOM Bit Setting
0	1	GOM bit cleared to 0
1	0	GOM bit set to 1
Other than above		No change in GOM bit

Remark: n = 0, 1

18.6.2 CAN global clock selection register (CnGMCS)

The CnGMCS register is an 8-bit register used to select the CAN module system clock.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 0FH.

Figure 18-24: CAN Global Clock Selection Register (CnGMCS)

After reset: 0FH R/W Offset address: 002H

	7	6	5	4	3	2	1	0
CnGMCS (n = 0, 1)	0	0	0	0	CCP3	CCP2	CCP1	CCP0

CCP3	CCP2	CCP1	CCP1	CAN Module System Clock (f_{CANMOD})
0	0	0	0	$f_{CAN}/1$
0	0	0	1	$f_{CAN}/2$
0	0	1	0	$f_{CAN}/3$
0	0	1	1	$f_{CAN}/4$
0	1	0	0	$f_{CAN}/5$
0	1	0	1	$f_{CAN}/6$
0	1	1	0	$f_{CAN}/7$
0	1	1	1	$f_{CAN}/8$
1	0	0	0	$f_{CAN}/9$
1	0	0	1	$f_{CAN}/10$
1	0	1	0	$f_{CAN}/11$
1	0	1	1	$f_{CAN}12$
1	1	0	0	$f_{CAN}/13$
1	1	0	1	$f_{CAN}/14$
1	1	1	0	$f_{CAN}/15$
1	1	1	1	$f_{CAN}/16$ (default value)

Remarks: 1. f_{CAN} = Clock supply to CAN = $f_{XX}/4$

2. n = 0, 1

18.6.3 CAN global automatic block transmission control register (CnGMABT)

The CnGMABT register is a 16-bit register used to control the automatic block transmission (ABT) operation.
This register can be read in 16-bit, 8-bit or 1-bit units. Write operation can be performed by a special bit set/clear function as described in **18.7 Bit Set/Clear Function**.
Reset input clears this register to 0000H.

Figure 18-25: CAN Global Automatic Block Transmission Control Register (CnGMABT) (1/2)

(a) Read Operation

After reset: 0000H Read Offset address: 006H

	15	14	13	12	11	10	9	8
CnGMABT (n = 0, 1)	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	ABTCLR	ABTTRG

ABTCLR	Automatic Block Transmission Engine Clear Status Bit
0	Clearing the automatic transmission engine is completed
1	The automatic transmission engine is being cleared.

- Set the ABTCLR bit to 1 while the ABTTRG bit is cleared (0).
  The operation is not guaranteed if the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1.
- When the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared to 0 as soon as the requested clearing processing is complete.

ABTTRG	Automatic Block Transmission Status Bit
0	Automatic block transmission is stopped.
1	Automatic block transmission is under execution.

Remark: n = 0, 1

Figure 18-25: CAN Global Automatic Block Transmission Control Register (CnGMABT) (2/2)

(b) Write Operation

After reset: – Write Offset address: 006H

CnGMABT (n = 0, 1)

15	14	13	12	11	10	9	8
0	0	0	0	0	0	Set ABTCLR	Set ABTTRG

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	Clear ABTTRG

Set ABTCLR	Automatic Block Transmission Engine Clear Request Bit
0	The automatic block transmission engine is in idle state or under operation.
1	Request to clear the automatic block transmission engine. After the automatic block transmission engine has been cleared, automatic block transmission is started from message buffer 0 by setting the ABTTRG bit to 1.

Set ABTTRG	Clear ABTTRG	Automatic Block Transmission Start Bit
0	1	Request to stop automatic block transmission.
1	0	Request to start automatic block transmission.
Other than above		No change in ABTTRG bit.

Cautions: 1. Before changing the normal operation mode with ABT to the initialization mode, be sure to set the CnGMABT register to the default value (00H).

2. Do not set the ABTTRG bit (ABTTRG = 1) in the initialization mode. If the ABTTRG bit is set in the initialization mode, the operation is not guaranteed after the CAN module has entered the normal operation mode with ABT.

Remark: n = 0, 1

18.6.4 CAN global configuration register (CnGMCONF)

The CnGMCONF register a 16-bit register that provides the configuration information of the CAN module.
This register can be read only in 16-bit, 8-bit or 1-bit units.
Reset input clears this register to 0011H.

Figure 18-26: CAN Global Configuration Register (CnGMCONF)

After reset: 0011H Read Offset address: 004H

CnGMCONF (n = 0, 1)

15	14	13	12	11	10	9	8
Note 1							GCONF8

7	6	5	4	3	2	1	0
Note 1		GCONF5	GCONF4	GCONF3	GCONF2	GCONF1	GCONF0

GCONF8	Mirror Function for DIAG macro
0	no Mirror function is implemented
1	Mirror function is implemented

GCONF5	GCONF4	GCONF3	Number of message buffers
0	0	1	16 message buffers
0	1	0	32 message buffers
0	1	1	48 message buffers
Other than above			Reserved

GCONF2	GCONF1	GCONF0	Number of CAN interface channels[Note 2]
0	0	1	1 CAN interface channels
Other than above			Reserved

Notes: 1. Undefined (reserved for future use)

2. This is not the number of the channels of the device.

Remark: n = 0, 1

18.6.5 CAN global automatic block transmission delay register (CnGMABTD)

The CnGMABTD register is an 8-bit register used to set the interval at which the data of the message buffer assigned to ABT is to be transmitted in the normal operation mode with ABT.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Figure 18-27: CAN Global Automatic Block Transmission Delay Register (CnGMABTD)

After reset: 00H R/W Offset address: 008H

CnGMABT (n = 0, 1)

7	6	5	4	3	2	1	0
0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

ABTD3	ABTD2	ABTD1	ABTD0	Data frame interval during automatic block transmission (unit: Data bit time (DBT))
0	0	0	0	0 DBT (default value)
0	0	0	1	2^5 DBT
0	0	1	0	2^6 DBT
0	0	1	1	2^7 DBT
0	1	0	0	2^8 DBT
0	1	0	1	2^9 DBT
0	1	1	0	2^{10} DBT
0	1	1	1	2^{11} DBT
1	0	0	0	2^{12} DBT
Other than above				Setting prohibited

Cautions: 1. Do not change the contents of the CnGMABTD register while the ABTTRG bit is set to 1.

2. The timing at which the ABT message is actually transmitted onto the CAN bus differs depending on the status of transmission from the other station or how a request to transmit a message other than an ABT message (message buffers 8 to m_{MAX}–1) is made.

Remark: n = 0, 1
m_{MAX} = 32

18.6.6 CAN module mask control registers (CnMASKmL, CnMASKmH)

The CnMASKmL and CnMASKmH registers are 16-bit registers used to extend the number of receivable messages into the same message buffer by masking part of the ID comparison of a message and invalidating the ID of the masked part.
These registers can be read and written in 16-bit, 8-bit or 1-bit units.
After reset input the registers are undefined.

Figure 18-28: CAN Module n Mask 1 Registers L, H (CnMASK1L, CnMASK1H)

After reset: undefined R/W Offset address: 000H

CnMASK1L (n = 0, 1)

15	14	13	12	11	10	9	8
CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
7	6	5	4	3	2	1	0
CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

After reset: undefined R/W Offset address: 002H

CnMASK1H (n = 0, 1)

15	14	13	12	11	10	9	8
0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
7	6	5	4	3	2	1	0
CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

CMIDx	Sets Mask Pattern of ID Bit.
0	The ID bit x of the message buffer is compared with the ID bit x of the received message frame
1	The ID bit x of the message buffer is not compared with the ID bit x of the received message frame (ID bit x is masked).

Remarks: 1. Masking is always defined by an ID length of 29 bits. If a mask is assigned to a message with a standard ID, CMID17 to CMID0 are ignored. Therefore, only CMID28 to CMID18 of the received ID are masked. The same mask can be used for both the standard and extended IDs.

2. n = 0, 1
m = 1 to 4
x = 0 to 28

Figure 18-29: CAN Module n Mask 2 Registers L. H (CnMASK2L, CnMASK2H)

After reset: undefined R/W Offset address: 004H

CnMASK2L (n = 0, 1)

15	14	13	12	11	10	9	8
CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
7	6	5	4	3	2	1	0
CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

After reset: undefined R/W Offset address: 006H

CnMASK2H (n = 0, 1)

15	14	13	12	11	10	9	8
0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
7	6	5	4	3	2	1	0
CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

CMIDx	Sets Mask Pattern of ID Bit.
0	The ID bit x of the message buffer is compared with the ID bit x of the received message frame
1	The ID bit x of the message buffer is not compared with the ID bit x of the received message frame (ID bit x is masked).

Remarks: 1. Masking is always defined by an ID length of 29 bits. If a mask is assigned to a message with a standard ID, CMID17 to CMID0 are ignored. Therefore, only CMID28 to CMID18 of the received ID are masked. The same mask can be used for both the standard and extended IDs.

2. n = 0, 1
x = 0 to 28

Figure 18-30: CAN Module n Mask 3 Registers (CnMASK3L, CnMASK3H))

After reset: undefined R/W Offset address: 008H

CnMASK3L (n = 0, 1)

15	14	13	12	11	10	9	8
CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
7	6	5	4	3	2	1	0
CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

After reset: undefined R/W Offset address: 00AH

CnMASK3H (n = 0, 1)

15	14	13	12	11	10	9	8
0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
7	6	5	4	3	2	1	0
CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

CMIDx	Sets Mask Pattern of ID Bit.
0	The ID bit x of the message buffer is compared with the ID bit x of the received message frame
1	The ID bit x of the message buffer is not compared with the ID bit x of the received message frame (ID bit x is masked).

Remarks: 1. Masking is always defined by an ID length of 29 bits. If a mask is assigned to a message with a standard ID, CMID17 to CMID0 are ignored. Therefore, only CMID28 to CMID18 of the received ID are masked. The same mask can be used for both the standard and extended IDs.

2. n = 0, 1
x = 0 to 28

Figure 18-31: CAN Module n Mask 4 Registers (CnMASK4L, CnMASK4H))

After reset: undefined R/W Offset address: 00CH

CnMASK4L (n = 0, 1)

15	14	13	12	11	10	9	8
CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
7	6	5	4	3	2	1	0
CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

After reset: undefined R/W Offset address: 00EH

CnMASK4H (n = 0, 1)

15	14	13	12	11	10	9	8
0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
7	6	5	4	3	2	1	0
CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

CMIDx	Sets Mask Pattern of ID Bit.
0	The ID bit x of the message buffer is compared with the ID bit x of the received message frame
1	The ID bit x of the message buffer is not compared with the ID bit x of the received message frame (ID bit x is masked).

Remarks: 1. Masking is always defined by an ID length of 29 bits. If a mask is assigned to a message with a standard ID, CMID17 to CMID0 are ignored. Therefore, only CMID28 to CMID18 of the received ID are masked. The same mask can be used for both the standard and extended IDs.

2. n = 0, 1
x = 0 to 28

18.6.7 CAN module control register (CnCTRL)

The CnCTRL register is a 16-bit register that controls the operation mode of the CAN module.
This register can be read in 16-bit, 8-bit or 1-bit units. Write operation can be performed by a special bit set/clear function as described in **18.7 "Bit Set/Clear Function" on page 801**.
Reset input clears this register to 0000H.

Figure 18-32: CAN Module n Control Register (CnCTRL) (1/4)

(a) Read Operation

After reset: 0000H Read Offset address: 010H

CnCTRL (n = 0, 1)

15	14	13	12	11	10	9	8
0	0	0	0	0	0	RSTAT	TSTAT

7	6	5	4	3	2	1	0
CCERC	AL	VALID	PSMODE1	PSMODE0	OPMODE2	OPMODE1	OPMODE0

RSTAT	Reception Status Bit
0	Reception is stopped
1	Reception is in progress
The RSTAT bit is set to 1 under the following conditions (timing) • The SOF bit of a receive frame is detected • On occurrence of arbitration loss during a transmit frame • The RSTAT bit is cleared to 0 under the following conditions (timing) • When a recessive level is detected at the second bit of the interframe space • On transition to the initialization mode at the first bit of the interframe space	

TSTAT	Transmission Status Bit
0	Transmission is stopped.
1	Transmission is in progress
The TSTAT bit is set to 1 under the following conditions (timing): • The SOF bit of a transmit frame is detected • The first bit of an error flag is detected during a transmit frame • The TSTAT bit is cleared to 0 under the following conditions (timing) • During transition to bus-off state • On occurrence of arbitration loss in transmit frame • On detection of recessive level at the second bit of the interframe space • On transition to the initialization mode at the first bit of the interframe space	

Remark: n = 0, 1

Figure 18-32: CANn Module Control Register (CnCTRL) Format (2/4)

CCERC	Error Counter Clear Bit
0	The CnERC and CnINFO registers are not cleared in the initialization mode.
1	The CnERC and CnINFO registers are cleared in the initialization mode
• The CCERC bit is used to clear the CnERC and CnINFO registers for re-initialization or forced recovery from the bus-off state. This bit can be set to 1 only in the initialization mode. • When the CnERC and CnINFO registers have been cleared, the CCERC bit is also cleared to 0 automatically. • The CCERC bit can be set to 1 at the same time as a request to change the initialization mode to an operation mode is made.4. The CCERC bit is read-only in the CAN sleep mode or CAN stop mode. • The receive data may be broken in case of setting the CCERC bit to (1) immediately after entering the INIT mode.	

AL	Bit to Set Operation in Case of Arbitration Loss
0	Re-transmission is not executed in case of an arbitration loss in the single-shot mode
1	Re-transmission is executed in case of an arbitration loss in the single-shot mode
The AL bit is valid only in the single-shot mode.	

VALID	Valid Receive Message Frame Detection Bit
0	A valid message frame has not been received since the VALID bit was last cleared to 0
1	A valid message frame has been received since the VALID bit was last cleared to 0
• Detection of a valid receive message frame is not dependent upon storage in the receive message buffer (data frame) or transmit message buffer (remote frame). • Clear the VALID bit (0) before changing the initialization mode to an operation mode. • If only two CAN nodes are connected to the CAN bus with one transmitting a message frame in the normal mode and the other in the reception mode, the VALID bit is not set to 1 before the transmitting node enters the error passive state. • In order to clear the VALID bit, set the Clear VALID bit to 1 first and confirm that the VALID bit is cleared. If it is not cleared, perform clearing processing again.	

PSMODE1	PSMODE0	Power Save Mode
0	0	No power save mode is selected
0	1	CAN sleep mode
1	0	Setting prohibited
1	1	CAN stop mode

Caution: Transition to and from the CAN stop mode must be made via CAN sleep mode. A request for direct transition to and from the CAN stop mode is ignored.

Remark: n = 0, 1

Figure 18-32: CANn Module Control Register (CnCTRL) Format (3/4)

PSMODE1	PSMODE0	Power Save Mode
0	0	No power save mode is selected
0	1	CAN sleep mode
1	0	Setting prohibited
1	1	CAN stop mode

OPMODE2	OPMODE1	OPMODE0	Operation Mode
0	0	0	No operation mode is selected (CAN module is in the initialization mode)
0	0	1	Normal operating mode
0	1	0	Normal operation mode with automatic block transmission function (normal operation mode with ABT)
0	1	1	Receive-only mode
1	0	0	Single-shot mode
1	0	1	Self-test mode
Other than above			Setting prohibited.
The OPMODE0 to OPMODE2 bits are read-only in the CAN sleep mode or CAN stop mode.			

(b) Write Operation

After reset: – Write Offset address: 010H

CnCTRL (n = 0, 1)

15	14	13	12	11	10	9	8
Set CCERC	Set AL	0	Set PSMODE1	Set PSMODE0	Set OPMODE2	Set OPMODE1	Set OPMODE0

7	6	5	4	3	2	1	0
0	Clear AL	Clear VALID	Clear PSMODE1	Clear PSMODE0	Clear OPMODE2	Clear OPMODE1	Clear OPMODE0

Set CCERC	Setting of CCERC Bit
0	CCERC bit is not changed
1	CCERC bit is set to 1

Set AL	Clear AL	Setting of AL Bit
0	1	AL bit is cleared to 0
1	0	AL bit is set to 1
1	0	Setting prohibited
1	1	CAN stop mode

Remark: n = 0, 1

Figure 18-32: CAN Module n Control Register (CnCTRL) (4/4)

Clear VALID	Setting of CCERC Bit
0	VALID bit is not changed
1	VALID bit is cleared to 0

Set PSMODE0	Clear PSMODE0	Setting of PSMODE0 Bit
0	1	PSMODE0 bit is cleared to 0
1	0	PSMODE bit is set to 1
Other than above		PSMODE0 bit is not changed

Set PSMODE1	Clear PSMODE1	Setting of PSMODE1 Bit
0	1	PSMODE1 bit is cleared to 0
1	0	PSMODE1 bit is set to 1
Other than above		PSMODE1 bit is not changed

Set OPMODE0	Clear OPMODE0	Setting of OPMODE0 Bit
0	1	OPMODE0 bit is cleared to 0
1	0	OPMODE0 bit is set to 1
Other than above		OPMODE0 bit is not changed

Set OPMODE1	Clear OPMODE1	Setting of OPMODE1 Bit
0	1	OPMODE1 bit is cleared to 0
1	0	OPMODE1 bit is set to 1
Other than above		OPMODE1 bit is not changed

Set OPMODE2	Clear OPMODE2	Setting of OPMODE2 Bit
0	1	OPMODE2 bit is cleared to 0
1	0	OPMODE2 bit is set to 1
Other than above		OPMODE2 bit is not changed

Remark: n = 0, 1

18.6.8 CAN module last error code register (CnLEC)

The CnLEC register provides the error information of the CAN protocol.

Figure 18-33: CAN Module n Last Error Code Register (CnLEC)

After reset: 00H R/W Offset address: 012H

	7	6	5	4	3	2	1	0
CnLEC (n = 0, 1)	0	0	0	0	0	LEC2	LEC1	LEC0

LEC2	LEC1	LEC0	Last CAN Protocol Error Information
0	0	0	No error
0	0	1	Stuff error
0	1	0	Form error
0	1	1	ACK error
1	0	0	Bit error (The CAN module tried to transmit a recessive-level bit as part of a transmit message (except the arbitration field), but the value on the CAN bus is a dominant-level bit.)
1	0	1	Bit error (The CAN module tried to transmit a dominant-level bit as part of a transmit message, ACK bit, error frame, or overload frame, but the value on the CAN bus is a recessive-level bit.)
1	1	0	CRC error
1	1	1	Undefined

Remarks: 1. The contents of the CnLEC register are not cleared when the CAN module changes from an operation mode to the initialization mode.

2. If an attempt is made to write a value other than 00H to the CnLEC register by software, the access is ignored.

3. n = 0, 1

18.6.9 CAN module information register (CnINFO)

The CnINFO register indicates the status of the CAN module.

Figure 18-34: CAN Module n Information Register (CnINFO)

After reset: 00H Read Offset address: 013H

	7	6	5	4	3	2	1	0
CnINFO (n = 0, 1)	0	0	0	BOFF	TEC1	TEC0	REC1	REC0

BOFF	Bus-off State Bit
0	Not bus-off state (transmit error counter ≤255) (The value of the transmit counter is less than 256)
1	Bus-off state (transmit error counter > 255) (The value of the transmit counter is 256 or more)

TECS1	TECS0	Transmission Error Counter Status Bit
0	0	The value of the transmission error counter is less than that of the warning level (<96)
0	1	The value of the transmission error counter is in the range of the warning level (96 to 127)
1	0	Undefined
1	1	The value of the transmission error counter is in the range of the error passive or bus-off state (≥ 128

RECS1	RECS0	Reception Error Counter Status Bit
0	0	The value of the reception error counter is less than that of the warning level (<96)
0	1	The value of the reception error counter is in the range of the warning level (96 to 127)
1	0	Undefined
1	1	The value of the reception error counter is in the error passive range (≥ 128)

Remark: n = 0, 1

18.6.10 CAN module error counter register (CnERC)

The CnERC register indicates the count value of the transmission/reception error counter.

Figure 18-35: CAN Module n Error Counter Register (CnERC)

After reset: 0000H Read Offset address: 014H

CnERC (n = 0, 1)

15	14	13	12	11	10	9	8
REPS	REC6	REC5	REC4	REC3	REC2	REC1	REC0
7	6	5	4	3	2	1	0
TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

REPS	Reception Error Passive Status Bit
0	The reception error counter is not in the error passive range (< 128)
1	The reception error counter is in the error passive range (≥ 128)

REC6 to REC0	Reception Error Counter Bit
0 to 127	Number of reception errors. These bits reflect the status of the reception error counter. The number of errors is defined by the CAN protocol.
REC6 to REC0 bits of the reception error counter are invalid in the reception error passive state (RECS1, RECS0 = 11B)	

TEC7 to TEC0	Transmission Error Counter Bit
0 to 255	Number of transmission errors. These bits reflect the status of the transmission error counter. The number of errors is defined by the CAN protocol.
TEC7 to TEC0 bits of the transmission error counter are invalid in the bus-off state (BOFF = 1).	

Remark: n = 0, 1

18.6.11 CAN module interrupt enable register (CnIE)

The CnIE register is used to enable or disable the interrupts of the CAN module.

Figure 18-36: CAN Module n Interrupt Enable Register (CnIE) (1/2)

(a) Read Operation

After reset: 0000H Read Offset address: 016H

CnIE (n = 0, 1)	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0

CIEm	CAN Module Interrupt m Enable Bit
0	CAN module interrupt m corresponding to interrupt status register CINTSm is disabled
1	CAN module interrupt m corresponding to interrupt status register CINTSm is enabled.

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

Figure 18-36: CANn Module Interrupt Enable Register (CnIE) Format (2/2)

(b) Write Operation

After reset: – Write Offset address: 016H

CnIE (n = 0, 1)

15	14	13	12	11	10	9	8
0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
7	6	5	4	3	2	1	0
0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0

Set CIEm	Clear CIEm	Setting of CIEm Bit
0	1	CIEm bit is cleared to 0
1	0	CIEm bit is set to 1
Other than above		CIEm bit is not changed

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

18.6.12 CAN module interrupt status register (CnINTS)

The CnINTS register indicates the interrupt status of the CAN module.

Figure 18-37: CAN Module n Interrupt Status Register (CnINTS) (1/2)

(a) Read Operation

After reset: 0000H Read Offset address: 018H

CnINTS (n = 0, 1)	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0

CINTS5	Status Bit of Wakeup Interrupt from CAN Sleep Mode
0	No wakeup interrupt from CAN sleep mode is pending
1	Wakeup interrupt from CAN sleep mode is pending
The CINTS5 bit is set only when the CAN module is woken up from the CAN sleep mode by a CAN bus operation. The CINTS5 bit is not set when the CAN sleep mode has been released by software.	

CINTS4	Status Bit of Arbitration Loss Interrupt
0	No arbitration loss interrupt is pending
1	Arbitration loss interrupt is pending

CINTS3	Status Bit of CAN Protocol Error Interrupt
0	No CAN protocol error interrupt is pending
1	CAN protocol error interrupt is pending

CINTS2	Status Bit of CAN Error Status Interrupt
0	No CAN error status interrupt is pending
1	CAN error status interrupt is pending

CINTS1	Status Bit of Interrupt on Reception Completion
0	No interrupt on completion of reception of valid message frame to message buffer is pending
1	Interrupt on completion of reception of valid message frame to message buffer is pending

CINTS1	Status Bit of Interrupt on Transmission Completion
0	No interrupt on normal completion of transmission of message frame from message buffer is pending
1	Interrupt on normal completion of transmission of message frame from message buffer is pending

Figure 18-37: CAN Module Interrupt Status Register (CnINTS) Format (2/2)

(b) Write Operation

After reset: – Write Offset address: 018H

	15	14	13	12	11	10	9	8
CnINTS (n = 0, 1)	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0

Clear CINTSm	Setting of CIEm Bit
1	CINTSm bit is cleared to 0
0	CINTSm bit is set to 1

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

18.6.13 CAN module bit rate prescaler register (CnBRP)

The CnBRP register is used to select the CAN protocol layer basic clock (f_{TQ}). The communication baud rate is set to the CnBTR register.

Caution: The CnBRP register can be write-accessed only in the initialization mode.

Figure 18-38: CAN Module n Bit Rate Prescaler Register (CnBRP)

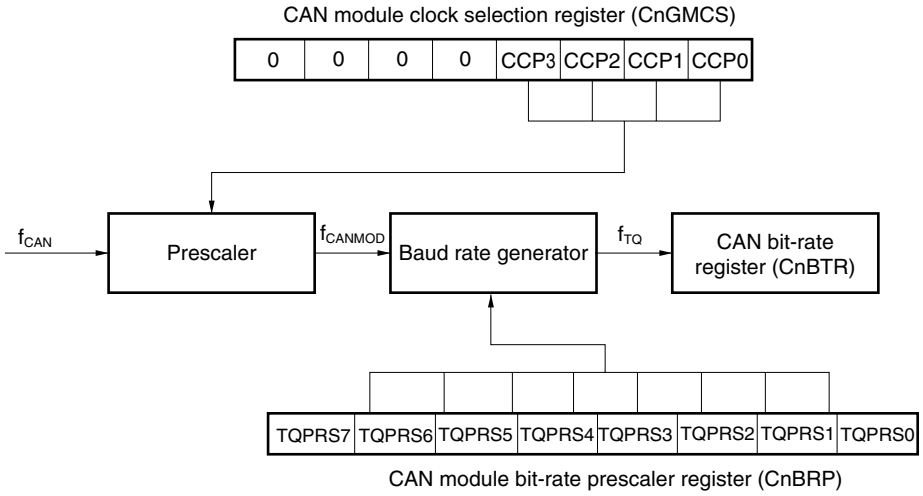
(a) Register Format

After reset: FFH R/W Offset address: 01AH

	7	6	5	4	3	2	1	0
CnBRP (n = 0, 1)	TQPRS7	TQPRS6	TQPRS5	TQPRS4	TQPRS3	TQPRS2	TQPRS1	TQPRS0

TQPRS7 to TQPRS0	CAN Protocol Layer Basic System Clock (f_{TQ})
0	$f_{CANMOD}/1$
1	$f_{CANMOD}/2$
...	...
n	$f_{CANMOD}/(n+1)$
...	...
255	$f_{CANMOD}/256$ (default value)

(b) Clock Supply of CAN Module



Remarks: 1. f_{CAN}: Clock supplied to CAN ($f_{CAN} = f_{XX}/4$)
f_{CANMOD}: CAN module system clock
f_{TQ}: CAN protocol layer basic system clock

2. n = 0, 1

18.6.14 CAN module bit rate register (CnBTR)

The CnBTR register is used to control the data bit time of the communication baud rate.

Figure 18-39: CAN Module n Bit Rate Register (CnBTR) (1/2)

(a) Register Format

After reset: 370FH R/W Offset address: 01CH

CnBTR (n = 0, 1)

15	14	13	12	11	10	9	8
0	0	SJW1	SJW0	0	TSEG22	TSEG21	TSEG20

7	6	5	4	3	2	1	0
0	0	0	0	TSEG13	TSEG12	TSEG11	TSEG10

SJW1	SJW0	Length of Synchronization Jump Width
0	0	1TQ
0	1	2TQ
1	0	3TQ
1	1	4TQ (default value)

TSEG22	TSEG21	TSEG20	Length of Time Segment 2
0	0	0	1TQ
0	0	1	2TQ
0	1	0	3TQ
0	1	1	4TQ
1	0	0	5TQ
1	0	1	6TQ
1	1	0	7TQ
1	1	1	8TQ (default value)

Remark: n = 0, 1

Figure 18-39: CAN Module n Bit Rate Register (CnBTR) (2/2)

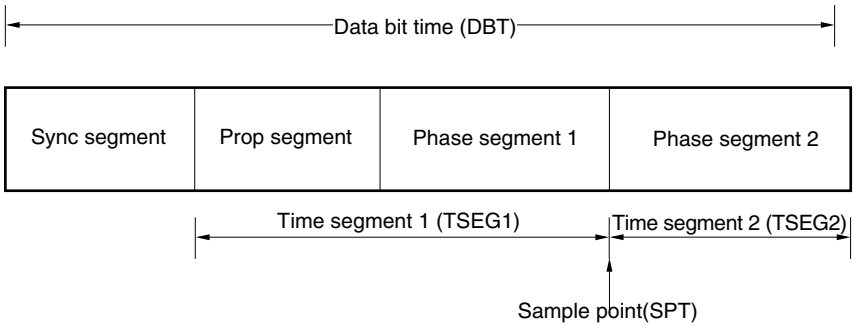
TSEG13	TSEG12	TSEG11	TSEG10	Length of Time Segment 1
0	0	0	0	Setting prohibited.
0	0	0	1	2TQ**Note**
0	0	1	0	3TQ**Note**
0	0	1	1	4TQ
0	1	0	0	5TQ
0	1	0	1	6TQ
0	1	1	0	7TQ
0	1	1	1	8TQ
1	0	0	0	9TQ
1	0	0	1	10TQ
1	0	1	0	11TQ
1	0	1	1	12TQ
1	1	0	0	13TQ
1	1	0	1	14TQ
1	1	1	0	15TQ
1	1	1	1	16TQ (default value)

Note: This setting must not be made when the CnBRP register = 00H.

Remarks: 1. TQ = $1/f_{TQ}$ (f_{TQ}: CAN protocol layer basic system clock)

2. n = 0, 1

(b) Definition of Data Bit Time



18.6.15 CAN module last in-pointer register (CnLIPT)

The CnLIPT register indicates the number of the message buffer in which a data frame or a remote frame was last stored.

Figure 18-40: CAN Module n Last In-Pointer Register (CnLIPT)

After reset: undefined R/W Offset address: 01EH

	7	6	5	4	3	2	1	0
CnLIPT (n = 0, 1)	LIPT7	LIPT6	LIPT5	LIPT4	LIPT3	LIPT2	LIPT1	LIPT0

LIPT7 to LIPT0	Last In-Pointer Register (CnLIPT)
0 to $(m_{MAX} - 1)$**Note**	When the CnLIPT register is read, the contents of the element indexed by the last in-pointer (LIPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame was last stored

Note: m_{MAX} = 32 (maximum number of message buffer)
Values greater than (m_{MAX} –1) are prohibited.

Remarks: 1. The read value of the CnLIPT register is undefined if a data frame or a remote frame has never been stored in the message buffer. If the RHPM bit of the CnRGPT register is set to 1 after the CAN module has changed from the initialization mode to an operation mode, therefore, the read value of the CnLIPT register is undefined.

2. n = 0, 1

18.6.16 CAN module receive history list register (CnRGPT)

The CnRGPT register is used to read the receive history list.

Figure 18-41: CAN Module n Receive History List Register (CnRGPT) (1/2)

(a) Read Operation

After reset: xx02H Read Offset address: 020H

CnRGPT (n = 0, 1)

15	14	13	12	11	10	9	8
RGPT7	RGPT6	RGPT5	RGPT4	RGPT3	RGPT2	RGPT1	RGPT0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	RHPM	ROVF

RGPT7 to RGPT0	Receive History List Get Pointer
0 to (m_{MAX} –1)Note 1	When the CnRGPT register is read, the contents of the element indexed by the receive history list get pointer (RGPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame has been stored.
Remark: The read value of RGPT0 to 7 is invalid when RHPM = 1.	

RHPM	Receive History List Pointer Match
0	The receive history list has at least one message buffer number that has not been read.
1	The receive history list has no message buffer numbers that has not been read.

ROVF	Receive History List Overflow Bit
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffer in which a new data frame or remote frame has been received and stored are recorded to the receive history list (the receive history list has a vacant element).
1	At least 23 entries have been stored since the host processor has serviced the RHL last time (i.e. read CnRGPT). Thus the sequence of receptions can not be recovered completely now. The first 22 entries are sequentially stored while the last entry can have been overwritten by newly received messages multiple times because all buffer numbers are stored at position LIPT-1 when ROVF bit is set. Note 2

Notes: 1. m_{MAX} = 32 (maximum number of message buffer)
Values greater than (m_{MAX} –1) are prohibited.

2. The receive history list will be updated, but the LIPT pointer will not be incremented. Always the position the LIPT pointer -1 is pointing to is overwritten.

Remark: n = 0, 1

Figure 18-41: CAN Module n Receive History List Register (CnRGPT) (2/2)

(b) Write Operation

After reset: – Write Offset address: 020H

CnRGPT (n = 0, 1)	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear ROVF

Clear ROVF	Setting of ROVF Bit
0	ROVF bit is not changed
1	ROVF bit is cleared to 0

Remark: n = 0, 1

18.6.17 CAN module last out-pointer register (CnLOPT)

The CnLOPT register indicates the number of the message buffer to which a data frame or a remote frame was transmitted last.

Figure 18-42: CAN Module n Last Out-Pointer Register (CnLOPT)

After reset: undefined Read Offset address: 022H

	7	6	5	4	3	2	1	0
CnLOPT (n = 0, 1)	LOPT7	LOPT6	LOPT5	LOPT4	LOPT3	LOPT2	LOPT1	LOPT0

LOPT7 to LOPT0	Last Out-Pointer of Transmit History List (LOPT)
0 to (m_{MAX} –1)**Note**	When the CnLOPT register is read, the contents of the element indexed by the last out-pointer (LOPT) of the receive history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

Note: m_{MAX} = 32

Remarks: 1. The value read from the CnLOPT register is undefined if a data frame or remote frame has never been transmitted from a message buffer. If the THPM bit is set to 1 after the CAN module has changed from the initialization mode to an operation mode, therefore, the read value of the CnLOPT register is undefined.

2. n = 0, 1

18.6.18 CAN module transmit history list register (CnTGPT)

The CnTGPT register is used to read the transmit history list.

Figure 18-43: CAN Module n Transmit History List Register (CnTGPT) (1/2)

(a) Read Operation

After reset: xx02H Read Offset address: 024H

	15	14	13	12	11	10	9	8
CnTGPT (n = 0, 1)	TGPT7	TGPT6	TGPT5	TGPT4	TGPT3	TGPT2	TGPT1	TGPT0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	THPM	TOVF

TGPT7 to TGPT0	Transmit History List Read Pointer
0 to $(m_{MAX} - 1)$ **Note 1**	When the CnTGPT register is read, the contents of the element indexed by the read pointer (TGPT) of the transmit history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.
Remark: The read value of TGPT0 to TGPT7 is invalid when THPM = 1.	

THPM	Transmit History Pointer Match
0	The transmit history list has at least one message buffer number that has not been read.
1	The transmit history list has no message buffer number that has not been read.

TOVF	Transmit History List Overflow Bit
0	All the message buffer numbers that have not been read are preserved.All the numbers of the message buffers to which a new data frame or remote frame has been transmitted are recorded to the transmit history list (the transmit history list has a vacant element)
1	At least 7 entries have been stored since the host processor has serviced the THL last time (i.e. read CnTGPT). Thus the sequence of transmissions can not be recovered completely now. The first 6 entries are sequentially stored while the last entry can have been overwritten by newly transmitted messages multiple times because all buffer numbers are stored at position LOPT-1 when TOVF bit is set. **Note 3**

Notes: 1. $m_{MAX} = 32$

2. The THL will be updated, but the LOPT pointer will not be incremented. Always the position the LOPT pointer -1 is pointing to is overwritten.

Remarks: 1. Transmission from message buffer 0 to 7 is not recorded to the transmit history list in the normal operation mode with ABT.

2. n = 0, 1

Figure 18-43: CAN Module n Transmit History List Register (CnTGPT) (2/2)

(b) Write Operation

After reset: – Write Offset address: 024H

CnTGPT (n = 0, 1)	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear TOVF

Clear TOVF	Setting of TOVF Bit
0	TOVF bit is not changed
1	TOVF bit is cleared to 0

Remark: n = 0, 1

18.6.19 CAN module time stamp register (CnTS)

The CnTS register is used to control the time stamp function.

Figure 18-44: CAN Module Time Stamp Register (CnTS) (1/2)

(a) Read Operation

After reset: 0000H Read Offset address: 026H

	15	14	13	12	11	10	9	8
CnTS (n = 0, 1)	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	TSLOCK	TSSEL	TSEN

TSLOCK	Time Stamp Lock Function Enable Bit
0	Time stamp lock function stopped. The TSOUT signal is toggled each time the selected time stamp capture event occurs.
1	Time stamp lock function enabled. The TSOUT signal is toggled each time the selected time stamp capture event occurs. However, the TSOUT output signal is locked when a data frame has been correctly received to message buffer 0.
• When TSLOCK is set (1) and a data frame has been correctly received to message buffer 0, the TSOUT output signal is locked. • When the TSOUT output signal is locked the TSEN bit is cleared (0) automatically.	

TSSEL	Time Stamp Capture Event Selection Bit
0	The time capture event is SOF
1	The time stamp capture event is the last bit of EOF

TSEN	TSOUT Operation Setting Bit
0	Disable TSOUT toggle operation
1	Enable TSOUT toggle operation
When the TSEN bit is set (1) the TSOUT signal is toggled each time the selected time stamp capture event occurs.	

Remarks: 1. The displayed CnTS register provides only the necessary bits for the Basic Time Stamp function.
The Advanced Time Stamp function requires a modified hardware.

2. The lock function of the time stamp function must not be used when the CAN module is in the normal operation mode with ABT

3. n = 0, 1

Figure 18-44: CAN Module Time Stamp Register (CnTS) (2/2)

(b) Write Operation

After reset: – Write Offset address: 026H

	15	14	13	12	11	10	9	8
CnTS (n = 0, 1)	0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
	7	6	5	4	3	2	1	0
	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN

Set TSLOCK	Clear TSLOCK	Setting of TSLOCK Bit
0	1	TSLOCK bit is cleared to 0
1	0	TSLOCK bit is set to 1
Other than above		TSLOCK bit is not changed

Set TSSEL	Clear TSSEL	Setting of TSSEL Bit
0	1	TSSEL bit is cleared to 0
1	0	TSSEL bit is set to 1
Other than above		TSSEL bit is not changed

Set TSEN	Clear TSEN	Setting of TSEN Bit
0	1	TSEN bit is cleared to 0
1	0	TSEN bit is set to 1
Other than above		TSEN bit is not changed

Remark: n = 0, 1

18.6.20 CAN message data byte register (CnMDATAxm) (x = 0 to 7), (CnMDATAzm) (z = 01, 23, 45, 67)

The CnMDATAxm, CnMDATAzm registers are used to store the data of a transmit/receive message. The CnMDATAzm registers can access the CnMDATAxm registers in 16-bit units.

Figure 18-45: CAN Message Data Byte Register (1/2)

(a) (CnMDATAxm) (x = 0 to 7)

After reset: undefined R/W Offset address: 000H

	7	6	5	4	3	2	1	0
CnMDATA0m (n = 0, 1)	MDATA07	MDATA06	MDATA05	MDATA04	MDATA03	MDATA02	MDATA01	MDATA00

After reset: undefined R/W Offset address: 001H

	7	6	5	4	3	2	1	0
CnMDATA1m (n = 0, 1)	MDATA17	MDATA16	MDATA15	MDATA14	MDATA13	MDATA12	MDATA11	MDATA10

After reset: undefined R/W Offset address: 002H

	7	6	5	4	3	2	1	0
CnMDATA2m (n = 0, 1)	MDATA27	MDATA26	MDATA25	MDATA24	MDATA23	MDATA22	MDATA21	MDATA20

After reset: undefined R/W Offset address: 003H

	7	6	5	4	3	2	1	0
CnMDATA3m (n = 0, 1)	MDATA37	MDATA36	MDATA35	MDATA34	MDATA33	MDATA32	MDATA31	MDATA30

After reset: undefined R/W Offset address: 004H

	7	6	5	4	3	2	1	0
CnMDATA4m (n = 0, 1)	MDATA47	MDATA46	MDATA45	MDATA44	MDATA43	MDATA42	MDATA41	MDATA40

After reset: undefined R/W Offset address: 005H

	7	6	5	4	3	2	1	0
CnMDATA5m (n = 0, 1)	MDATA57	MDATA56	MDATA55	MDATA54	MDATA53	MDATA52	MDATA51	MDATA50

After reset: undefined R/W Offset address: 006H

	7	6	5	4	3	2	1	0
CnMDATA6m (n = 0, 1)	MDATA67	MDATA66	MDATA65	MDATA64	MDATA63	MDATA62	MDATA61	MDATA60

After reset: undefined R/W Offset address: 007H

	7	6	5	4	3	2	1	0
CnMDATA7m (n = 0, 1)	MDATA77	MDATA76	MDATA75	MDATA74	MDATA73	MDATA72	MDATA71	MDATA70

Remark: n = 0, 1; m = 0 to 31

Figure 18-45: CAN Message Data Byte Register (2/2)

(b) (CnMDATAzm) (z = 01, 23, 45, 67)

After reset: undefined R/W Offset address: 000H

CnMDATA01m (n = 0, 1)

15	14	13	12	11	10	9	8
MDATA0115	MDATA0114	MDATA0113	MDATA0112	MDATA0111	MDATA0110	MDATA019	MDATA018

7	6	5	4	3	2	1	0
MDATA017	MDATA016	MDATA015	MDATA014	MDATA013	MDATA012	MDATA011	MDATA010

After reset: undefined R/W Offset address: 002H

CnMDATA23m (n = 0, 1)

15	14	13	12	11	10	9	8
MDATA2315	MDATA2314	MDATA2313	MDATA2312	MDATA2311	MDATA2310	MDATA239	MDATA238

7	6	5	4	3	2	1	0
MDATA237	MDATA236	MDATA235	MDATA234	MDATA233	MDATA232	MDATA231	MDATA230

After reset: undefined R/W Offset address: 004H

CnMDATA45m (n = 0, 1)

15	14	13	12	11	10	9	8
MDATA4515	MDATA4514	MDATA4513	MDATA4512	MDATA4511	MDATA4510	MDATA459	MDATA458

7	6	5	4	3	2	1	0
MDATA457	MDATA456	MDATA455	MDATA454	MDATA453	MDATA452	MDATA451	MDATA450

After reset: undefined R/W Offset address: 006H

CnMDATA67m (n = 0, 1)

15	14	13	12	11	10	9	8
MDATA6715	MDATA6714	MDATA6713	MDATA6712	MDATA6711	MDATA6710	MDATA679	MDATA678

7	6	5	4	3	2	1	0
MDATA677	MDATA676	MDATA675	MDATA674	MDATA673	MDATA672	MDATA671	MDATA670

Remark: n = 0, 1
m = 0 to 31

18.6.21 CAN message data length register m (CnMDLCm)

The CnMDLCm register is used to set the number of bytes of the data field of a message buffer.

Cautions: 1. Be sure to set bits 7 to 4 to 0000B.

2. Receive data is stored in as many CnMDATAx as the number of bytes (however, the upper limit is 8) corresponding to DLC. CnMDATAx in which no data is stored is undefined.

Figure 18-46: CAN Message Data Length Register m (CnMDLCm)

After reset: undefined R/W Offset address: 008H

	7	6	5	4	3	2	1	0
CnMDLCm (n = 0, 1)	0	0	0	0	MDLC3	MDLC2	MDLC1^	MDLC0

MDLC3	MDLC2	MDLC1	MDLC0	Data Length Of Transmit/Receive Message
0	0	0	0	0 byte
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than above				Setting prohibited (If these bits are set during transmission, 8-byte data is transmitted regardless of the set DLC value when a data frame is transmitted. However, the DLC actually transmitted to the CAN bus is the DLC value set to this register.)**Note**

Note: The data and DLC value actually transmitted to CAN bus are as follows:

Type of Transmit Frame	Length of Transmit Data	DLC Transmitted
Data frame	Number of bytes specified by DLC (however, 8 bytes if DLC ≥ 8)	MDLC[3:0]
Remote frame	0 bytes	

Remark: n = 0, 1
m = 0 to 31

18.6.22 CAN message configuration register (CnMCONFm)

The CnMCONFm register is used to specify the type of the message buffer and to set a mask.

Caution: Be sure to write 0 to bits 2 and 1 of the CnCONFm register.

Figure 18-47: CAN Message Configuration Register (CnMCONFm)

After reset: undefined R/W Offset address: 009H

	7	6	5	4	3	2	1	0
CnMCONFm (n = 0, 1)	OWS	RTR	MT2	MT1	MT0	0	0	MA0

OWS	Overwrite Select Bit
0	The message buffer that has already received a data frame[Note] is not overwritten by a newly received data frame. The newly received data frame is discarded
1	The message buffer that has already received a data frame[Note] is overwritten by a newly received data frame
A remote frame is received and stored, regardless of the setting of bits OWS and DN. A remote frame that satisfies the other conditions (ID matches, RTR = 0, TRQ = 0) is always received and stored in the corresponding message buffer (interrupt generated, DN flag set, MDLC0 to MDLC3 bits updated, and recorded to the receive history list). **Note:** The “message buffer that has already received a data frame” is a receive message buffer whose DN bit has been set to 1.	

RTR	Remote Frame Request Bit
0	Transmit a data frame
1	Transmit a remote frame
The RTR bit specifies the type of message frame that is transmitted from a message buffer defined as a transmit message buffer.Even if a valid remote frame has been received, RTR of the transmit message buffer that has received the frame remains cleared to 0.Even if a remote frame whose ID matches has been received from the CAN bus with the RTR bit of the transmit message buffer set to 1 to transmit a remote frame, that remote frame is not received or stored (interrupt generated, DN flag set, MDLC0 to MDLC3 bits updated, and recorded to the receive history list).	

Remark: n = 0, 1

Figure 18-47: CAN Message Configuration Register (CnMCONFm) (2/2)

MT2	MT1	MT0	Message Buffer Type Setting Bit
0	0	0	Transmit message buffer
0	0	1	Receive message buffer (no mask setting)
0	1	0	Receive message buffer (mask 1 set)
0	1	1	Receive message buffer (mask 2 set)
1	0	0	Receive message buffer (mask 3 set)
1	0	1	Receive message buffer (mask 4 set)
Other than above			Setting prohibited.

MA0	Message Buffer Assignment Bit
0	Message buffer not used
1	Message buffer used

Remark: n = 0, 1

18.6.23 CAN message ID register m (CnMIDLm, CnMIDHm)

The CnMIDLm and CnMIDHm registers are used to set an identifier (ID).

Caution: Be sure to write 0 to bits 14 and 13 of the CnMIDHm register.

Figure 18-48: CAN Message ID Register m (CnMIDLm, CnMIDHm)

After reset: undefined R/W Offset address: 00AH

	15	14	13	12	11	10	9	8
CnMIDLm (n = 0, 1)	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
	7	6	5	4	3	2	1	0
	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

After reset: undefined R/W Offset address: 00CH

	15	14	13	12	11	10	9	8
CnMIDHm (n = 0, 1)	IDE	0	0	ID28	ID27	ID26	ID25	ID24
	7	6	5	4	3	2	1	0
	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

IDE	Format Mode Specification Bit
0	Standard format mode (ID28 to ID18: 11 bits)[Note]
1	Extended format mode (ID28 to ID0: 29 bits)

Note: The ID17 to ID0 bits are not used.

ID28 to ID0	Message ID
ID28 to ID18	Standard ID value of 11 bits (when IDE = 0)
ID28 to ID0	Extended ID value of 29 bits (when IDE = 1)

Remark: n = 0, 1
m = 0 to 31

18.6.24 CAN message control register m (CnMCTRLm)

The CnMCTRLm register is used to control the operation of the message buffer.

Figure 18-49: CAN Message Control Register m (CnMCTRLm) (1/3)

(a) Read Operation

After reset: undefined Read Offset address: 00EH

	15	14	13	12	11	10	9	8
CnMCTRLm (n = 0, 1)	0	0	MUC	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	MOW	IE	DN	TRQ	RDY

MUC	Message Buffer Data Updating Bit
0	The CAN module is not updating the message buffer (reception and storage)
1	The CAN module is updating the message buffer (reception and storage)
The MUC bit is undefined until the first reception and storage is performed	

MOW	Message Buffer Overwrite Status Bit
0	The message buffer is not overwritten by a newly received data frame
1	The message buffer is overwritten by a newly received data frame
MOW is not set to 1 even if a remote frame is received and stored in the transmit message buffer with DN bit = 1.	

IE	Message Buffer Interrupt Request Enable Bit
0	Receive message buffer: Valid message reception completion interrupt disabled. Transmit message buffer: Normal message transmission completion interrupt disabled.
1	Receive message buffer: Valid message reception completion interrupt enabled. Transmit message buffer: Normal message transmission completion interrupt enabled.

DN	Message Buffer Data Updating Bit
0	A data frame or remote frame is not stored in the message buffer
1	A data frame or remote frame is stored in the message buffer

Remark: n = 0, 1
m = 0 to 31

Figure 18-49: CAN Message Control Register m (CnMCTRLm) (2/3)

TRQ	Message Buffer Transmission Request Bit
0	No message frame transmitting request that is pending or being transmitted is in the message buffer.
1	The message buffer is holding transmission of a message frame pending or is transmitting a message frame.

RDY	Message Buffer Ready Bit
0	The message buffer can be written by software. The CAN module cannot write to the message buffer
1	Writing the message buffer by software is ignored (except a write access to the RDY, TRQ, DN, and MOW bits). The CAN module can write to the message buffer
Caution:	**Do not clear the RDY bit (0) during message transmission. Follow the transmission abort process about clearing the RDY bit (0) for redefinition of the message buffer.**

(b) Write Operation

After reset: – Write Offset address: 00EH

CnMCTRLm (n = 0, 1)	15	14	13	12	11	10	9	8
	0	0	0	0	Set IE	0	Set TRQ	Set RDY
	7	6	5	4	3	2	1	0
	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY

Clear MOW	MOW Bit Setting
0	MOW bit is not changed
1	MOW bit is cleared to 0

Set IE	Clear IE	Setting of IE Bit
0	1	IE bit is cleared to 0
1	0	IE bit is set to 1
Other than above		IE bit is not changed

Clear DN	Setting of DN Bit
1	DN bit is cleared to 0
0	DN bit is not changed
Caution:	**Do not set the DN bit to 1 by software. Be sure to write 0 to bit 10**

Remark: n = 0, 1
m = 0 to 31

Figure 18-49: CANn Message Control Register m (CnMCTRLm) Format (3/3)

Set TRQ	Clear TRQ	Setting of TRQ Bit
0	1	TRQ bit is cleared to 0
1	0	TRQ bit is set to 1
Other than above		TRQ bit is not changed

Set RDY	Clear RDY	Setting of RDY Bit
0	1	RDY bit is cleared to 0
1	0	RDY bit is set to 1
Other than above		RDY bit is not changed

Remark: n = 0, 1
m = 0 to 31

18.7 Bit Set/Clear Function

The CAN control registers include registers whose bits can be set or cleared via the CPU and via the CAN interface. An operation error occurs if the following registers are written directly. Do not write any values directly via bit manipulation, read/modify/write, or direct writing of target values.

- CAN global control register (CnGMCTRL)
- CAN global automatic block transmission control register (CnGMABT)
- CAN module control register (CnCTRL)
- CAN module interrupt enable register (CnIE)
- CAN module interrupt status register (CnINTS)
- CAN module receive history list register (CnRGPT)
- CAN module transmit history list register (CnTGPT)
- CAN module time stamp register (CnTS)
- CAN message control register (CnMCTRLm)

Remark: n: 0 - 4 = Number of channel
m: 0 - 31 = message buffer number

All the 16 bits in the above registers can be read via the usual method. Use the procedure described in Figure 18-50 below to set or clear the lower 8 bits in these registers.
Setting or clearing of lower 8 bits in the above registers is performed in combination with the higher 8 bits (refer to the 16-bit data after a write operation in Figure 18-51). Figure 18-50 shows how the values of set bits or clear bits relate to set/clear/no change operations in the corresponding register.

Figure 18-50: Example of Bit Setting/Clearing Operations

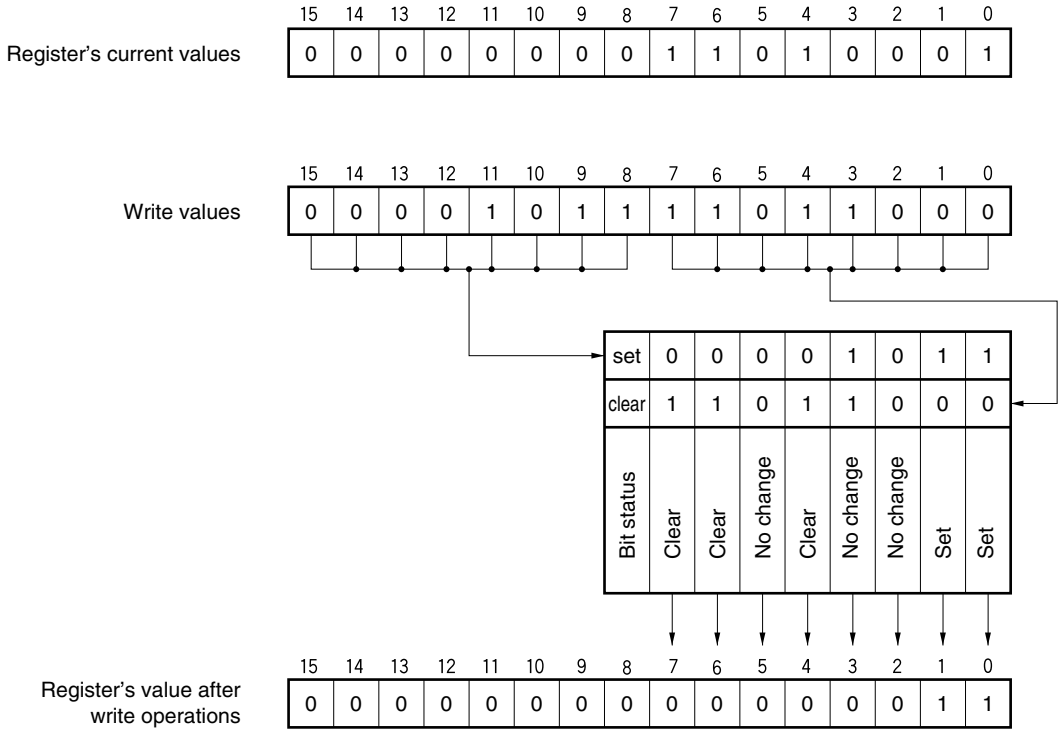


Figure 18-51: 16-Bit Data during Write Operation

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

Set n	Clear n	Status of bit n after bit set/clear operation
0	0	No change
0	1	0
1	0	1
1	1	No change

Remark: n = 0 to 7

18.8 CAN Controller Initialization

18.8.1 Initialization of CAN module

Before the CAN module operation is enabled, the CAN module system clock needs to be determined by setting the CCP0 to CCP3 bits of the CnGMCS register by software. Do not change the setting of the CAN module system clock after CAN module operation is enabled.
The CAN module is enabled by setting the GOM bit of the CnGMCTRL register.
For the procedure of initializing the CAN module, refer to **18.16 "Operation of CAN Controller" on page 837**.

18.8.2 Initialization of message buffer

After the CAN module is enabled, the message buffers contain undefined values. A minimum initialization for all the message buffers, even for those not used in the application, is necessary before switching the CAN module from the initialization mode to one of the operation modes.

- Clear the RDY, TRQ, and DN bits of the CnMCTRLm register to 0.
- Clear the MA0 bit of the CnMCONFm register to 0.

18.8.3 Redefinition of message buffer

Redefining a message buffer means changing the ID and control information of the message buffer while a message is being received or transmitted, without affecting other transmission/reception operations.

(1) To redefine message buffer in initialization mode

Place the CAN module in the initialization mode once and then change the ID and control information of the message buffer in the initialization mode. After changing the ID and control information, set the CAN module in an operation mode.

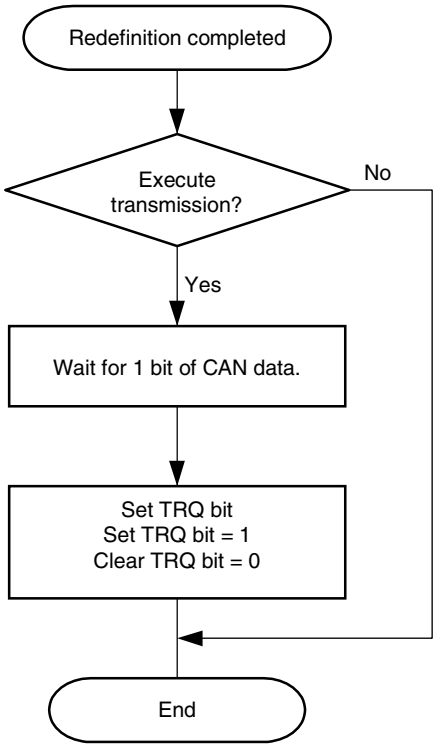
(2) To redefine message buffer during reception

Perform redefinition as shown in **Figure 18-64, "Message Buffer Redefinition," on page 840**.

(3) To redefine message buffer during transmission

To rewrite the contents of a transmit message buffer to which a transmission request has been set, perform transmission abort processing (refer to **18.10.4 (1)"Transmission abort in normal operation mode" on page 818** and **18.10.4 (3)"Transmission abort in normal operation mode with automatic block transmission (ABT)" on page 818**). Confirm that transmission has been aborted or completed, and then redefine the message buffer. After redefining the transmit message buffer, set a transmission request using the procedure described below. When setting a transmission request to a message buffer that has been redefined without aborting the transmission in progress, however, the 1-bit wait time is not necessary.

Figure 18-52: Setting Transmission Request (TRQ) to Transmit Message Buffer after Redefining



Cautions: 1. When a message is received, reception filtering is performed in accordance with the ID and mask set to each receive message buffer. If the procedure in Figure Figure 18-64, "Message Buffer Redefinition," on page 840 is not observed, the contents of the message buffer after it has been redefined may contradict the result of reception (result of reception filtering). If this happens, check that the ID and IDE received first and stored in the message buffer following redefinition are those stored after the message buffer has been redefined. If no ID and IDE are stored after redefinition, redefine the message buffer again.

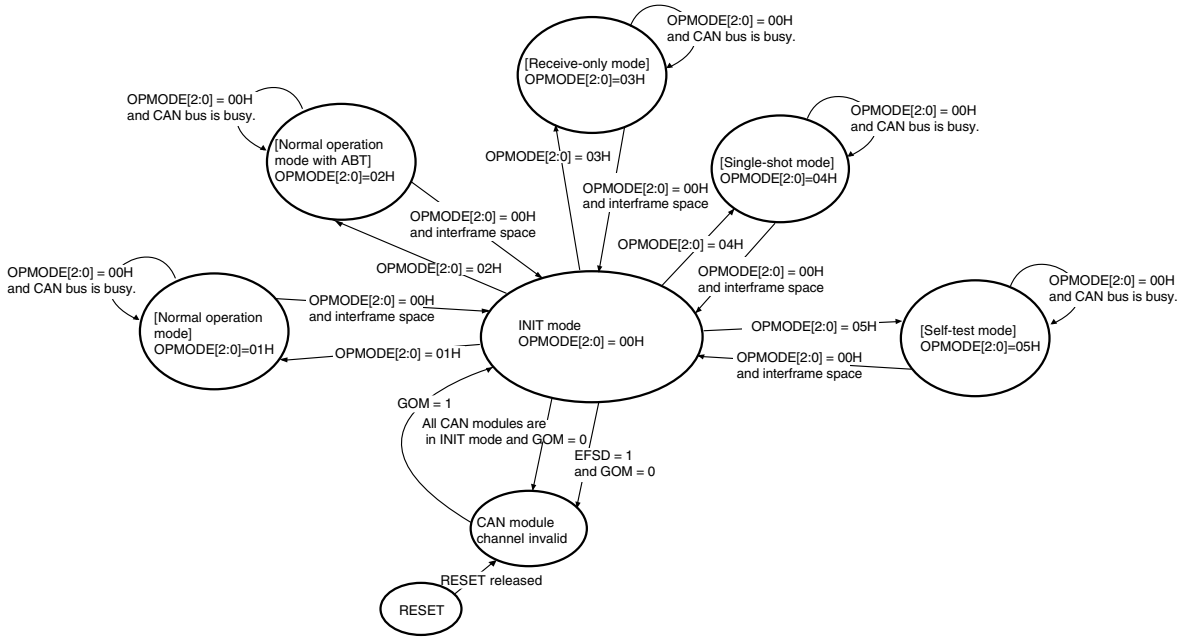
2. When a message is transmitted, the transmission priority is checked in accordance with the ID, IDE, and RTR bits set to each transmit message buffer to which a transmission request was set. The transmit message buffer having the highest priority is selected for transmission. If the procedure in Figure 18-52 is not observed, a message with an ID not having the highest priority may be transmitted after redefinition.

18.8.4 Transition from Initialization mode to Operation mode

The CAN module can be switched to the following operation modes:

- Normal operation mode
- Normal operation mode with ABT
- Receive-only mode
- Single-shot mode
- Self-test mode

Figure 18-53: Transition to Operation Modes



The transition from the initialization mode to an operation mode is controlled by the bit string OPMODE2 to OPMODE0 in the CnCTRL register.
Changing from one operation mode into another requires shifting to the initialization mode in between. Do not change one operation mode to another directly; otherwise the operation will not be guaranteed.
Requests for transition from the operation mode to the initialization mode are held pending when the CAN bus is not in the interframe space (i.e., frame reception or transmission is in progress), and the CAN module enters the initialization mode at the first bit in the interframe space (the value of OPMOD2 to OPMODE0 bits are changed to 00H). After issuing a request to change the mode to the initialization mode, read the OPMODE2 to OIPMODE0 bits until their value becomes 000B to confirm that the module has entered the initialization mode (refer to **Figure 18-62, “Re-initialization,” on page 838**).

18.8.5 Resetting error counter CnERC of CAN module

If it is necessary to reset the CAN module error counter CnERC and the CAN module information register CnINFO when re-initialization or forced recovery from the bus-off state is made, set the CCERC bit of the CnCTRL register to 1 in the initialization mode. When this bit is set to 1, the CAN module error counter CnERC and the CAN module information register CnINFO are cleared to their default values.

18.9 Message Reception

18.9.1 Message reception

In all the operation modes, the complete message buffer area is analyzed to find a suitable buffer to store a newly received message. All message buffers satisfying the following conditions are included in that evaluation (RX-search process).

- Used as a message buffer
  (MA0 bit of CnMCONFm register set to 1B.)
- Set as a receive message buffer
  (MT2 to MT0 bits of CnMCONFm register set to 001B, 010B, 011B, 100B, or 101B.)
- Ready for reception
  (RDY bit of CnMCTRLm register set to 1.)

When two or more message buffers of the CAN module receive a message, the message is stored according to the priority explained below. The message is always stored in the message buffer with the highest priority, not in a message buffer with a low priority. For example, when an unmasked receive message buffer and a receive message buffer linked to mask 1 have the same ID, the received message is not stored in the message buffer linked to mask 1, even if that message buffer has not received a message and a message has already been received in the unmasked receive message buffer. In other words, when a condition has been set to store a message in two or more message buffers with different priorities, the message buffer with the highest priority always stores the message; the message is not stored in message buffers with a lower priority. This also applies when the message buffer with the highest priority is unable to receive and store a message (i.e., when DN = 1 indicating that a message has already been received, but rewriting is disabled because OWS = 0). In this case, the message is not actually received and stored in the candidate message buffer with the highest priority, but neither is it stored in a message buffer with a lower priority.

Table 18-25: Message Reception

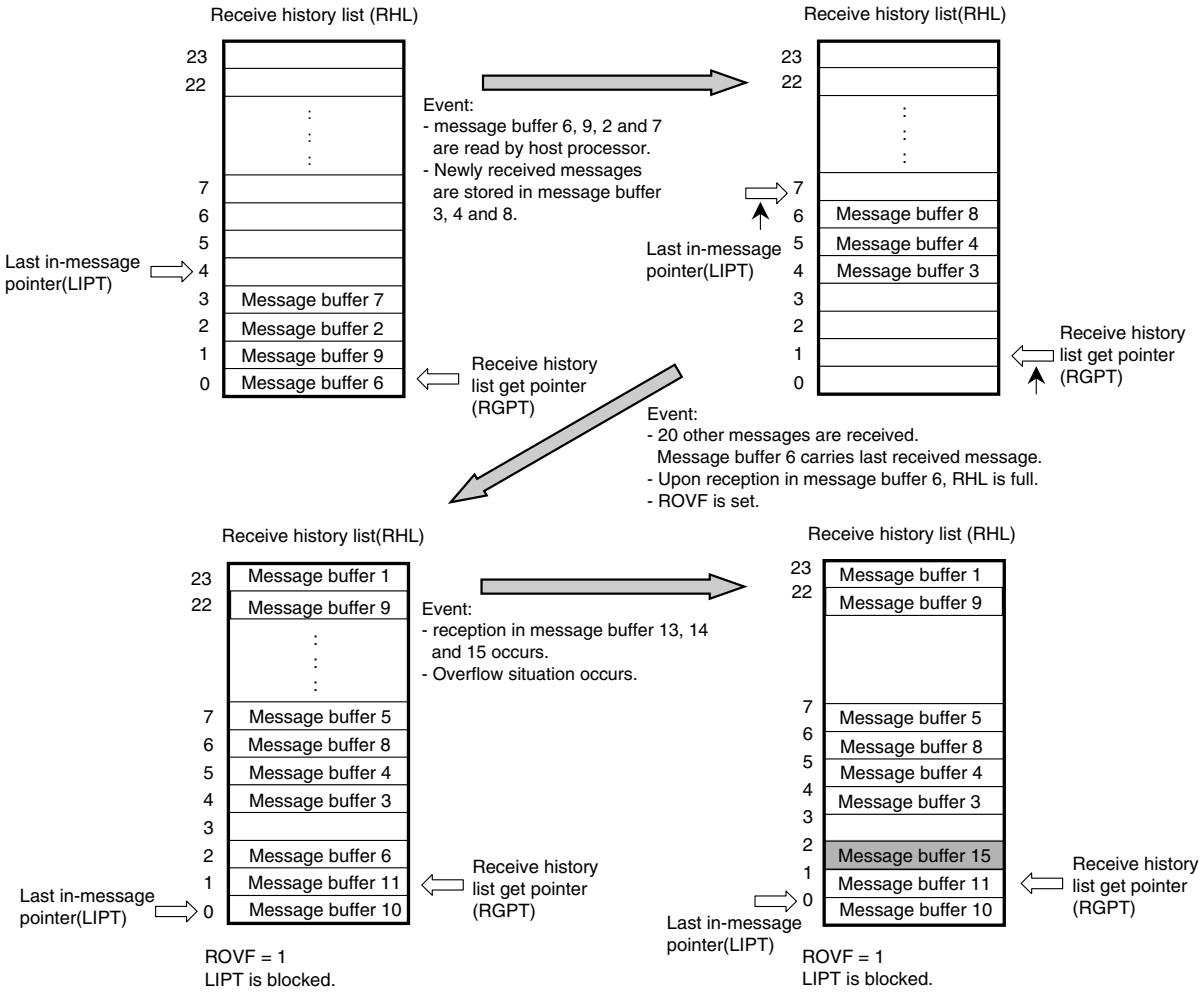
Priority	Storing Condition If Same ID is Set	
1 (high)	Unmasked message buffer	DN = 0
		DN = 1 and OWS = 1
2	Message buffer linked to mask 1	DN = 0
		DN = 1 and OWS = 1
3	Message buffer linked to mask 2	DN = 0
		DN = 1 and OWS = 1
4	Message buffer linked to mask 3	DN = 0
		DN = 1 and OWS = 1
5 (low)	Message buffer linked to mask 4	DN = 0
		DN = 1 and OWS = 1

18.9.2 Receive history list function

The receive history list (RHL) function records in the receive history list the number of the receive message buffer in which each data frame or remote frame was received and stored. The RHL consists of storage elements equivalent to up to 23 messages, the last in-message pointer (LIPT) with the corresponding CnLIPT register and the receive history list get pointer (RGPT) with the corresponding CnRGPT register.
The RHL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.
The CnLIPT register holds the contents of the RHL element indicated by the value of the LIPT pointer minus 1. By reading the CnLIPT register, therefore, the number of the message buffer that received and stored a data frame or remote frame first can be checked. The LIPT pointer is utilized as a write pointer that indicates to what part of the RHL a message buffer number is recorded. Any time a data frame or remote frame is received and stored, the corresponding message buffer number is recorded to the RHL element indicated by the LIPT pointer. Each time recording to the RHL has been completed, the LIPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.
The RGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the RHL. This pointer indicates the first RHL element that the CPU has not read yet. By reading the CnRGPT register by software, the number of a message buffer that has received and stored a data frame or remote frame can be read. Each time a message buffer number is read from the CnRGPT register, the RGPT pointer is automatically incremented.
If the value of the RGPT pointer matches the value of the LIPT pointer, the RHPM bit (receive history list pointer match) of the CnRGPT register is set to 1. This indicates that no message buffer number that has not been read remains in the RHL. If a new message buffer number is recorded, the LIPT pointer is incremented and because its value no longer matches the value of the RGPT pointer, the RHPM bit is cleared. In other words, the numbers of the unread message buffers exist in the RHL.
If the LIPT pointer is incremented and matches the value of the RGPT pointer minus 1, the ROVF bit (receive history list overflow) of the CnRGPT register is set to 1. This indicates that the RHL is full of numbers of message buffers that have not been read. When further message reception and storing occur, the last recorded message buffer number is overwritten by the number of the message buffer that received and stored the new message. In this case, after the ROVF bit has been set (1), the recorded message buffer numbers in the RHL do not completely reflect the chronological order.
However the non-recovered receptions in the RHL are still recoverable. Therefore the application needs to browse all RX-buffer and check the DN bits.

As long as the RHL contains 23 or less entries the sequence of occurrence is maintained. If more receptions occur without reading the RHL by the host processor, complete sequence of receptions can not be recovered.

Figure 18-54: Receive History List



ROVF = 1 defines that LIPT equals RGPT - 1 while message buffer number stored to element indicated by LIPT - 1.

18.9.3 Mask function

It can be defined whether masking of the identifier that is set to a message buffer is linked with another message buffer.
By using the mask function, the identifier of a message received from the CAN bus can be compared with the identifier set to a message buffer in advance. Regardless of whether the masked ID is set to "0" or "1", the received message can be stored in the defined message buffer.
While the mask function is in effect, an identifier bit that is defined to be "1" by a mask in the received message is not compared with the corresponding identifier bit in the message buffer.
However, this comparison is performed for any bit whose value is defined as "0" by the mask.
For example, let us assume that all messages that have a standard-format ID, in which bits ID27 to ID25 are "0" and bits ID24 and ID22 are "1", are to be stored in message buffer 14. The procedure for this example is shown below.

Figure 18-55: Mask Function Identifier Examples (1/2)

(a) Identifier to be stored in message buffer

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

Remark: x = don't care

(b) Identifier to be configured in message buffer 14 (example)
(using CANn message ID registers L14 and H14 (CnMIDL14 and CnMIDH14))

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x
ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
x	x	x	x	x	x	x	x	x	x	x
ID6	ID5	ID4	ID3	ID2	ID1	ID0				
x	x	x	x	x	x	x				

ID with ID27 to ID25 cleared to "0" and ID24 and ID22 set to "1" is registered (initialized) to message buffer 14.

Remark: Message buffer 14 is set as a standard format identifier that is linked to mask 1 (MT2 to MT0 bits of CnMCONF14 register are set to 010B).

Figure 18-55: Mask Function Identifier Examples (2/2)

(c) Mask setting for CAN module 1 (mask 1) (Example)
(Using CAN1 address mask 1 registers L and H (C1MASKL1 and C1MASKH1))

CMID28	CMID27	CMID26	CMID25	CMID24	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18
1	0	0	0	0	1	0	1	1	1	1
CMID17	CMID16	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	CMID7
1	1	1	1	1	1	1	1	1	1	1
CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0				
1	1	1	1	1	1	1				

Remark: 1: Not compared (masked)
0: Compared

The CMID27 to CMID24 and CMID22 bits are cleared to "0", and CMID28, CMID23, and CMID21 to CMID0 bits are set to "1".

18.9.4 Multi buffer receive block function

The multi buffer receive block (MBRB) function is used to store a block of data in two or more message buffers sequentially with no CPU interaction, by setting the same ID to two or more message buffers with the same message buffer type. These message buffers can be allocated anywhere in the message buffer memory, they do not even have to follow each other adjacently.
Suppose, for example, the same message buffer type is set to 10 message buffers, message buffers 10 to 19, and the same ID is set to each message buffer. If the first message whose ID matches the ID of the message buffers is received, it is stored in message buffer 10. At this point, the DN bit of message buffer 10 is set, prohibiting overwriting the message buffer when subsequent messages are received.
If the next message with a matching ID is received, it is received and stored in message buffer 11. Each time a message with a matching ID is received, it is sequentially (in the ascending order) stored in message buffers 12, 13, and so on. Even when a data block consisting of multiple messages is received, the messages can be stored and received without overwriting the previously received matching-ID data.
Whether a data block has been received and stored can be checked by setting the IE bit of the CnMCTRLm register of each message buffer. For example, if a data block consists of k messages, k message buffers are initialized for reception of the data block. The IE bit in message buffers 0 to (k-2) is cleared to 0 (interrupts disabled), and the IE bit in message buffer k-1 is set to 1 (interrupts enabled). In this case, a reception completion interrupt occurs when a message has been received and stored in message buffer k-1, indicating that MBRB has become full. Alternatively, by clearing the IE bit of message buffers 0 to (k-3) and setting the IE bit of message buffer k-2, a warning that MBRB is about to overflow can be issued.
The basic conditions of storing receive data in each message buffer for the MBRB are the same as the conditions of storing data in a single message buffer.

Cautions:

1. **MBRB can be configured for each of the same message buffer types. Therefore, even if a message buffer of another MBRB whose ID matches but whose message buffer type is different has a vacancy, the received message is not stored in that message buffer, but instead discarded.**
2. **MBRB does not have a ring buffer structure. Therefore, after a message is stored in the message buffer having the highest number in the MBRB configuration, a newly received message will not be stored in the message buffer having the lowest message buffer number.**
3. **MBRB operates based on the reception and storage conditions; there are no settings dedicated to MBRB, such as function enable bits. By setting the same message buffer type and ID to two or more message buffers, MBRB is automatically configured.**
4. **With MBRB, "matching ID" means "matching ID after mask". Even if the ID set to each message buffer is not the same, if the ID that is masked by the mask register matches, it is considered a matching ID and the buffer that has this ID is treated as the storage destination of a message.**
5. **The priority between MBRBs is mentioned in the Table 18-25, “Message Reception,” on page 806.**

18.9.5 Remote frame reception

In all the operation modes, when a remote frame is received, the message buffer that is to store the remote frame is searched from all the message buffers satisfying the following conditions.

- Used as a message buffer
  (MA0 bit of CnMCONFm register set to 1B.)
- Set as a transmit message buffer
  (MT2 to MT0 bits in CnMCONFm register set to 000B)
- Ready for reception
  (RDY bit of CnMCTRLm register set to 1.)
- Set to transmit message
  (RTR bit of CnMCONFm register is cleared to 0.)
- Transmission request is not set.
  (TRQ bit of CnMCTRLm register is cleared to 1.)

Upon acceptance of a remote frame, the following actions are executed if the ID of the received remote frame matches the ID of a message buffer that satisfies the above conditions.

- The DLC3 to DLC0 bit string in the CnMDLCm register stores the received DLC value.
- CnMDATA0m to CnMDATA7m in the data area are not updated (data before reception is saved).
- The DN bit of the CnMCTRLm register is set to 1.
- The CINTS1 bit of the CnINTS register is set to 1 (if the IE bit in the CnMCTRLm register of the message buffer that receives and stores the frame is set to 1).
- The reception completion interrupt (INTCnREC) is output (if the IE bit in the CnMCTRLm register of the message buffer that receives and stores the frame is set to 1 and if the CIE1 bit of the CnIE register is set to 1).
- The message buffer number is recorded to the receive history list.

Caution: When a message buffer is searched for receiving and storing a remote frame, overwrite control by the OWS bit of the CnMCONFm register of the message buffer and the DN bit of the CnMCTRLm register are not affected.
If more than one transmit message buffer has the same ID and the ID of the received remote frame matches that ID, the remote frame is stored in the transmit message buffer with the lowest message buffer number.

18.10 Message Transmission

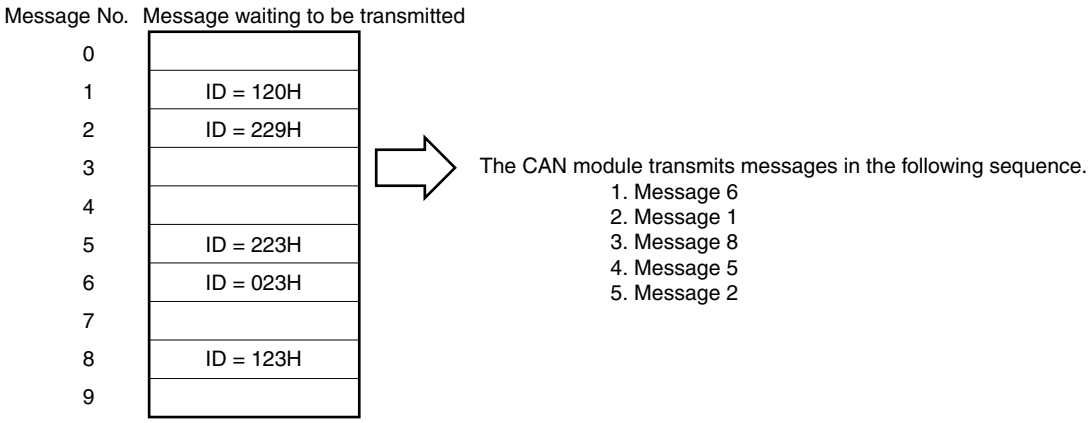
18.10.1 Message transmission

A message buffer with its TRQ bit set to 1 participates in the search for the most high-prioritized message when the following conditions are fulfilled. This behaviour is valid for all operational modes.

- Used as a message buffer
  (MA0 bit of CnMCONFm register set to 1B.)
- Set as a transmit message buffer
  (MT2 to MT0 bits of CnMCONFm register set to 000B.)
- Ready for transmission
  (RDY bit of CnMCTRLm register set to 1.)

The CAN system is a multi-master communication system. In a system like this, the priority of message transmission is determined based on message identifiers (IDs). To facilitate transmission processing by software when there are several messages awaiting transmission, the CAN module uses hardware to check the ID of the message with the highest priority and automatically identifies that message. This eliminates the need for software-based priority control.
Transmission priority is controlled by the identifier (ID).

Figure 18-56: Message Processing Example



After the transmit message search, the transmit message with the highest priority of the transmit message buffers that have a pending transmission request (message buffers with the TRQ bit set to 1 in advance) is transmitted.
If a new transmission request is set, the transmit message buffer with the new transmission request is compared with the transmit message buffer with a pending transmission request. If the new transmission request has a higher priority, it is transmitted, unless transmission of a message with a low priority has already started. If transmission of a message with a low priority has already started, however, the new transmission request is transmitted later. The highest priority is determined according to the following rules.

Table 18-26: Message Transmission

Priority	Conditions	Description
1 (high)	Value of first 11 bits of ID [ID28 to ID18]:	The message frame with the lowest value represented by the first 11 bits of the ID is transmitted first. If the value of an 11-bit standard ID is equal to or smaller than the first 11 bits of a 29-bit extended ID, the 11-bit standard ID has a higher priority than message frame with the 29-bit extended ID.
2	Frame type	A data frame with an 11-bit standard ID (RTR bit is cleared to 0) has a higher priority than a remote frame with a standard ID and a message frame with an extended ID.
3	ID type	A message frame with a standard ID (IDE bit is cleared to 0) has a higher priority than a message frame with an extended ID.
4	Value of lower 18 bits of ID [ID17 to ID0]:	If more than one transmission-pending extended ID message frame have equal values in the first 11 bits of the ID and the same frame type (equal RTR bit values), the message frame with the lowest value in the lower 18 bits of its extended ID is transmitted first.
5 (low)	Message buffer number	If two or more message buffers request transmission of message frames with the same ID, the message from the message buffer with the lowest message buffer number is transmitted first.

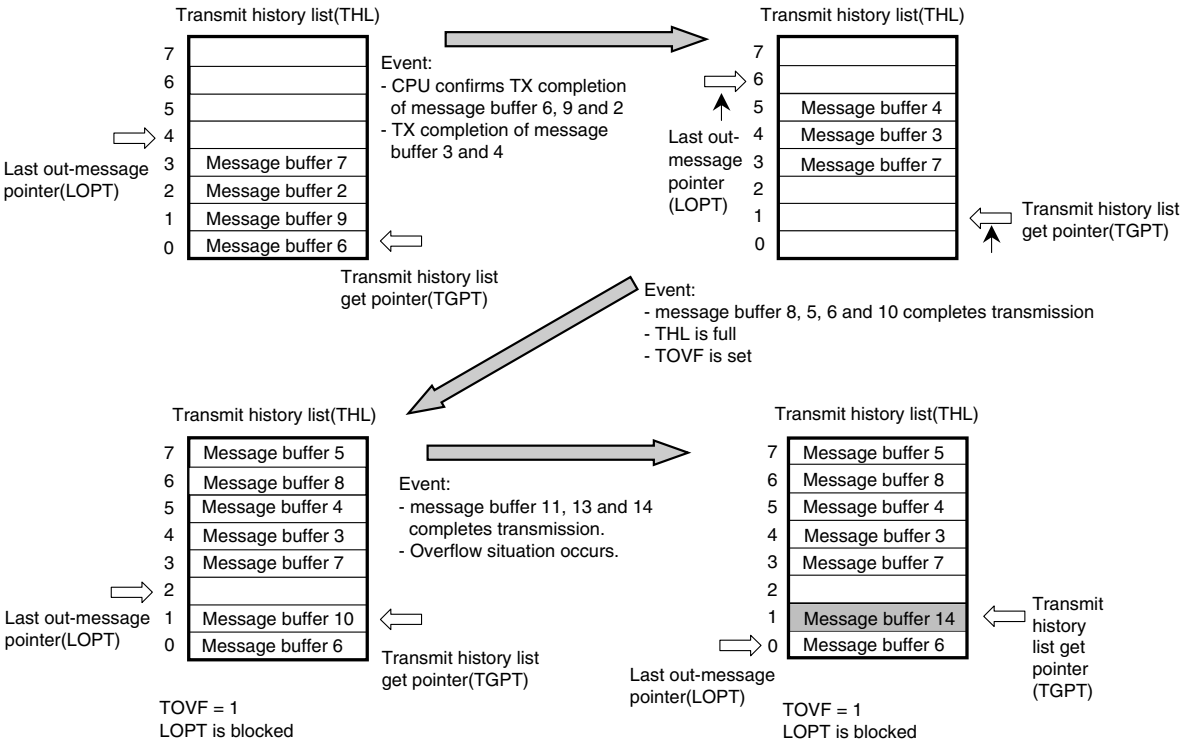
Remark: If automatic block transmission request bit ABTTRG is set to 1 in the normal operation mode with ABT, the TRQ bit is set to 1 only for one message buffer in the ABT message buffer group.
If the ABT mode was triggered by ABTTRG (1), one TRQ is set to 1 in the ABT area (buffer 0 through 7). Beyond this TRQ, the application can request transmissions (set TRQ to 1) for other TX-message buffers that do not belong to the ABT area. In that case an interval arbitration process (TX-search) evaluates all TX-message buffers with TRQ set to 1 and chooses the message buffer that contains the highest prioritized identifier for the next transmission. If there are 2 or more identifiers that have the highest priority (i.e. identical identifiers), the message located at the lowest message buffer number is transmitted at first.
Upon successful transmission of a message frame, the following operations are performed.

- The TRQ flag of the corresponding transmit message buffer is automatically cleared to 0.
- The transmission completion status bit CINTS0 of the CnINTS register is set to 1 (if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).
- An interrupt request signal INTCnTRX is output (if the CIE0 bit of the CnIE register is set to 1 and if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).

18.10.2 Transmit history list function

The transmit history list (THL) function records in the transmit history list the number of the transmit message buffer from which data or remote frames have been were sent. The THL consists of storage elements equivalent to up to seven messages, the last out-message pointer (LOPT) with the corresponding CnLOPT register, and the transmit history list get pointer (TGPT) with the corresponding CnTGPT register.
The THL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.
The CnLOPT register holds the contents of the THL element indicated by the value of the LOPT pointer minus 1. By reading the CnLOPT register, therefore, the number of the message buffer that transmitted a data frame or remote frame first can be checked. The LOPT pointer is utilized as a write pointer that indicates to what part of the THL a message buffer number is recorded. Any time a data frame or remote frame is transmitted, the corresponding message buffer number is recorded to the THL element indicated by the LOPT pointer. Each time recording to the THL has been completed, the LOPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.
The TGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the THL. This pointer indicates the first THL element that the CPU has not yet read. By reading the CnTGPT register by software, the number of a message buffer that has completed transmission can be read. Each time a message buffer number is read from the CnTGPT register, the TGPT pointer is automatically incremented.
If the value of the TGPT pointer matches the value of the LOPT pointer, the THPM bit (transmit history list pointer match) of the CnTGPT register is set to 1. This indicates that no message buffer numbers that have not been read remain in the THL. If a new message buffer number is recorded, the LOPT pointer is incremented and because its value no longer matches the value of the TGPT pointer, the THPM bit is cleared. In other words, the numbers of the unread message buffers exist in the THL.
If the LOPT pointer is incremented and matches the value of the TGPT pointer minus 1, the TOVF bit (transmit history list overflow) of the CnTGPT register is set to 1. This indicates that the THL is full of message buffer numbers that have not been read. If a new message is received and stored, the message buffer number recorded last is overwritten by the number of the message buffer that transmitted its message afterwards. After the TOVF bit has been set (1), therefore, the recorded message buffer numbers in the THL do not completely reflect the chronological order.
However the other transmitted messages can be found by a CPU search applied to all transmit message buffers unless the CPU has not overwritten a transmit object in one of these buffers beforehand. In total up to six transmission completions can occur without overflowing the THL.

Figure 18-57: Transmit History List



TOVF = 1 defines that LOPT equals TOPT - 1 while message buffer number stored to element indicated by LOPT - 1.

18.10.3 Automatic block transmission (ABT)

The automatic block transmission (ABT) function is used to transmit two or more data frames successively with no CPU interaction. The maximum number of transmit message buffers assigned to the ABT function is eight (message buffer numbers 0 to 7).
By setting OPMODE2 to OPMODE0 bits of the CnCTRL register to 010B, “normal operation mode with automatic block transmission function” (hereafter referred to as ABT mode) can be selected.
To issue an ABT transmission request, define the message buffers by software first. Set the MA0 bit (1) in all the message buffers used for ABT, and define all the buffers as transmit message buffers by setting MT2 to MT0 bits to 000B. Be sure to set the same ID for each message buffer for ATB even when that ID is being used for all the message buffers. To use two or more IDs, set the ID of each message buffer by using the CnMIDLm and CnMIDHm registers. Set the CnMDLCm and CnMDATA0m to CnMDATA7m registers before issuing a transmission request for the ABT function.
After initialization of message buffers for ABT is finished, the RDY bit needs to be set (1). In the ABT mode, the TRQ bit does not have to be manipulated by software.
After the data for the ABT message buffers has been prepared, set the ABTTRG bit to 1. Automatic block transmission is then started. When ABT is started, the TRQ bit in the first message buffer (message buffer 0) is automatically set to 1. After transmission of the data of message buffer 0 has finished, TRQ of the next message buffer, message buffer 1, is set automatically. In this way, transmission is executed successively.
A delay time can be inserted by program in the interval in which the transmission request (TRQ) is automatically set while successive transmission is being executed. The delay time to be inserted is defined by the CnGMABTD register. The unit of the delay time is DBT (data bit time). DBT depends on the setting of the CnBRP and CnBTR registers.
Among transmit objects within the ABT-area, the priority of the transmission ID is not evaluated. The data of message buffers 0 to 7 are sequentially transmitted. When transmission of the data frame from message buffer 7 has been completed, the ABTTRG bit is automatically cleared to 0 and the ABT operation is finished.

If the RDY bit of an ABT message buffer is cleared during ABT, no data frame is transmitted from that buffer, ABT is stopped, and the ABTTRG bit is cleared. After that, transmission can be resumed from the message buffer where ABT stopped, by setting the RDY and ABTTRG bits to 1 by software. To not resume transmission from the message buffer where ABT stopped, the internal ABT engine can be reset by setting the ABTCLR bit to 1 while ABT mode is stopped and ABTTRG is cleared to 0. In this case, transmission is started from message buffer 0 if the ABTCLR bit is cleared to 0 and then the ABTTRG bit is set to 1.
An interrupt can be used to check if data frames have been transmitted from all the message buffers for ABT. To do so, the IE bit of the CnMCTRLm register of each message buffer except the last message buffer needs to be cleared (0). If a transmit message buffer other than those used by the ABT function (message buffer 8 to m_{MAX} - 1[Note]) is assigned to a transmit message buffer, the priority of the message to be transmitted is determined by the priority of the transmission message buffer ID of the ABT message buffer whose transmission is currently held pending and the transmission ID of the message buffers other than those used by the ABT function.
Transmission of a data frame from an ABT message buffer is not recorded in the transmit history list (THL).

Cautions: 1. Set the ABTCLR bit to 1 while the ABTTRG bit is cleared to 0 in order to resume ABT operation at buffer No.0. If the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1, the subsequent operation is not guaranteed.

2. If the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared immediately after the processing of the clearing request is completed.

3. Do not set the ABTTRG bit in the initialization mode. If the ABTTRG bit is set in the initialization mode, the proper operation is not guaranteed after the mode is changed from the initialization mode to the ABT mode.

4. Do not set TRQ of the ABT message buffers to 1 by software in the normal operation mode with ABT. Otherwise, the operation is not guaranteed.

5. The CnGMABTD register is used to set the delay time that is inserted in the period from completion of the preceding ABT message to setting of the TRQ bit for the next ABT message when the transmission requests are set in the order of message numbers for each message for ABT that is successively transmitted in the ABT mode. The timing at which the messages are actually transmitted onto the CAN bus varies depending on the status of transmission from other stations and the status of the setting of the transmission request for messages other than the ABT messages (message buffer 8 to m_{MAX} - 1[Note]).

6. If a transmission request is made for a message other than an ABT message and if no delay time is inserted in the interval in which transmission requests for ABT are automatically set (CnGMABTD = 00H), messages other than ABT messages may be transmitted not depending on their priority compared to the priority of the ABT message.

7. Do not clear the RDY bit to 0 when ABTTRG = 1.

8. If a message is received from another node while normal operation mode with ABT is active, the TX-message from the ABT-area may be transmitted with delay of one frame although CnGMABTD register was set up with 00h.

Note: m_{MAX} = 32

18.10.4 Transmission abort process

(1) Transmission abort in normal operation mode

The user can clear the TRQ bit of the CnMCTRLm register to 0 to abort a transmission request. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the CnCTRL register and the CnTGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing in **Figure 18-71, "Transmission Abort Processing (Except Normal Operation Mode with ABT)," on page 847**).

(2) Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)

The user can clear the ABTTRG bit of the CnGMABT register to 0 to abort a transmission request. After checking the ABTTRG bit of the CnGMABT register = 0, clear the TRQ bit of the CnMCTRLm register to 0. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the CnCTRL register and the CnTGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing).

(3) Transmission abort in normal operation mode with automatic block transmission (ABT)

To abort ABT that is already started, clear the ABTTRG bit of the CnGMABT register to 0. In this case, the ABTTRG bit remains 1 if an ABT message is currently being transmitted and until the transmission is completed (successfully or not), and is cleared to 0 as soon as transmission is finished. This aborts ABT.

If the last transmission (before ABT) was successful, the normal operation mode with ABT is left with the internal ABT pointer pointing to the next message buffer to be transmitted.

In the case of an erroneous transmission, the position of the internal ABT pointer depends on the status of the TRQ bit in the last transmitted message buffer. If the TRQ bit is set to 1 when clearing the ABTTRG bit is requested, the internal ABT pointer points to the last transmitted message buffer (for details, refer to the process in **Figure 18-72, "Transmission Abort Processing Except for ABT Transmission (Normal Operation Mode with ABT)," on page 848**). If the TRQ bit is cleared to 0 when clearing the ABTTRG bit is requested, the internal ABT pointer is incremented (+1) and points to the next message buffer in the ABT area.

Caution: Be sure to abort ABT by clearing ABTTRG to 0. The operation is not guaranteed if aborting transmission is requested by clearing RDY.

When the normal operation mode with ABT is resumed after ABT has been aborted and ABTTRG is set to 1, the next ABT message buffer to be transmitted can be determined from the following table.

Table 18-27: Transmission Abort

Status of TRQ of ABT Message Buffer	Abort After Successful Transmission	Abort after erroneous transmission
Set (1)	Next message buffer in the ABT area**Note**	Same message buffer in the ABT area
Cleared (0)	Next message buffer in the ABT area**Note**	Next message buffer in the ABT area**Note**

Note: The above resumption operation can be performed only if a message buffer ready for ABT exists in the ABT area. For example, an abort request that is issued while ABT of message buffer 7 is in progress is regarded as completion of ABT, rather than abort, if transmission of message buffer 7 has been successfully completed, even if ABTTRG is cleared to 0. If the RDY bit in the next message buffer in the ABT area is cleared to 0, the internal ABT pointer is retained, but the resumption operation is not performed even if ABTTRG is set to 1, and ABT ends immediately.

18.10.5 Remote frame transmission

Remote frames can be transmitted only from transmit message buffers. Set whether a data frame or remote frame is transmitted via the RTR bit of the CnMCONFm register. Setting (1) the RTR bit sets remote frame transmission.

18.11 Power Save Modes

18.11.1 CAN sleep mode

The CAN sleep mode can be used to set the CAN controller to standby mode in order to reduce power consumption. The CAN module can enter the CAN sleep mode from all operation modes. Release of the CAN sleep mode returns the CAN module to exactly the same operation mode from which the CAN sleep mode was entered.
In the CAN sleep mode, the CAN module does not transmit messages, even when transmission requests are issued or pending.

(1) Entering CAN sleep mode

The CPU issues a CAN sleep mode transition request by writing 01B to the PSMODE1, PSMODE0 bits of the CnCTRL register.
This transition request is only acknowledged only under the following conditions.

- The CAN module is already in one of the following operation modes
  - Normal operation mode
  - Normal operation mode with ABT
  - Receive-only mode
  - Single-shot mode
  - Self-test mode
  - CAN stop mode in all the above operation modes
- The CAN bus state is bus idle (the 4th bit in the interframe space is recessive)[Note]
- No transmission request is pending

Note: If the CAN bus is fixed to dominant, the request for transition to the CAN sleep mode is held pending.

If any of the conditions mentioned above is not met, the CAN module will operate as follows:

- If the CAN sleep mode is requested from the initialization mode, the CAN sleep mode transition request is ignored and the CAN module remains in the initialization mode.
- If the CAN bus state is not bus idle (i.e., the CAN bus state is either transmitting or receiving) when the CAN sleep mode is requested in one of the operation modes, immediate transition to the CAN sleep mode is not possible. In this case, the CAN sleep mode transition request is held pending until the CAN bus state becomes bus idle (the 4th bit in the interframe space is recessive). In the time from the CAN sleep mode request to successful transition, the PSMODE1, PSMODE0 bits remain 00B. When the module has entered the CAN sleep mode, PSMODE1, PSMODE0 bits are set to 01B.
- If a request for transition to the initialization mode and a request for transition to the CAN sleep are made at the same time while the CAN module is in one of the operation modes, the request for the initialization mode is enabled. The CAN module enters the initialization mode at a predetermined timing. At this time, the CAN sleep mode request is not held pending and is ignored.
- Even when initialization mode and sleep mode are requested simultaneously (i.e. the first request has not been granted while the second request is made), the request for initialization has priority over the sleep mode request. The sleep mode request is cancelled when the initialization mode is requested. When a pending request for initialization mode is present, a subsequent request for sleep mode request is cancelled right at the point in time where it was submitted.

(2) Status in CAN sleep mode

The CAN module is in one of the following states after it enters the CAN sleep mode:

- The internal operating clock is stopped and the power consumption is minimized.
- The function to detect the falling edge of the CAN reception pin (CRXDn) remains in effect to wake up the CAN module from the CAN bus.
- To wake up the CAN module from the CPU, data can be written to PSMODE1, PSMODE0 bits of the CAN module control register (CnCTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for CnLIPT, CnRGPT, CnLOPT, and CnTGPT.
- The CAN message buffer registers cannot be written or read.
- A request for transition to the initialization mode is not acknowledged and is ignored.

(3) Releasing CAN sleep mode

The CAN sleep mode is released by the following events:

- When the CPU writes 00B to the PSMODE1, PSMODE0 bits of the CnCTRL register
- A falling edge at the CAN reception pin (CRXDn) (i.e. the CAN bus level shifts from recessive to dominant)

Caution: Even if the falling edge belongs to the SOF of a receive message, this message will not be received and stored. If the CPU has turned off the clock to the CAN while the CAN was in sleep mode, even subsequently the CAN sleep mode will not be released and PSMODE1, PSMODE0 bits will continue to be 01B unless the clock to the CAN is supplied again. In addition to this, the receive message will not be received after that.

After releasing the sleep mode, the CAN module returns to the operation mode from which the CAN sleep mode was requested and the PSMODE1, PSMODE0bits of the CnCTRL register are reset to 00B. If the CAN sleep mode is released by a change in the CAN bus state, the CINTS5 bit of the CnINTS register is set to 1, regardless of the CIE bit of the CnIE register. After the CAN module is released from the CAN sleep mode, it participates in the CAN bus again by automatically detecting 11 consecutive recessive-level bits on the CAN bus.
When a request for transition to the initialization mode is made while the CAN module is in the CAN sleep mode, that request is ignored; the CPU has to be released from sleep mode by software first before entering the initialization mode.

18.11.2 CAN stop mode

The CAN stop mode can be used to set the CAN controller to standby mode to reduce power consumption. The CAN module can enter the CAN stop mode only from the CAN sleep mode. Release of the CAN stop mode puts the CAN module in the CAN sleep mode.
The CAN stop mode can only be released (entering CAN sleep mode) by writing 01B to the PSMODE1, PSMODE0 bits of the CnCTRL register and not by a change in the CAN bus state. No message is transmitted even when transmission requests are issued or pending.

(1) Entering CAN stop mode

A CAN stop mode transition request is issued by writing 11B to the PSMODE1, PSMODE0 bits of the
CnCTRL register.
A CAN stop mode request is only acknowledged when the CAN module is in the CAN sleep mode. In all other modes, the request is ignored.

Caution: To set the CAN module to the CAN stop mode, the module must be in the CAN Sleep mode. To confirm that the module is in the sleep mode, check that PSMODE1, PSMODE0 = 01B, and then request the CAN stop mode. If a bus change occurs at the CAN reception pin (CRXD) while this process is being performed, the CAN sleep mode is automatically released. In this case, the CAN stop mode transition request cannot be acknowledged.

(2) Status in CAN stop mode

The CAN module is in one of the following states after it enters the CAN stop mode:

- The internal operating clock is stopped and the power consumption is minimized.
- To wake up the CAN module from the CPU, data can be written to PSMODE1, PSMODE0 of the CAN module control register (CnCTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for CnLIPT, CnRGPT, CnLOPT, and CnTGPT.
- The CAN message buffer registers cannot be written or read.
- An initialization mode transition request is not acknowledged and is ignored.

(3) Releasing CAN stop mode

The CAN stop mode can only be released by writing 01B to the PSMODE1, PSMODE0 bits of the CnCTRL register. After releasing the CAN stop mode, the CAN module enters the CAN sleep mode.
When the initialization mode is requested while the CAN module is in the CAN stop mode, that request is ignored; the CPU has to release the stop mode and subsequently CAN sleep mode before entering the initialization mode. It is impossible to enter the other operation mode directly from the CAN stop mode not entering the CAN sleep mode, that request is ignored.

18.11.3 Example of using power saving modes

In some application systems, it may be necessary to place the CPU in a power saving mode to reduce the power consumption. By using the power saving mode specific to the CAN module and the power saving mode specific to the CPU in combination, the CPU can be woken up from the power saving status by the CAN bus.

Here is an example of using the power saving modes.

First, put the CAN module in the CAN sleep mode (PSMODE = 01B). Next, put the CPU in the power saving mode. If an edge transition from recessive to dominant is detected at the CAN reception pin (CRXDn) in this status, the CINTS5 bit in the CAN module is set to 1. If the CIE5 bit of the CnCTRL register is set to 1, a wakeup interrupt (INTCnWUP) is generated. The CAN module is automatically released from the CAN sleep mode (PSMODE = 00B) and returns to the normal operation mode. The CPU, in response to INTCnWUP, can release its own power saving mode and return to the normal operation mode.
To further reduce the power consumption of the CPU, the internal clocks, including that of the CAN module, may be stopped. In this case, the operating clock supplied to the CAN module is stopped after the CAN module is put in the CAN sleep mode. Then the CPU enters a power saving mode in which the clock supplied to the CPU is stopped. If an edge transition from recessive to dominant is detected at the CAN reception pin (CRXDn) in this status, the CAN module can set the CINTS5 bit to 1 and generate the wakeup interrupt (INTCnWUP) even if it is not supplied with the clock. The other functions, however, do not operate because clock supply to the CAN module is stopped, and the module remains in the CAN sleep mode. The CPU, in response to INTCnWUP, releases its power saving mode, resumes supply of the internal clocks, including the clock to the CAN module, after the oscillation stabilisation time has elapsed, and starts instruction execution. The CAN module is immediately released from the CAN sleep mode when clock supply is resumed, and returns to the normal operation mode (PSMODE = 00B).

18.12 Interrupt Function

18.12.1 Interrupts generated by CAN module

The CAN module provides 6 different interrupt sources.
The occurrence of these interrupt sources is stored in interrupt status registers. Four separate interrupt request signals are generated from the six interrupt sources. When an interrupt request signal that corresponds to two or more interrupt sources is generated, the interrupt sources can be identified by using an interrupt status register. After an interrupt source has occurred, the corresponding interrupt status bit must be cleared to 0 by software.

Table 18-28: List of CAN Module Interrupt Sources

No	Interrupt Status Bit		Interrupt Enable Bit		Interrupt Request Signal	Interrupt Source Description
	Name	Register	Name	Register		
1	CINTS0**Note 1**	CnINTS	CIE0**Note 1**	CnIE	INTCnTRX	Message frame successfully transmitted from message buffer m
2	CINTS1**Note 1**	CnINTS	CIE1**Note 1**	CnIE	INTCnREC	Valid message frame reception in message buffer m
3	CINTS2	CnINTS	CIE2	CnIE	INTCnERR	CAN module error state interrupt**Note 2**
4	CINTS3	CnINTS	CIE3	CnIE		CAN module protocol error interrupt**Note 3**
5	CINTS4	CnINTS	CIE4	CnIE		CAN module arbitration loss interrupt
6	CINTS5	CnINTS	CIE5	CnIE	INTCnWUP	CAN module wakeup interrupt from CAN sleep mode**Note 4**

Notes: 1. The IE bit (message buffer interrupt enable bit) in the CnMCTRL register of the corresponding message buffer has to be set to 1 for that message buffer to participate in the interrupt generation process.

2. This interrupt is generated when the transmission/reception error counter is at the warning level, or in the error passive or bus-off state.

3. This interrupt is generated when a stuff error, form error, ACK error, bit error, or CRC error occurs.

4. This interrupt is generated when the CAN module is woken up from the CAN sleep mode because a falling edge is detected at the CAN reception pin (CAN bus transition from recessive to dominant).

Remark: n = 0, 1

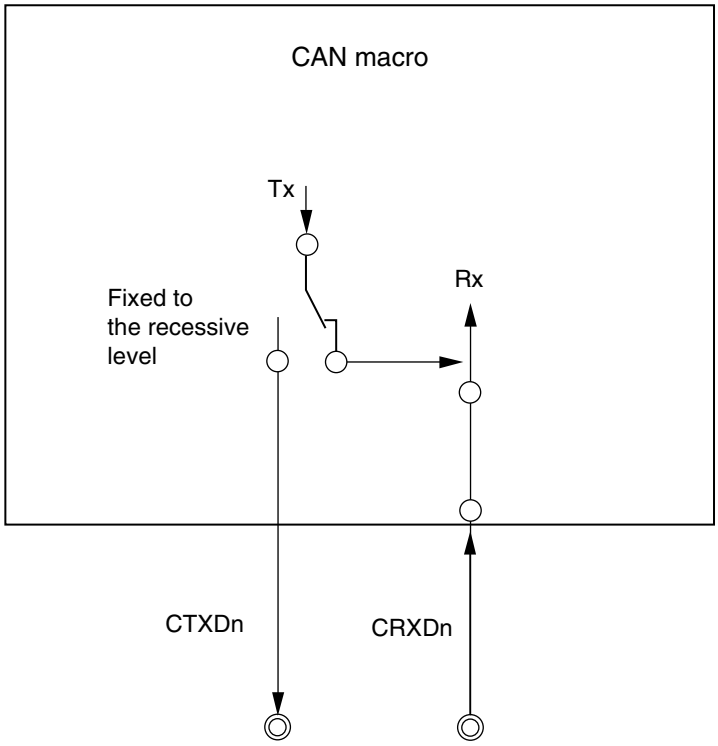
18.13 Diagnosis Functions and Special Operational Modes

The CAN module provides a receive-only mode, single-shot mode, and self-test mode to support CAN bus diagnosis functions or the operation of specific CAN communication methods.

18.13.1 Receive-only mode

The receive-only mode is used to monitor receive messages without causing any interference on the CAN bus and can be used for CAN bus analysis nodes.
For example, this mode can be used for automatic baud-rate detection. The baud rate in the CAN module is changed until "valid reception" is detected, so that the baud rates in the module match ("valid reception" means a message frame has been received in the CAN protocol layer without occurrence of an error and with an appropriate ACK between nodes connected to the CAN bus). A valid reception does not require message frames to be stored in a receive message buffer (data frames) or transmit message buffer (remote frames). The event of valid reception is indicated by setting the VALID bit of the CnCTRL register (1).

Figure 18-58: CAN Module Terminal Connection in Receive-Only Mode



In the receive-only mode, no message frames can be transmitted from the CAN module to the CAN bus. Transmit requests issued for message buffers defined as transmit message buffers are held pending.
In the receive-only mode, the CAN transmission pin (CTXDn) in the CAN module is fixed to the recessive level. Therefore, no active error flag can be transmitted from the CAN module to the CAN bus even when a CAN bus error is detected while receiving a message frame. Since no transmission can be issued from the CAN module, the transmission error counter TEC is never updated. Therefore, a CAN module in the receive-only mode does not enter the bus-off state.

Furthermore, ACK is not returned to the CAN bus in this mode upon the valid reception of a message frame. Internally, the local node recognizes that it has transmitted ACK. An overload frame cannot be transmitted to the CAN bus.

Caution: If only two CAN nodes are connected to the CAN bus and one of them is operating in the receive-only mode, there is no ACK on the CAN bus. Due to the missing ACK, the transmitting node will transmit an active error flag, and repeat transmitting a message frame. The transmitting node becomes error passive after transmitting the message frame 16 times (assuming that the error counter was 0 in the beginning and no other errors have occurred). After the message frame for the 17th time is transmitted, the transmitting node generates a passive error flag. The receiving node in the receive-only mode detects the first valid message frame at this point, and the VALID bit is set to 1 for the first time.

18.13.2 Single-shot mode

In the single-shot mode, automatic re-transmission as defined in the CAN protocol is switched off. (According to the CAN protocol, a message frame transmission that has been aborted by either arbitration loss or error occurrence has to be repeated without control by software.) All other behaviour of single shot mode is identical to normal operation mode. Features of single shot mode can not be used in combination with normal mode with ABT.
The single-shot mode disables the re-transmission of an aborted message frame transmission according to the setting of the AL bit of the CnCTRL register. When the AL bit is cleared to 0, re-transmission upon arbitration loss and upon error occurrence is disabled. If the AL bit is set to 1, re-transmission upon error occurrence is disabled, but re-transmission upon arbitration loss is enabled. As a consequence, the TRQ bit in a message buffer defined as a transmit message buffer is cleared to 0 by the
following events:

- Successful transmission of the message frame
- Arbitration loss while sending the message frame
- Error occurrence while sending the message frame

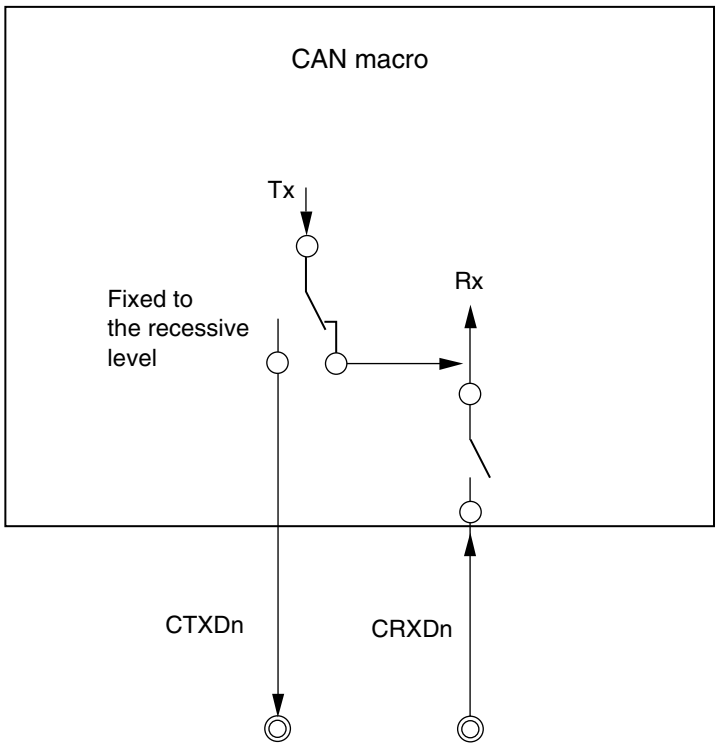
The events arbitration loss and error occurrence can be distinguished by checking the CINTS4 and CINTS3 bits of the CnINTS register respectively, and the type of the error can be identified by reading the LEC2 to LEC0 bits of the CnLEC register.
Upon successful transmission of the message frame, the transmit completion interrupt bit CINTS0 of the CnINTS register is set to 1. If the CIE0 bit of the CnIE register is set to 1 at this time, an interrupt request signal is output.
The single-shot mode can be used when emulating time-triggered communication methods (e.g. TTCAN level 1).

Caution: The AL bit is only valid in Single-shot mode. It does not influence the operation of re-transmission upon arbitration loss in the other operation modes.

18.13.3 Self-test mode

In the self-test mode, message frame transmission and message frame reception can be tested without connecting the CAN node to the CAN bus or without affecting the CAN bus.
In the self-test mode, the CAN module is completely disconnected from the CAN bus, but transmission and reception are internally looped back. The CAN transmission pin (CTXDn) is fixed to the recessive level.
If the falling edge on the CAN reception pin (CRXDn) is detected after the CAN module has entered the CAN sleep mode from the self-test mode, however, the module is released from the CAN sleep mode in the same manner as the other operation modes. To keep the module in the CAN sleep mode, use the CAN reception pin (CRXDn) as a port pin.

Figure 18-59: CAN Module Terminal Connection in Self-Test Mode



18.14 Time Stamp Function

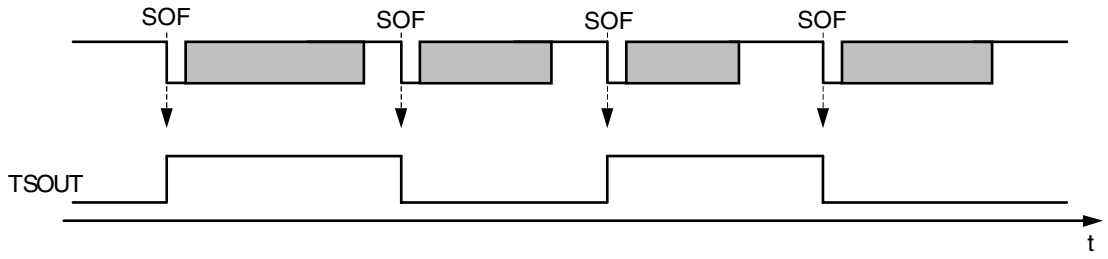
CAN is an asynchronous, serial protocol. All nodes connected to the CAN bus have a local, autonomous clock. As a consequence, the clocks of the nodes have no relation (i.e., the clocks are asynchronous and may have different frequencies).
In some applications, however, a common time base over the network (= global time base) is needed. In order to build up a global time base, a time stamp function is used. The essential mechanism of a time stamp function is the capture of timer values triggered by signals on the CAN bus.

The CAN controller supports the capturing of timer values triggered by a specific frame. An on-chip 16-bit capture timer unit in a microcontroller system is used in addition to the CAN controller. The 16-bit capture timer unit captures the timer value according to a trigger signal (TSOUT) for capturing that is output when a data frame is received from the CAN controller. The CPU can retrieve the time of occurrence of the capture event, i.e., the time stamp of the message received from the CAN bus, by reading the captured value. TSOUT can be selected from the following two event sources and is specified by the TSSEL bit of the CnTS register:

- SOF event (start of frame) (TSSEL = 0)
- EOF event (last bit of end of frame) (TSSEL = 1)

The TSOUT signal is enabled by setting the TSEN bit of the CnTS register to 1.

Figure 18-60: Timing Diagram of Capture Signal TSOUT



TSOUT toggles its level upon occurrence of the selected event during data frame reception (in the above timing diagram, the SOF is used as the trigger event source). To capture a timer value by using TSOUT, the capture timer unit must detect the capture signal at both the rising edge and falling edge.
This time stamp function is controlled by the TSLOCK bit of the CnTS register. When TSLOCK is cleared to 0, TSOUT toggles upon occurrence of the selected event. If TSLOCK is set to 1, TSOUT toggles upon occurrence of the selected event, but the toggle is stopped as the TSEN bit is automatically cleared to 0 when a data frame starts to be received and stored in message buffer 0. This suppresses the subsequent toggle occurrence by TSOUT, so that the time stamp value toggled last (= captured last) can be saved as the time stamp value of the time at which the data frame was received in message buffer 0.

Caution: The time stamp function using TSLOCK is to stop toggle of TSOUT by receiving a data frame in message buffer 0. Therefore, message buffer 0 must be set as a receive message buffer. Since a receive message buffer cannot receive a remote frame, toggle of TSOUT cannot be stopped by reception of a remote frame. Toggle of TSOUT does not stop when a data frame is received in a message buffer other than message buffer 0. For these reasons, a data frame cannot be received in message buffer 0 when the CAN module is in the normal operation mode with ABT, because message buffer 0 must be set as a transmit message buffer. In this operation mode, therefore, the function to stop toggle of TSOUT by TSLOCK cannot be used.

18.15 Baud Rate Settings

Make sure that the settings are within the range of limit values for ensuring correct operation of the CAN controller, as follows.

(a) 5TQ ≤SPT (sampling point) ≤17 TQ
SPT = TSEG1 + 1

(b) 8 TQ ≤DBT (data bit time) ≤25 TQ
DBT = TSEG1 + TSEG2 + 1TQ = TSEG2 + SPT

(c) 1 TQ ≤SJW (synchronization jump width) ≤4TQ
SJW ≤DBT – SPT

(d) 4 ≤TSEG1 ≤16 [3 (Setting value of TSEG1[3:0] ≤15]

(e) 1 ≤TSEG2 ≤8 [0 (Setting value of TSEG2[2:0] ≤7]

Remark: TQ = $1/f_{TQ}$ (f_{TQ}: CAN protocol layer basic system clock)
TSEG1[3:0] (Bits 3 to 0 of CANn bit rate register (CnBTR))
TSEG2[2:0] (Bits 10 to 8 of CANn bit rate register (CnBTR))

Table 18-29 shows the combinations of bit rates that satisfy the above conditions.

Table 18-29: Settable Bit Rate Combinations (1/3)

Valid Bit Rate Setting					CnBTR Register Setting Value		Sampling Point (Unit,%)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1[3:0]	TSEG2[2:0]	
25	1	8	8	8	1111	111	68.0
24	1	7	8	8	1110	111	66.7
24	1	9	7	7	1111	110	70.8
23	1	6	8	8	1101	111	65.2
23	1	8	7	7	1110	110	69.6
23	1	10	6	6	1111	101	73.9
22	1	5	8	8	1100	111	63.6
22	1	7	7	7	1101	110	68.2
22	1	9	6	6	1110	101	72.7
22	1	11	5	5	1111	100	77.3
21	1	4	8	8	1011	111	61.9
21	1	6	7	7	1100	110	66.7
21	1	8	6	6	1101	101	71.4
21	1	10	5	5	1110	100	76.2
21	1	12	4	4	1111	011	81.0
20	1	3	8	8	1010	111	60.0
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4
17	1	2	7	7	1000	110	58.8

Table 18-29: Settable Bit Rate Combinations (2/3)

Valid Bit Rate Setting					CnBTR Register Setting Value		Sampling Point (Unit,%)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1[3:0]	TSEG2[2:0]	
17	1	4	6	6	1001	101	64.7
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7
15	1	12	1	1	1100	000	93.3
14	1	1	6	6	0110	101	57.1
14	1	3	5	5	0111	100	64.3
14	1	5	4	4	1000	011	71.4
14	1	7	3	3	1001	010	78.6
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9

Table 18-29: Settable Bit Rate Combinations (3/3)

Valid Bit Rate Setting					CnBTR Register Setting Value		Sampling Point (Unit,%)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1[3:0]	TSEG2[2:0]	
10	1	1	4	4	0100	011	60.0
10	1	3	3	3	0101	010	70.0
10	1	5	2	2	0110	001	80.0
10	1	7	1	1	0111	000	90.0
9	1	2	3	3	0100	010	66.7
9	1	4	2	2	0101	001	77.8
9	1	6	1	1	0110	000	88.9
8	1	1	3	3	0011	010	62.5
8	1	3	2	2	0100	001	75.0
8	1	5	1	1	0101	000	87.5
7 Note	1	2	2	2	0011	001	71.4
7 Note	1	4	1	1	0100	000	85.7
6 Note	1	1	2	2	0010	001	66.7
6 Note	1	3	1	1	0011	000	83.3
5 Note	1	2	1	1	0010	000	80.0
4 Note	1	1	1	1	0001	000	75.0

Note: Setting with a DBT value of 7 or less is valid only when the value of the CnBRP register is other than 00H.

Caution: The values in Table 18-29 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

18.15.1 Representative examples of baud rate settings

Tables 18-30 and 18-31 show representative examples of baud rate setting.

Table 18-30: Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (1/2)

Set Baud Rate Value (Unit:kbps)	Division Ratio of CnBRP	CnBRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					CnBTR Register Setting Value		Sampling point (Unit:%)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	8	1	1	3	3	0011	010	62.5
1000	1	00000000	8	1	3	2	2	0100	001	75.0
1000	1	00000000	8	1	5	1	1	0101	000	87.5
500	1	00000000	16	1	1	7	7	0111	110	56.3
500	1	00000000	16	1	3	6	6	1000	101	62.5
500	1	00000000	16	1	5	5	5	1001	100	68.8
500	1	00000000	16	1	7	4	4	1010	011	75.0
500	1	00000000	16	1	9	3	3	1011	010	81.3
500	1	00000000	16	1	11	2	2	1100	001	87.5
500	1	00000000	16	1	13	1	1	1101	000	93.8
500	2	00000001	8	1	1	3	3	0011	010	62.5
500	2	00000001	8	1	3	2	2	0100	001	75.0
500	2	00000001	8	1	5	1	1	0101	000	87.5
250	2	00000001	16	1	1	7	7	0111	110	56.3
250	2	00000001	16	1	3	6	6	1000	101	62.5
250	2	00000001	16	1	5	5	5	1001	100	68.8
250	2	00000001	16	1	7	4	4	1010	011	75.0
250	2	00000001	16	1	9	3	3	1011	010	81.3
250	2	00000001	16	1	11	2	2	1100	001	87.5
250	2	00000001	16	1	13	1	1	1101	000	93.8
250	4	00000011	8	1	3	2	2	0100	001	75.0
250	4	00000011	8	1	5	1	1	0101	000	87.5
125	4	00000011	16	1	1	7	7	0111	110	56.3
125	4	00000011	16	1	3	6	6	1000	101	62.5
125	4	00000011	16	1	5	5	5	1001	100	68.8
125	4	00000011	16	1	7	4	4	1010	011	75.0
125	4	00000011	16	1	9	3	3	1011	010	81.3
125	4	00000011	16	1	11	2	2	1100	001	87.5
125	4	00000011	16	1	13	1	1	1101	000	93.8
125	8	00000111	8	1	3	2	2	0100	001	75.0
125	8	00000111	8	1	5	1	1	0101	000	87.5
100	4	00000011	20	1	7	6	6	1100	101	70.0
100	4	00000011	20	1	9	5	5	1101	100	75.0
100	5	00000100	16	1	7	4	4	1010	011	75.0
100	5	00000100	16	1	9	3	3	1011	010	81.3

Table 18-30: Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (2/2)

Set Baud Rate Value (Unit:kbps)	Division Ratio of CnBRP	CnBRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					CnBTR Register Setting Value		Sampling point (Unit:%)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1 [3:0]	TSEG2 [2:0]	
100	8	00000111	10	1	3	3	3	0101	010	70.0
100	8	00000111	10	1	5	2	2	0110	001	80.0
100	10	00001001	8	1	3	2	2	0100	001	75.0
100	10	00001001	8	1	5	1	1	0101	000	87.5
83.3	4	00000011	24	1	7	8	8	1110	111	66.7
83.3	4	00000011	24	1	9	7	7	1111	110	70.8
83.3	6	00000101	16	1	5	5	5	1001	100	68.8
83.3	6	00000101	16	1	7	4	4	1010	011	75.0
83.3	6	00000101	16	1	9	3	3	1011	010	81.3
83.3	6	00000101	16	1	11	2	2	1100	001	87.5
83.3	8	00000111	12	1	5	3	3	0111	010	75.0
83.3	8	00000111	12	1	7	2	2	1000	001	83.3
83.3	12	00001011	8	1	3	2	2	0100	001	75.0
83.3	12	00001011	8	1	5	1	1	0101	000	87.5
33.3	10	00001001	24	1	7	8	8	1110	111	66.7
33.3	10	00001001	24	1	9	7	7	1111	110	70.8
33.3	12	00001011	20	1	7	6	6	1100	101	70.0
33.3	12	00001011	20	1	9	5	5	1101	100	75.0
33.3	15	00001110	16	1	7	4	4	1010	011	75.0
33.3	15	00001110	16	1	9	3	3	1011	010	81.3
33.3	16	00001111	15	1	6	4	4	1001	011	73.3
33.3	16	00001111	15	1	8	3	3	1010	010	80.0
33.3	20	00010011	12	1	5	3	3	0111	010	75.0
33.3	20	00010011	12	1	7	2	2	1000	001	83.3
33.3	24	00010111	10	1	3	3	3	0101	010	70.0
33.3	24	00010111	10	1	5	2	2	0110	001	80.0
33.3	30	00011101	8	1	3	2	2	0100	001	75.0
33.3	30	00011101	8	1	5	1	1	0101	000	87.5

Caution: The values in Table 18-30 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Table 18-31: Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (1/2)

Set Baud Rate Value (Unit:kbps)	Division Ratio of CnBRP	CnBRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					CnBTR Register Setting Value		Sampling point (Unit:%)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	16	1	1	7	7	0111	110	56.3
1000	1	00000000	16	1	3	6	6	1000	101	62.5
1000	1	00000000	16	1	5	5	5	1001	100	68.8
1000	1	00000000	16	1	7	4	4	1010	011	75.0
1000	1	00000000	16	1	9	3	3	1011	010	81.3
1000	1	00000000	16	1	11	2	2	1100	001	87.5
1000	1	00000000	16	1	13	1	1	1101	000	93.8
1000	2	00000001	8	1	3	2	2	0100	001	75.0
1000	2	00000001	8	1	5	1	1	0101	000	87.5
500	2	00000001	16	1	1	7	7	0111	110	56.3
500	2	00000001	16	1	3	6	6	1000	101	62.5
500	2	00000001	16	1	5	5	5	1001	100	68.8
500	2	00000001	16	1	7	4	4	1010	011	75.0
500	2	00000001	16	1	9	3	3	1011	010	81.3
500	2	00000001	16	1	11	2	2	1100	001	87.5
500	2	00000001	16	1	13	1	1	1101	000	93.8
500	4	00000011	8	1	3	2	2	0100	001	75.0
500	4	00000011	8	1	5	1	1	0101	000	87.5
250	4	00000011	16	1	3	6	6	1000	101	62.5
250	4	00000011	16	1	5	5	5	1001	100	68.8
250	4	00000011	16	1	7	4	4	1010	011	75.0
250	4	00000011	16	1	9	3	3	1011	010	81.3
250	4	00000011	16	1	11	2	2	1100	001	87.5
250	8	00000111	8	1	3	2	2	0100	001	75.0
250	8	00000111	8	1	5	1	1	0101	000	87.5
125	8	00000111	16	1	3	6	6	1000	101	62.5
125	8	00000111	16	1	7	4	4	1010	011	75.0
125	8	00000111	16	1	9	3	3	1011	010	81.3
125	8	00000111	16	1	11	2	2	1100	001	87.5
125	16	00001111	8	1	3	2	2	0100	001	75.0
125	16	00001111	8	1	5	1	1	0101	000	87.5
100	8	00000111	20	1	9	5	5	1101	100	75.0
100	8	00000111	20	1	11	4	4	1110	011	80.0
100	10	00001001	16	1	7	4	4	1010	011	75.0
100	10	00001001	16	1	9	3	3	1011	010	81.3
100	16	00001111	10	1	3	3	3	0101	010	70.0
100	16	00001111	10	1	5	2	2	0110	001	80.0
100	20	00010011	8	1	3	2	2	0100	001	75.0

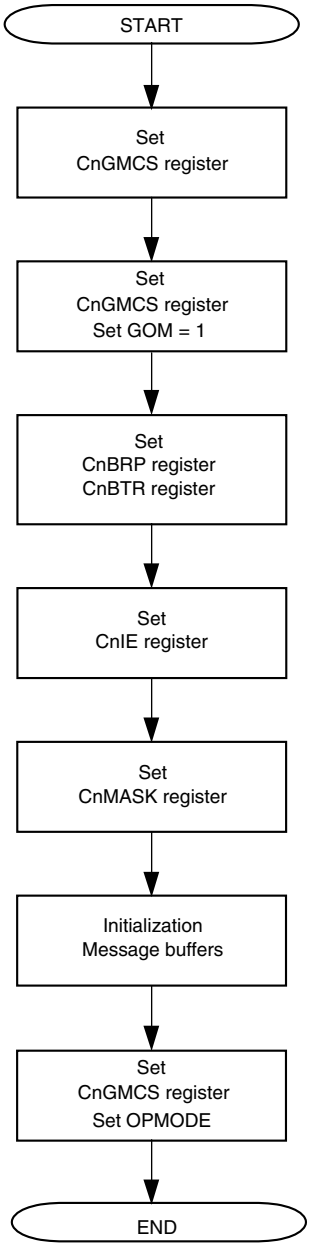
Table 18-31: Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (2/2)

Set Baud Rate Value (Unit:kbps)	Division Ratio of CnBRP	CnBRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					CnBTR Register Setting Value		Sampling point (Unit:%)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1 [3:0]	TSEG2 [2:0]	
83.3	8	00000111	24	1	7	8	8	1110	111	66.7
83.3	8	00000111	24	1	9	7	7	1111	110	70.8
83.3	12	00001011	16	1	7	4	4	1010	011	75.0
83.3	12	00001011	16	1	9	3	3	1011	010	81.3
83.3	12	00001011	16	1	11	2	2	1100	001	87.5
83.3	16	00001111	12	1	5	3	3	0111	010	75.0
83.3	16	00001111	12	1	7	2	2	1000	001	83.3
83.3	24	00010111	8	1	3	2	2	0100	001	75.0
83.3	24	00010111	8	1	5	1	1	0101	000	87.5
33.3	30	00011101	24	1	7	8	8	1110	111	66.7
33.3	30	00011101	24	1	9	7	7	1111	110	70.8
33.3	24	00010111	20	1	9	5	5	1101	100	75.0
33.3	24	00010111	20	1	11	4	4	1110	011	80.0
33.3	30	00011101	16	1	7	4	4	1010	011	75.0
33.3	30	00011101	16	1	9	3	3	1011	010	81.3
33.3	32	00011111	15	1	8	3	3	1010	010	80.0
33.3	32	00011111	15	1	10	2	2	1011	001	86.7
33.3	37	00100100	13	1	6	3	3	1000	010	76.9
33.3	37	00100100	13	1	8	2	2	1001	001	84.6
33.3	40	00100111	12	1	5	3	3	0111	010	75.0
33.3	40	00100111	12	1	7	2	2	1000	001	83.3
33.3	48	00101111	10	1	3	3	3	0101	010	70.0
33.3	48	00101111	10	1	5	2	2	0110	001	80.0
33.3	60	00111011	8	1	3	2	2	0100	001	75.0
33.3	60	00111011	8	1	5	1	1	0101	000	87.5

Caution: The values in Table 18-31 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

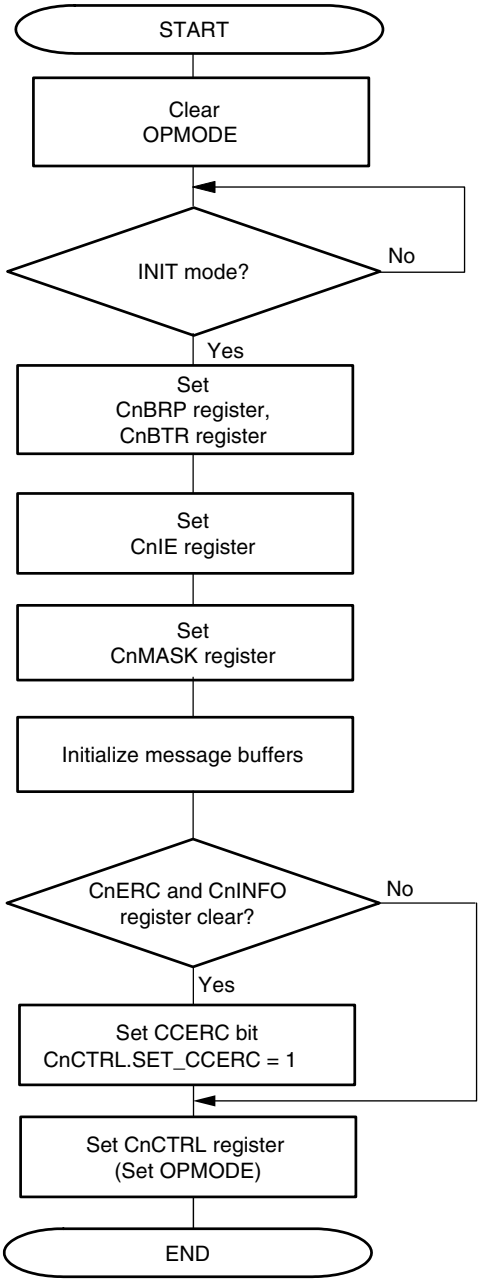
18.16 Operation of CAN Controller

Figure 18-61: Initialization



Remark: OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

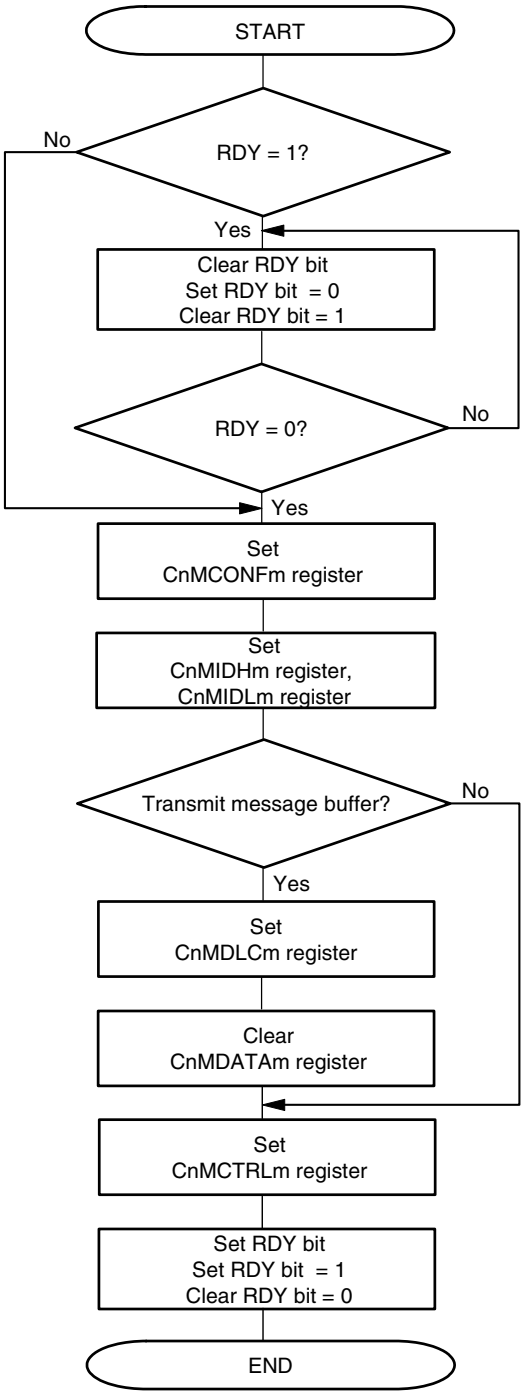
Figure 18-62: Re-initialization



Caution: After setting the CAN module to the initialization mode, avoid setting the module to another operation mode immediately after. If it is necessary to immediately set the module to another operation mode, be sure to access registers other than the CnCTRL and CnGMCTRL registers (e.g. set a message buffer).

Remark: OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 18-63: Message Buffer Initialization

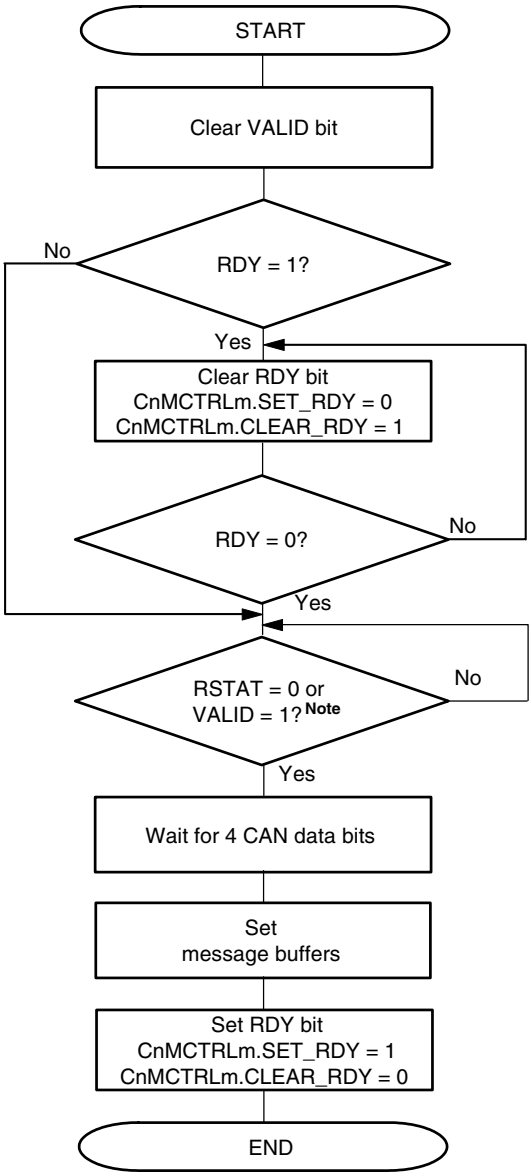


Cautions: 1. Before a message buffer is initialized, the RDY bit must be cleared.

2. Make the following settings for message buffers not used by the application:
- Clear the RDY, TRQ, and DN bits of the CnMCTRLm register to 0.
- Clear the MA0 bit of the CnMCONFm register to 0.

Figure 18-64 shows the processing for a receive message buffer (MT2 to MT0 bits of CnMCONFm register = 001B to 101B).

Figure 18-64: Message Buffer Redefinition



Note: Confirm that a message is being received because RDY bit must be set after a message is completely received.

Figure 18-65 shows the processing for a transmit message buffer during transmission (MT2 to MT0 bits of CnMCONFm register = 00B)

Figure 18-65: Message Buffer Redefinition during Transmission

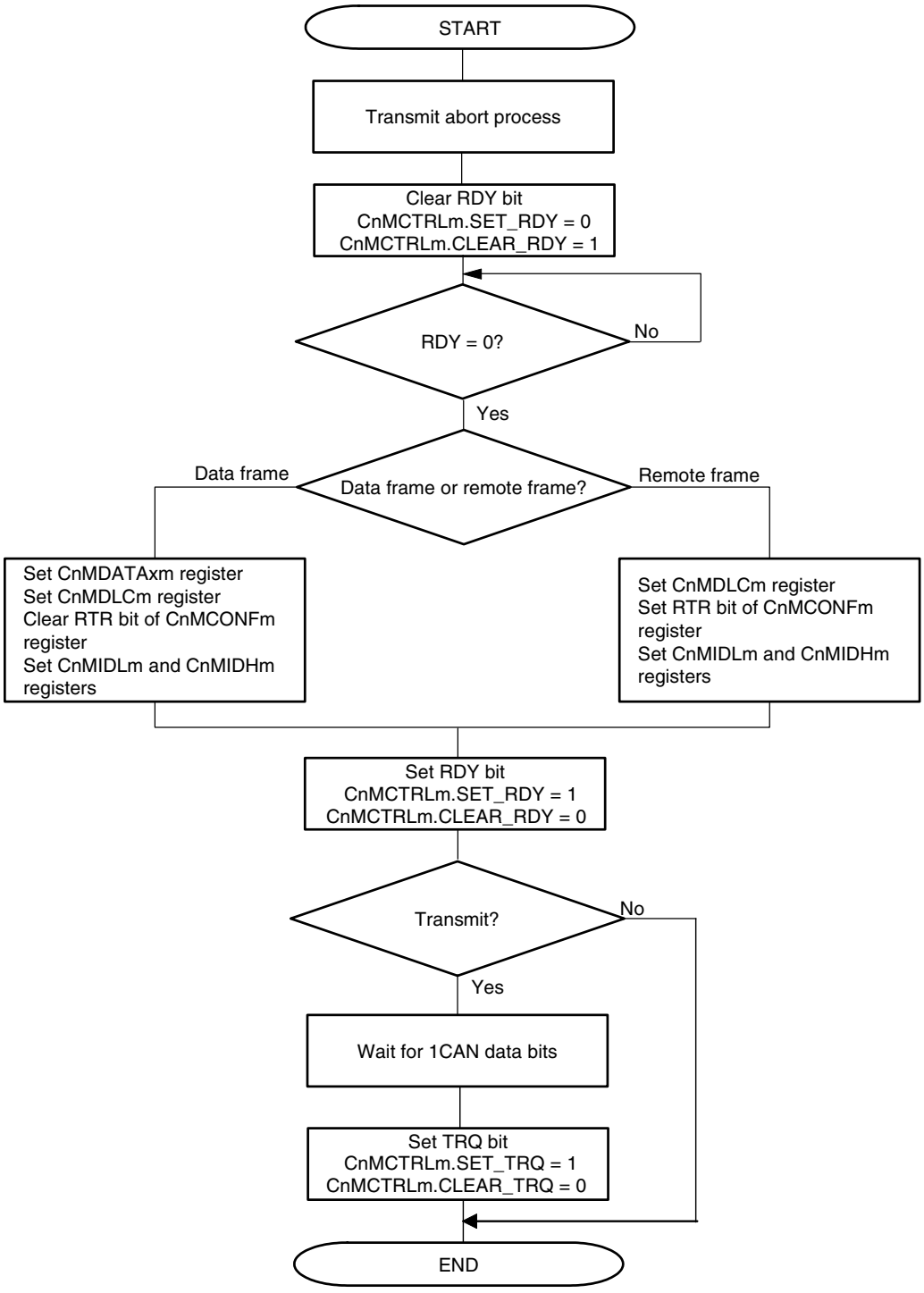
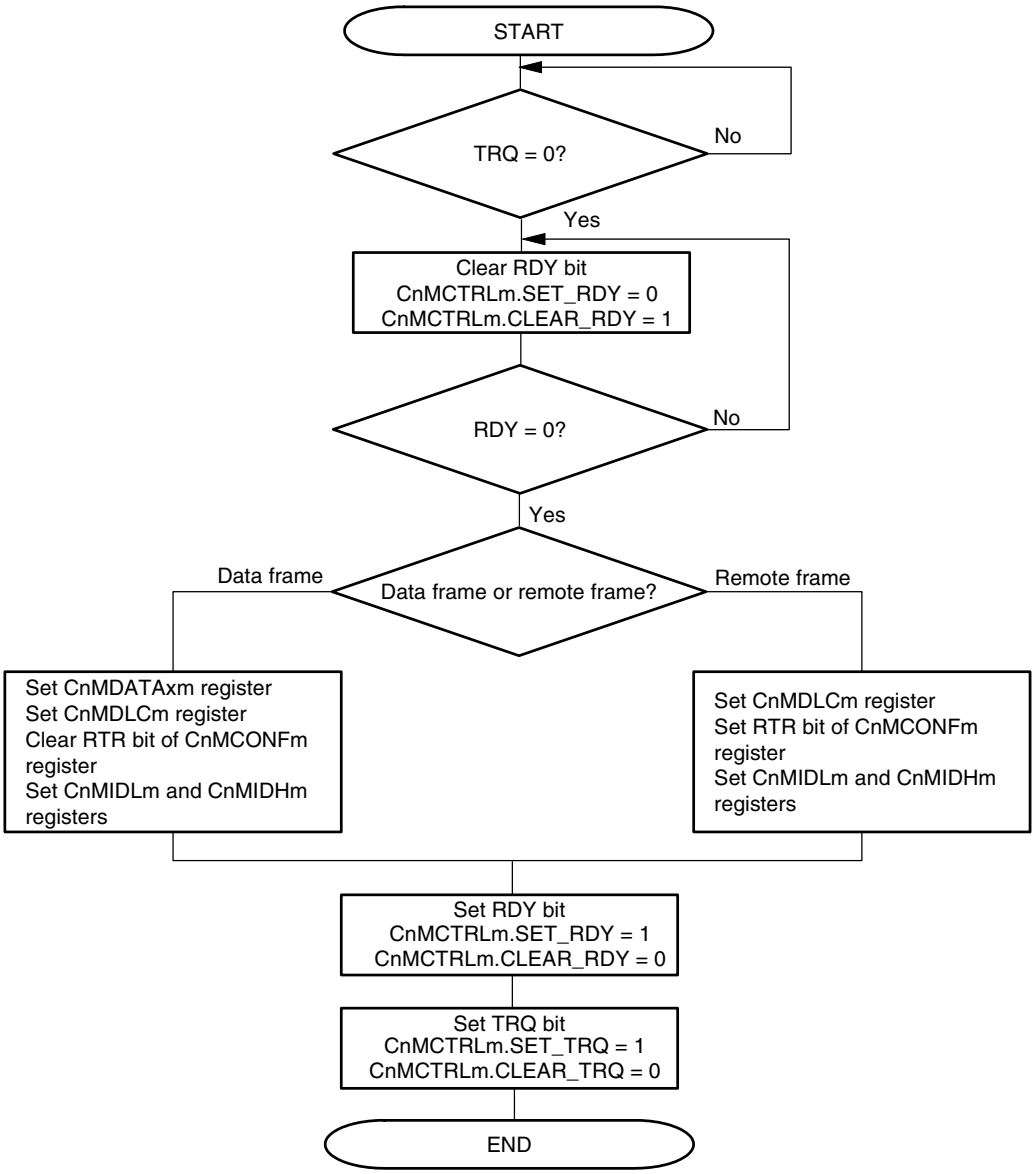


Figure 18-66 shows the processing for a transmit message buffer (MT2 to MT0 bits of CnMCONFm register = 000B).

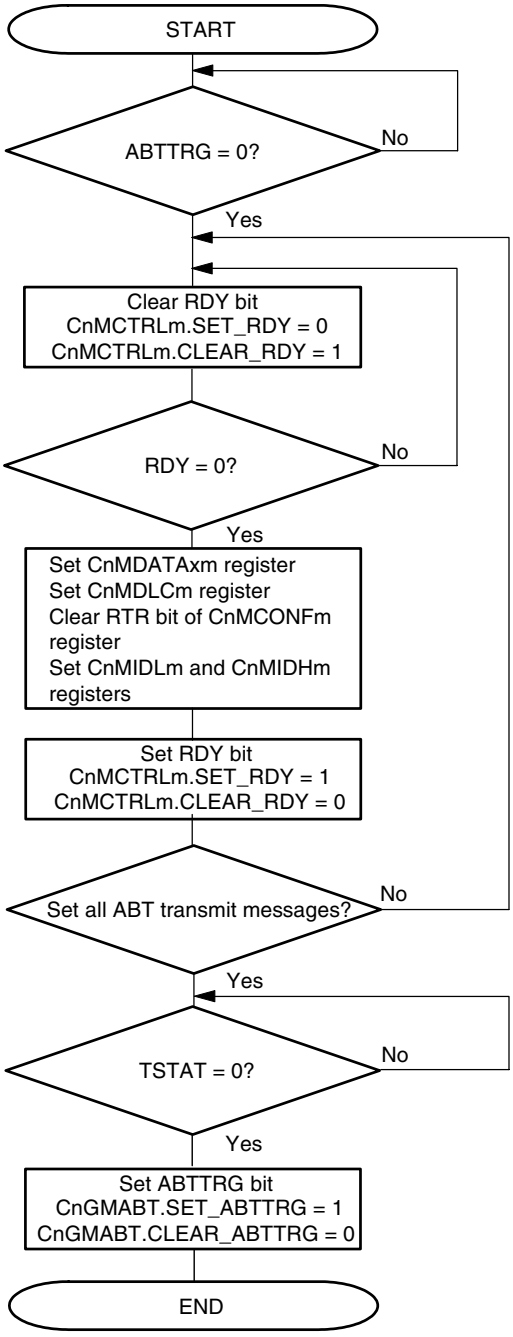
Figure 18-66: Message Transmit Processing



Caution: The TRQ bit should be set after the RDY bit is set.
The RDY bit and TRQ bit should not be set at the same time.

Figure 18-67 shows the processing for a transmit message buffer (MT2 to MT0 bits of CnMCONFm register = 000B).

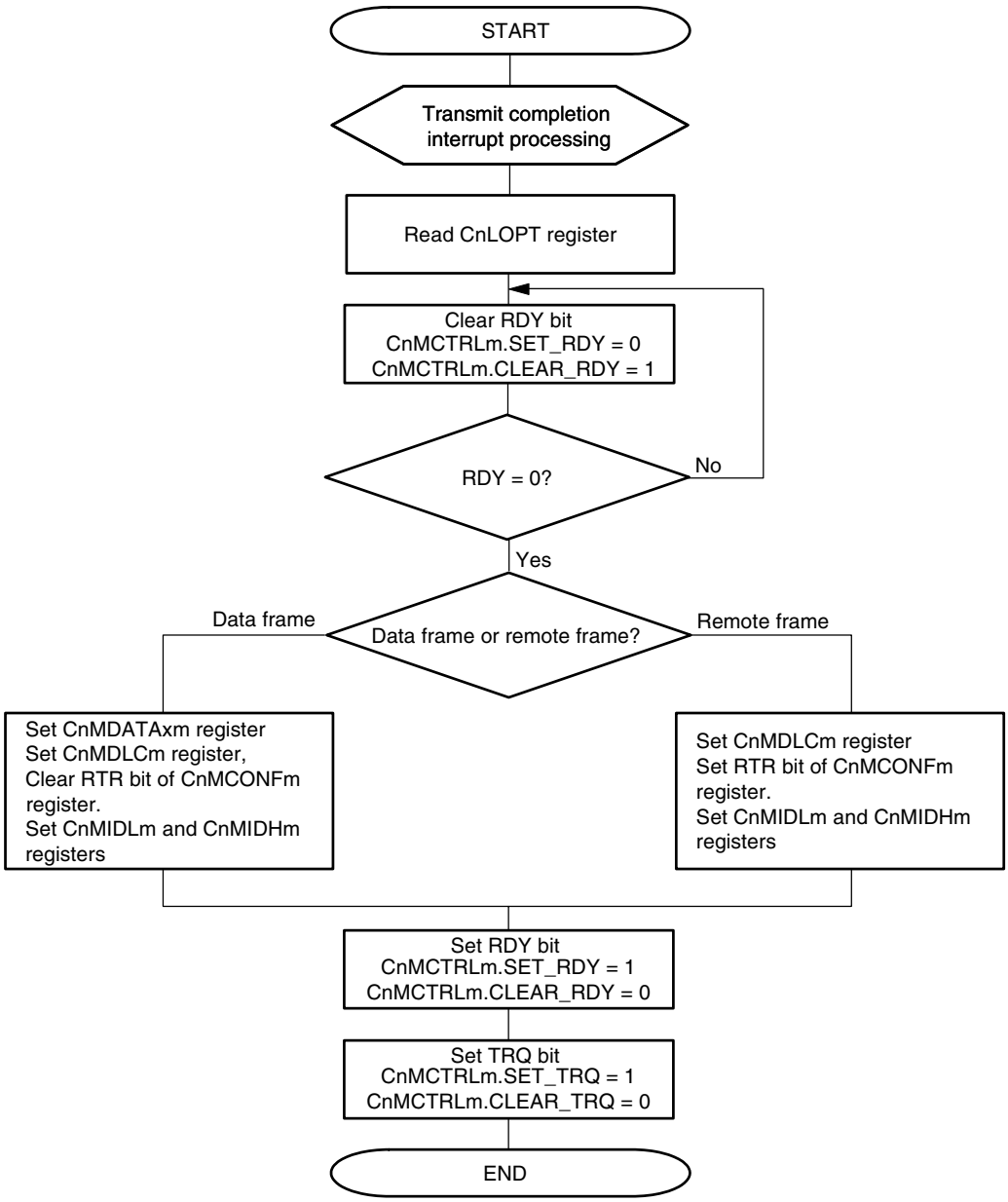
Figure 18-67: Message Transmit Processing (Normal Operation Mode with ABT)



Remark: This processing (normal operation mode with ABS) can only be applied to message buffers 0 to 7. For message buffers other than the ABT message buffers, refer to Figure 18-66.

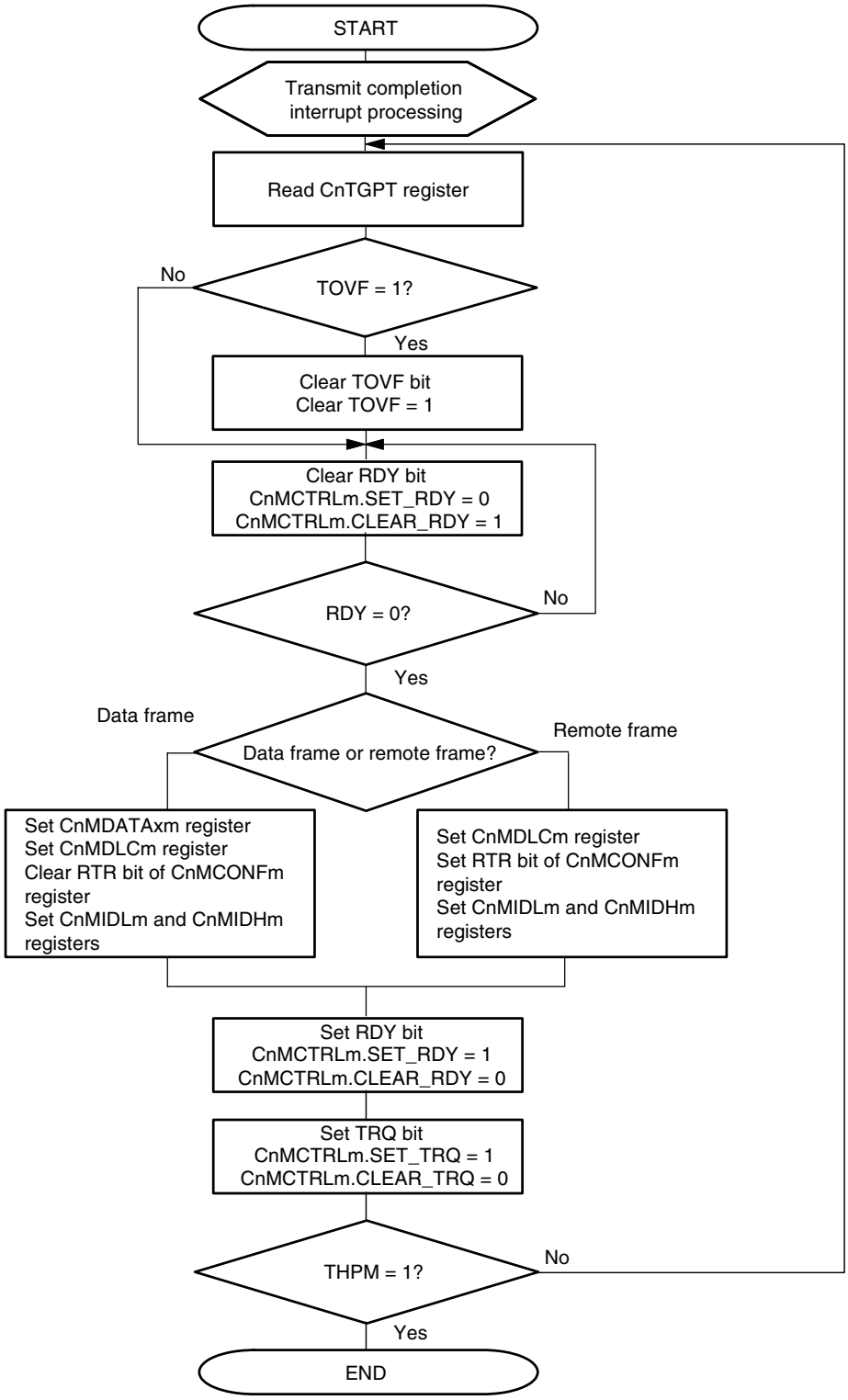
Caution: The ABTTRG bit should be set to 1 after the TSTAT bit is cleared to 0. Checking the TSTAT bit and setting the ABTTRG bit to 1 must be processed continuously.

Figure 18-68: Transmission via Interrupt (Using CnLOPT register)



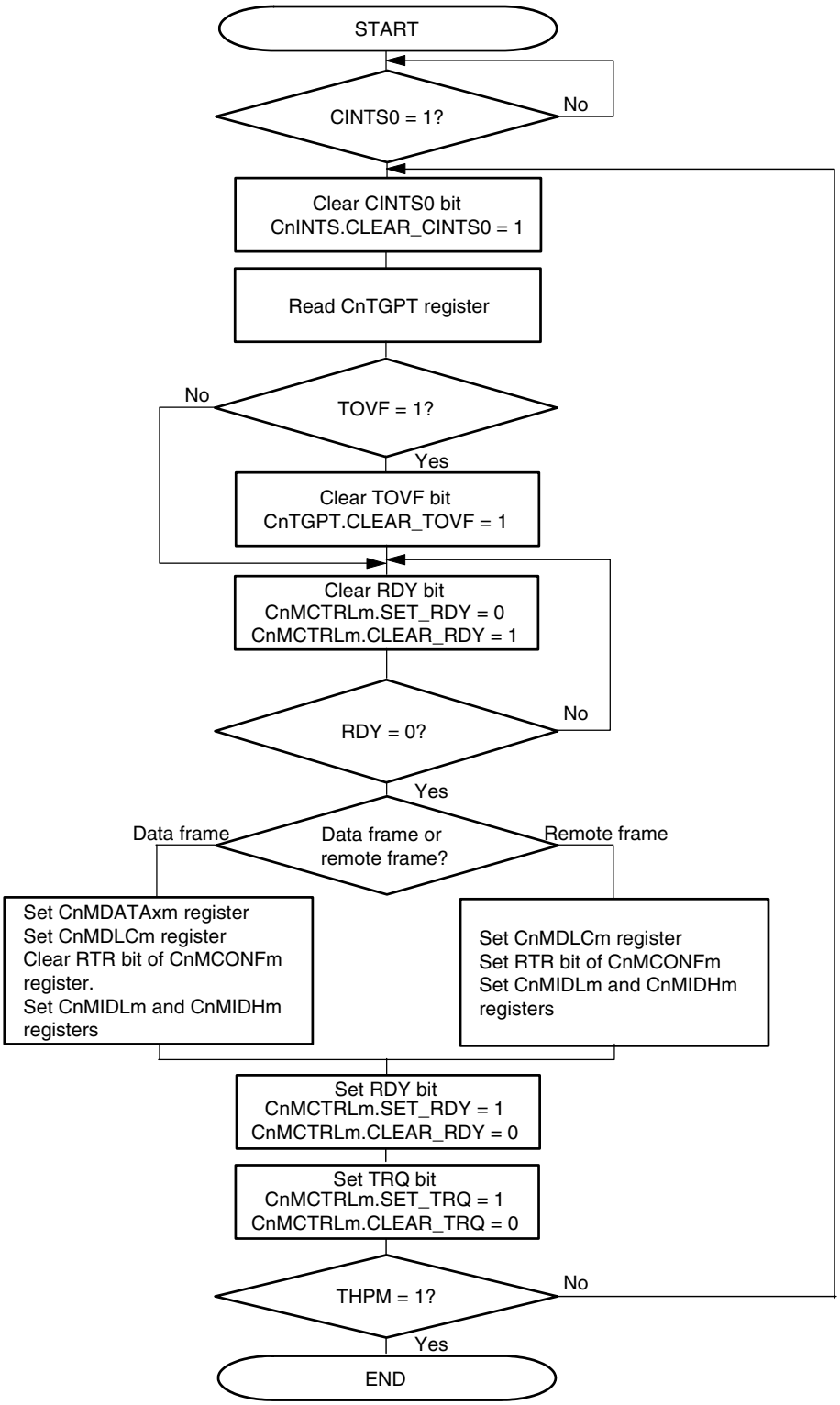
Caution: The TRQ bit should be set after the RDY bit is set.
The RDY bit and TRQ bit should not be set at the same time.

Figure 18-69: Transmit via Interrupt (Using CnTGPT register)



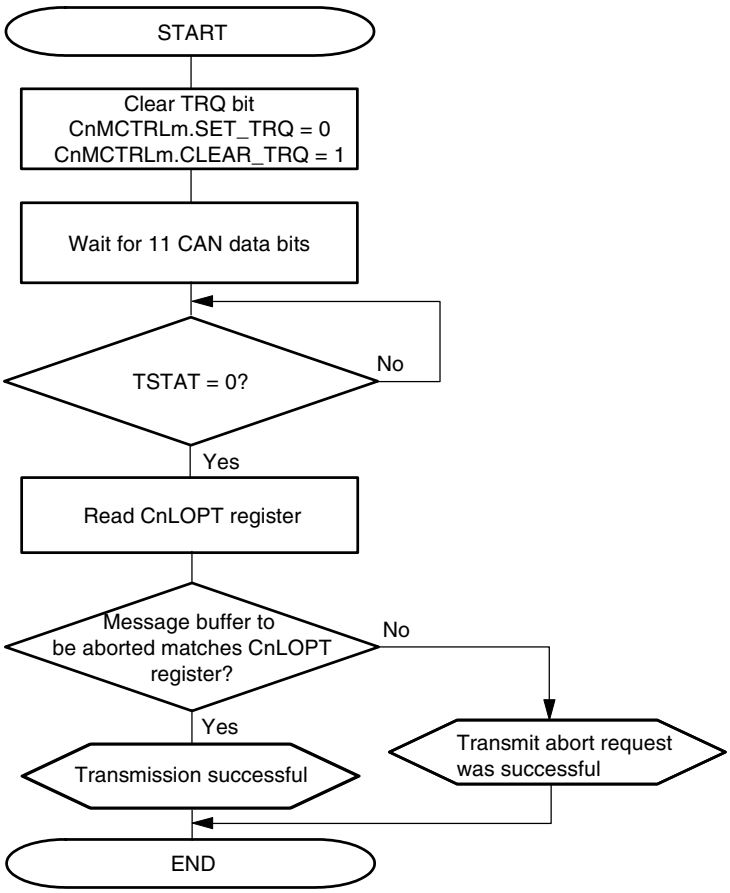
Caution: The TRQ bit should be set after the RDY bit is set.
The RDY bit and TRQ bit should not be set at the same time

Figure 18-70: Transmission via Software Polling



Caution: The TRQ bit should be set after the RDY bit is set.
The RDY bit and TRQ bit should not be set at the same time.

Figure 18-71: Transmission Abort Processing (Except Normal Operation Mode with ABT)



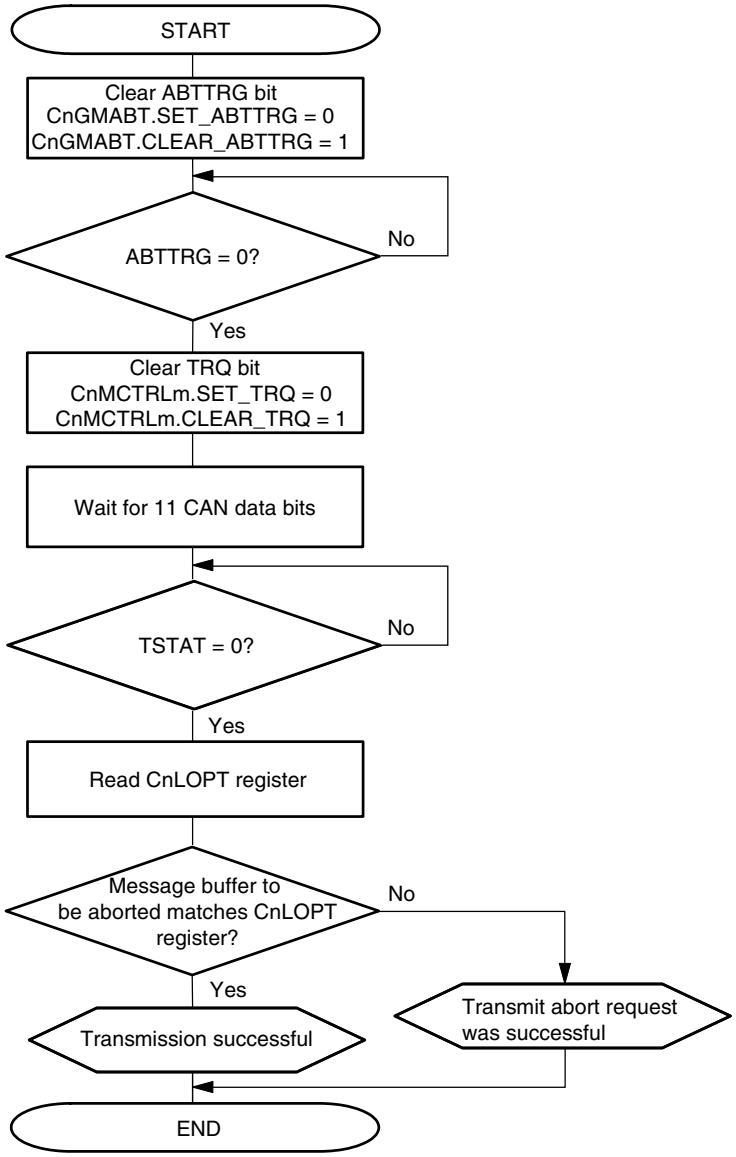
Cautions: 1. Execute transmission request abort processing by clearing the TRQ bit, not the RDY bit.

2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.

3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute the new transmission request including in the other message buffers while transmission abort processing is in progress.

Figure 18-72: Transmission Abort Processing Except for ABT Transmission (Normal Operation Mode with ABT)



Cautions: 1. Execute transmission request abort processing by clearing the TRQ bit, not the RDY bit.

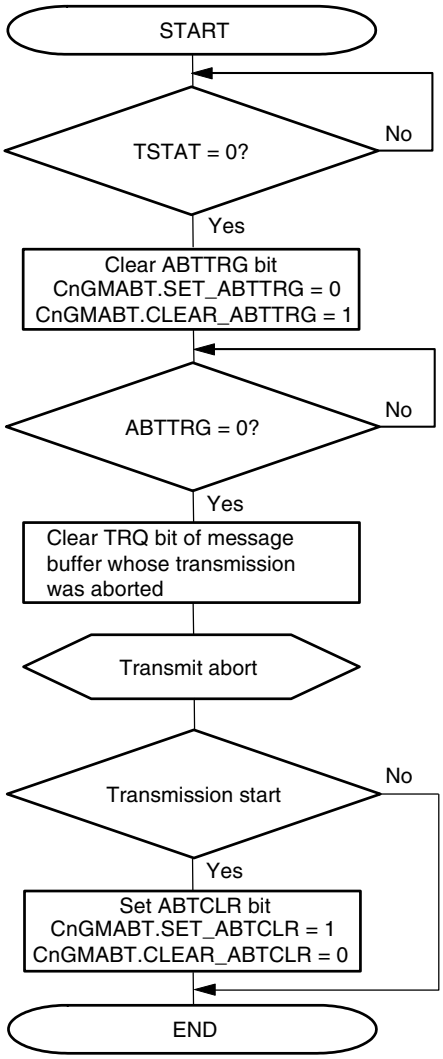
2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.

3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute the new transmission request including in the other message buffers while transmission abort processing is in progress.

Figure 18-73 shows the processing not to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

Figure 18-73: ABT Transmission Abort Processing (Normal Operation Mode with ABT)

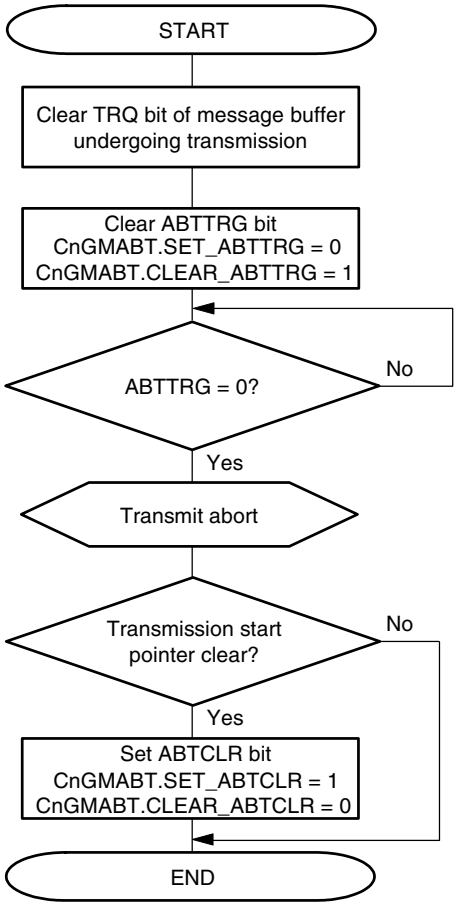


Cautions: 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. Make a CAN sleep mode/CAN stop mode transition request after ABTTRG is cleared (after ABT mode is aborted) following the procedure shown in Figure 18-73 or 18-74. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 18-71.

Figure 18-74 shows the processing to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

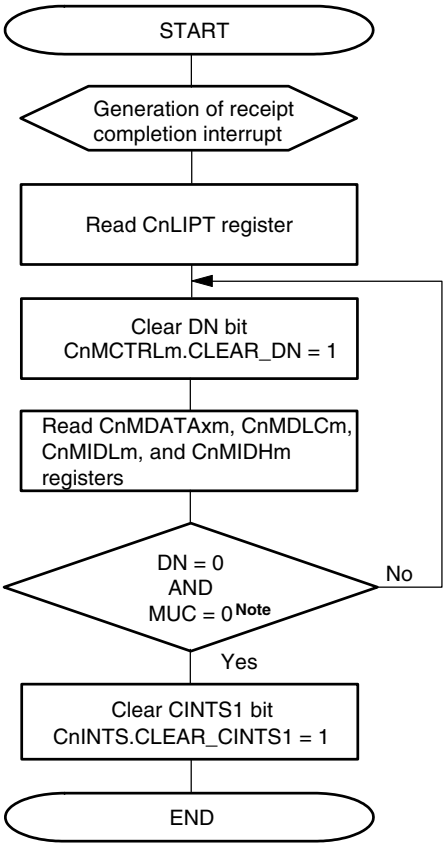
Figure 18-74: ABT Transmission Request Abort Processing (Normal Operation Mode with ABT)



Cautions: 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

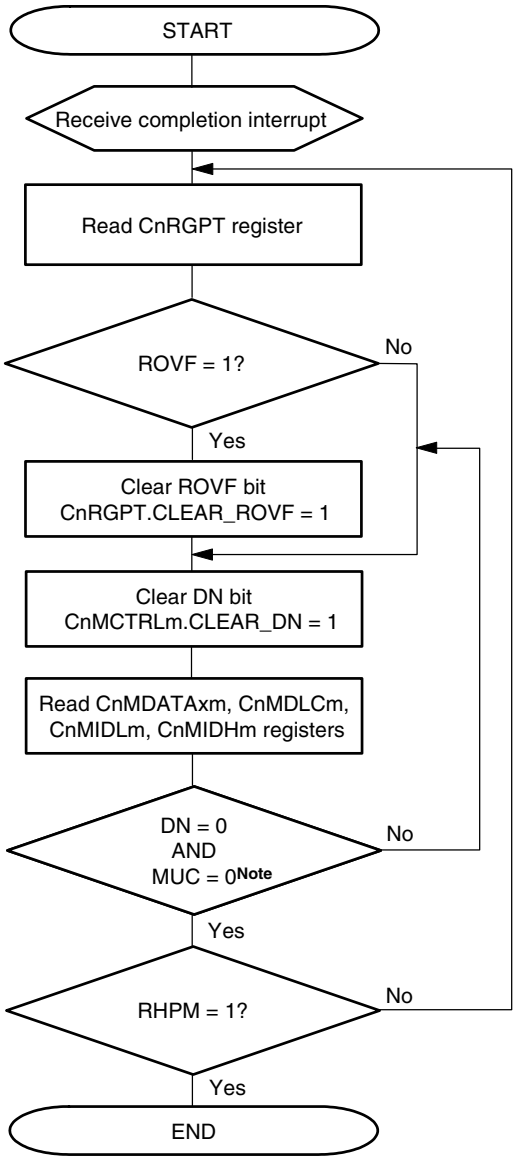
2. Make a CAN sleep mode/CAN stop mode request after ABTTRG is cleared (after ABT mode is stopped) following the procedure shown in Figures 18-73 or 18-74. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 18-71.

Figure 18-75: Reception via Interrupt (Using CnLIPT Register)



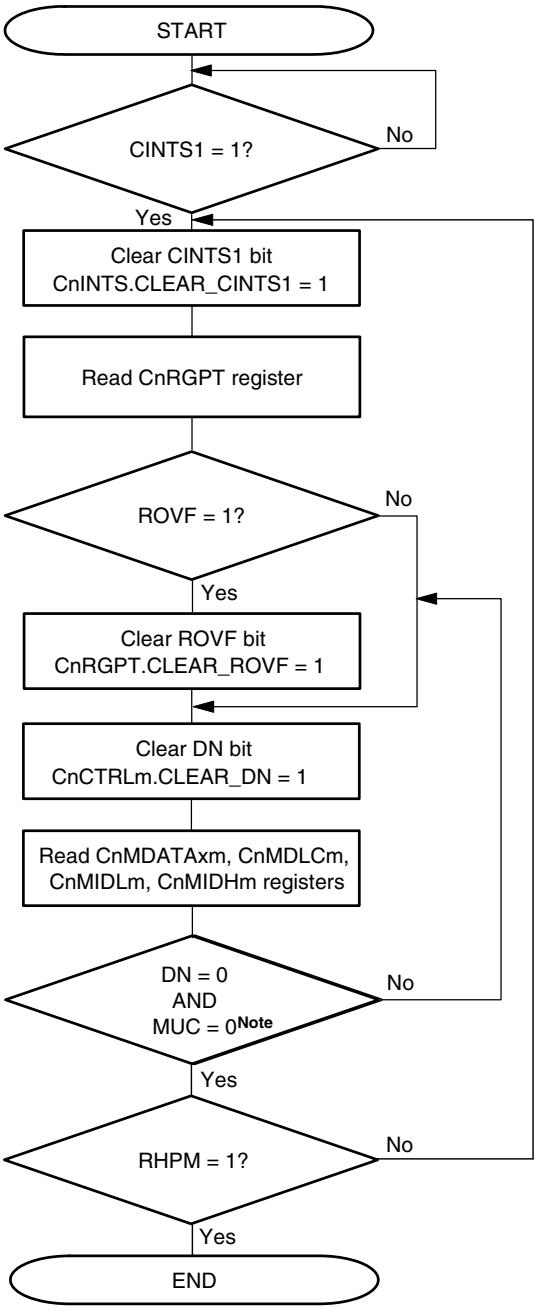
Note: Check the MUC and DN bits using one read access.

Figure 18-76: Reception via Interrupt (Using CnRGPT Register)



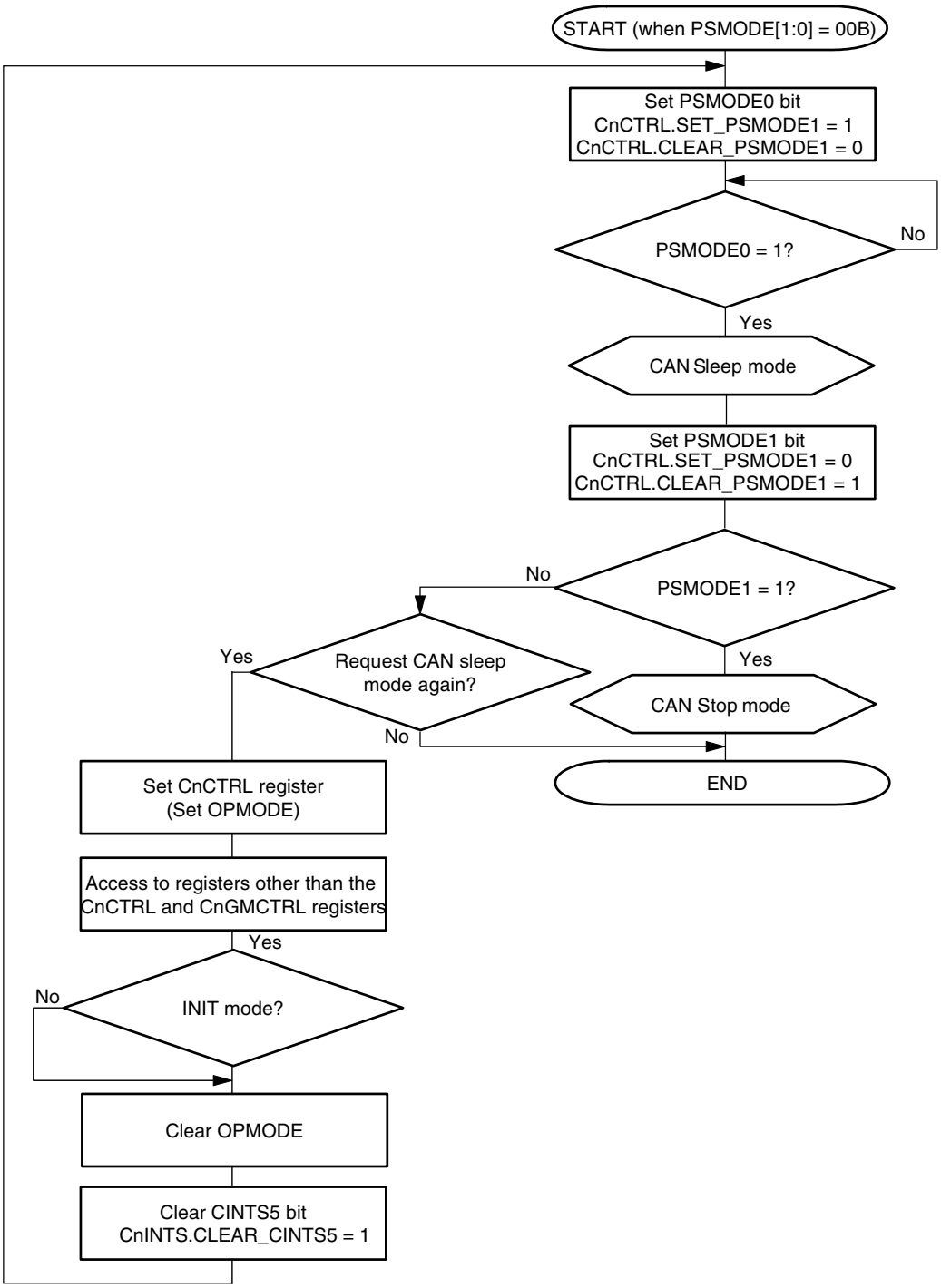
Note: Check the MUC and DN bits using one read access

Figure 18-77: Reception via Software Polling



Note: Check the MUC and DN bits using one read access

Figure 18-78: Setting CAN sleep Mode/Stop Mode



Cautions: 1. To abort transmission before making a request for the CAN sleep mode, perform processing according to Figures 18-71 and 18-72.

2. If the host CPU wants to enter a power save mode as well, the interrupt processing needs to be disabled before the CPU validates that sleep mode has been entered. If the interrupt processing can not be disabled, the host CPU will never wakeup by CAN bus activity when the CAN sleep mode is released between validation of the sleep state and execution of the i.e. CPU HALT instruction.

Figure 18-79: Clear CAN Sleep/Stop Mode

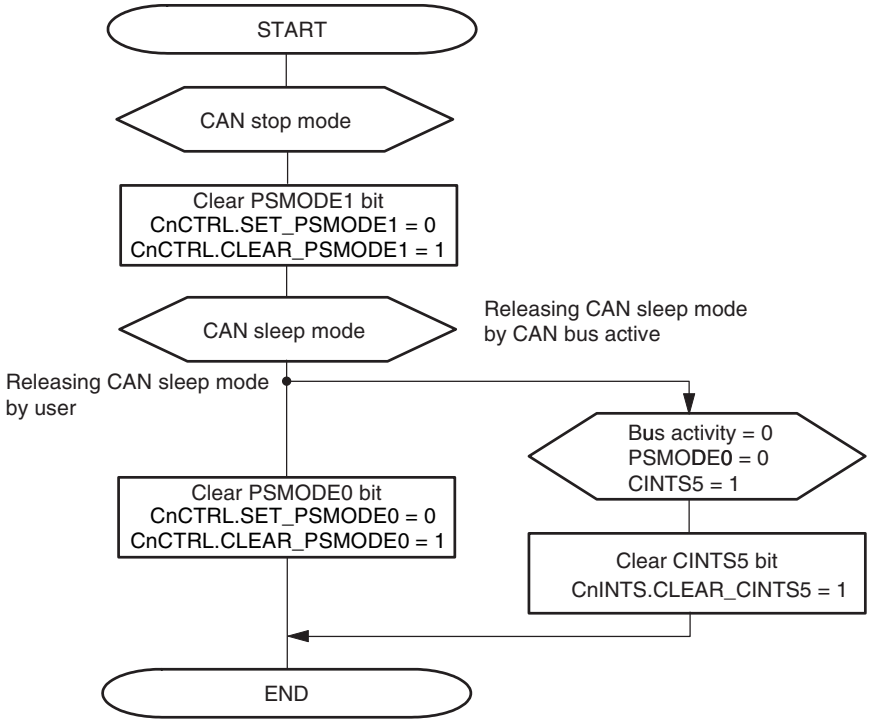


Figure 18-80: Bus-Off Recovery

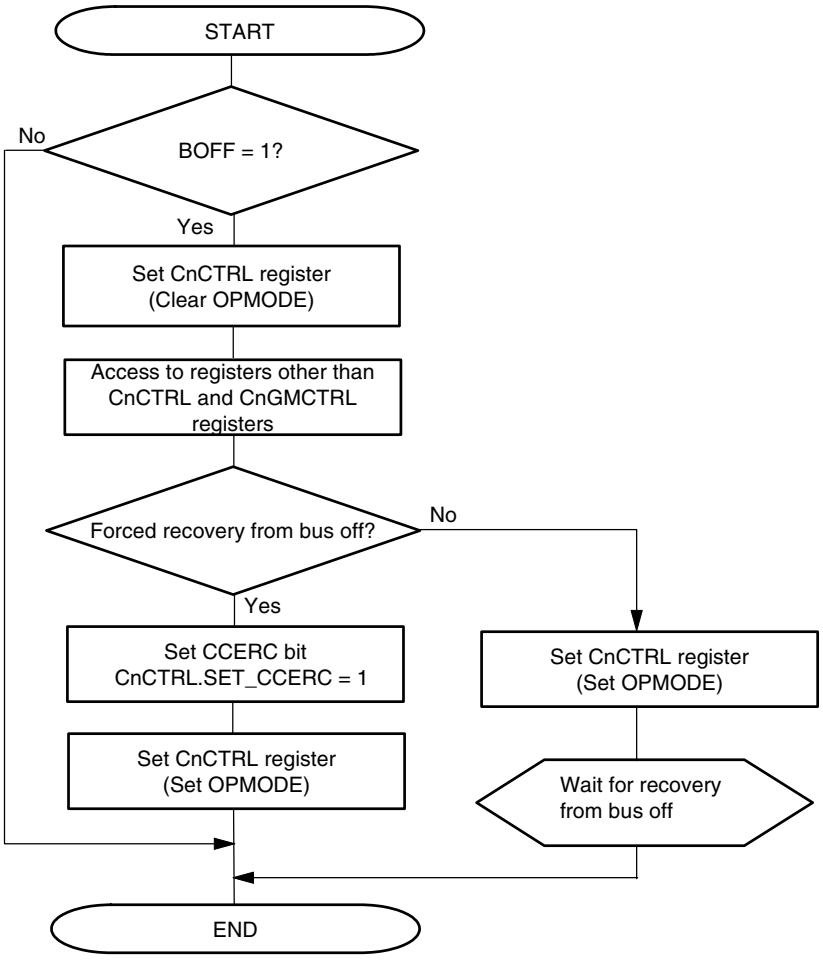


Figure 18-81: Normal Shutdown Process

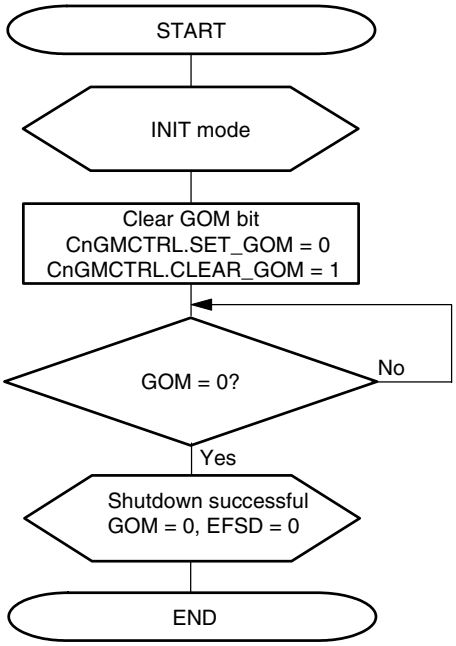
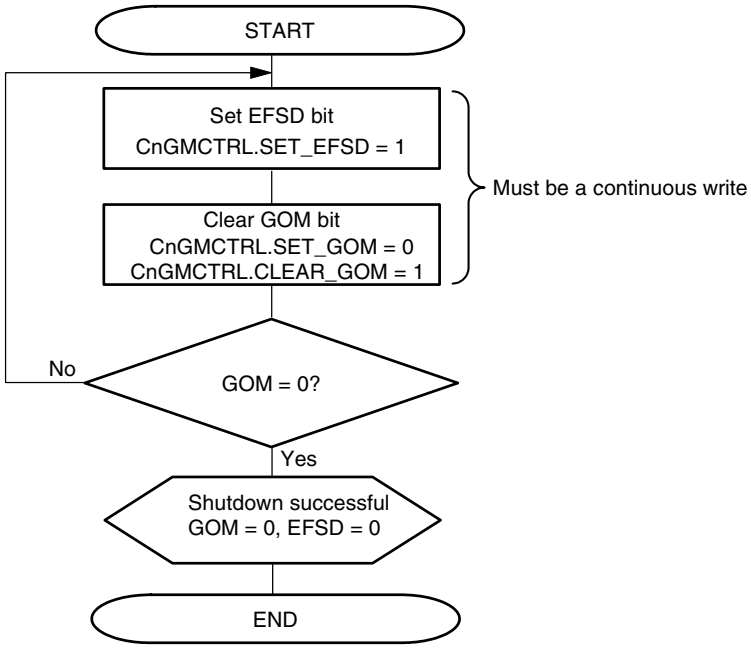


Figure 18-82: Forced Shutdown Process



Caution: Do not read- or write-access any registers by software between setting the EFSD bit and clearing the GOM bit.

Remark: OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 18-83: Error Handling

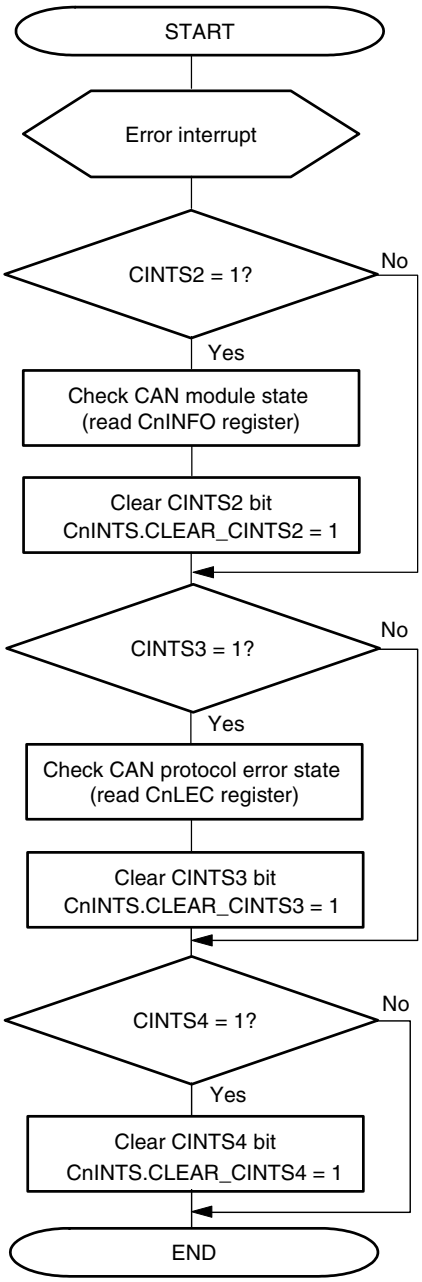


Figure 18-84: Setting CPU Standby (from CAN Sleep Mode)

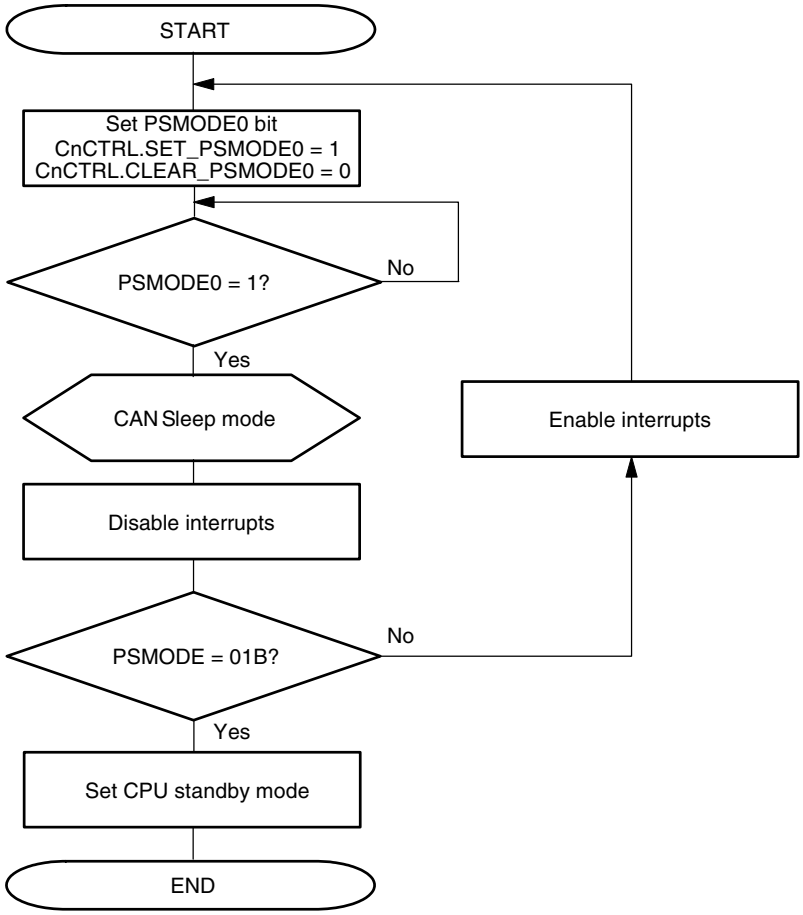
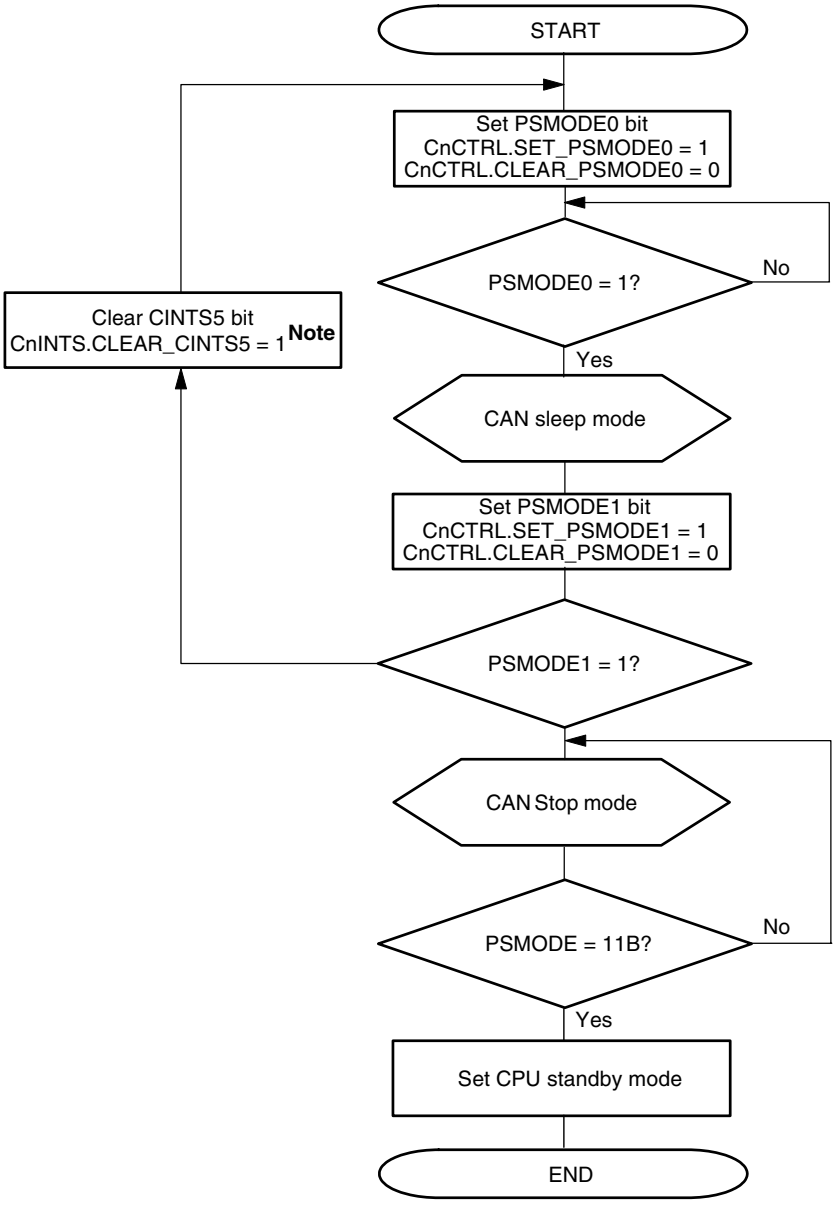


Figure 18-85: Setting CPU Standby (from CAN Stop Mode)



Note: During wakeup interrupts

Caution: The CAN stop mode can only be released by writing 01B to the PSMODE1, PSMODE0 bits of the CnCTRL register and not by a change in the CAN bus state.

Chapter 19 Random Number Generator

V850E/PH2 incorporates a hardware random number generator (RNG).

19.1 Features

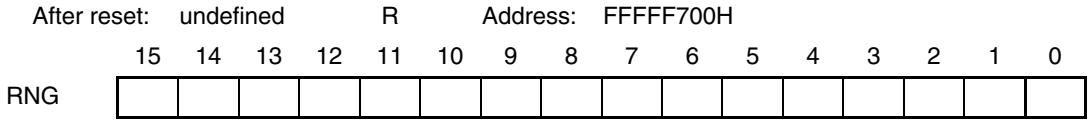
- Random number sequence passing FIPS and Maurer test
- Random number format: 16 bits
- Seed generated by hardware

19.2 Configuration

(1) Random number register (RNG)

The RNG register is a 16-bit register that holds the random number.
After read access to this register a certain time is required to generate the next random number. If a consecutive read access takes place before the new random number has been generated, the read access will be delayed.
The RNG register is read-only, in 16-bit units.
Reset input causes an undefined register content.

Figure 19-1: Random Number Register (RNG)



19.3 Operation

19.3.1 Access timing

After read access to the RNG register it needs a certain time to generate the next random number. Moreover, when a consecutive read access takes place before the new random number has been generated, the read access will be delayed.
The access timing to the RNG register is as follows.

Single read access to RNG register (when VSWC register = 13H):

$$T_{single} = 102{,}5 \cdot f_{XX}^{-1}$$

Consecutive read access to RNG register:

$$T_{consecutive} = T_{single} + (1024 \cdot f_{XX}^{-1})$$

Chapter 20 Port Functions

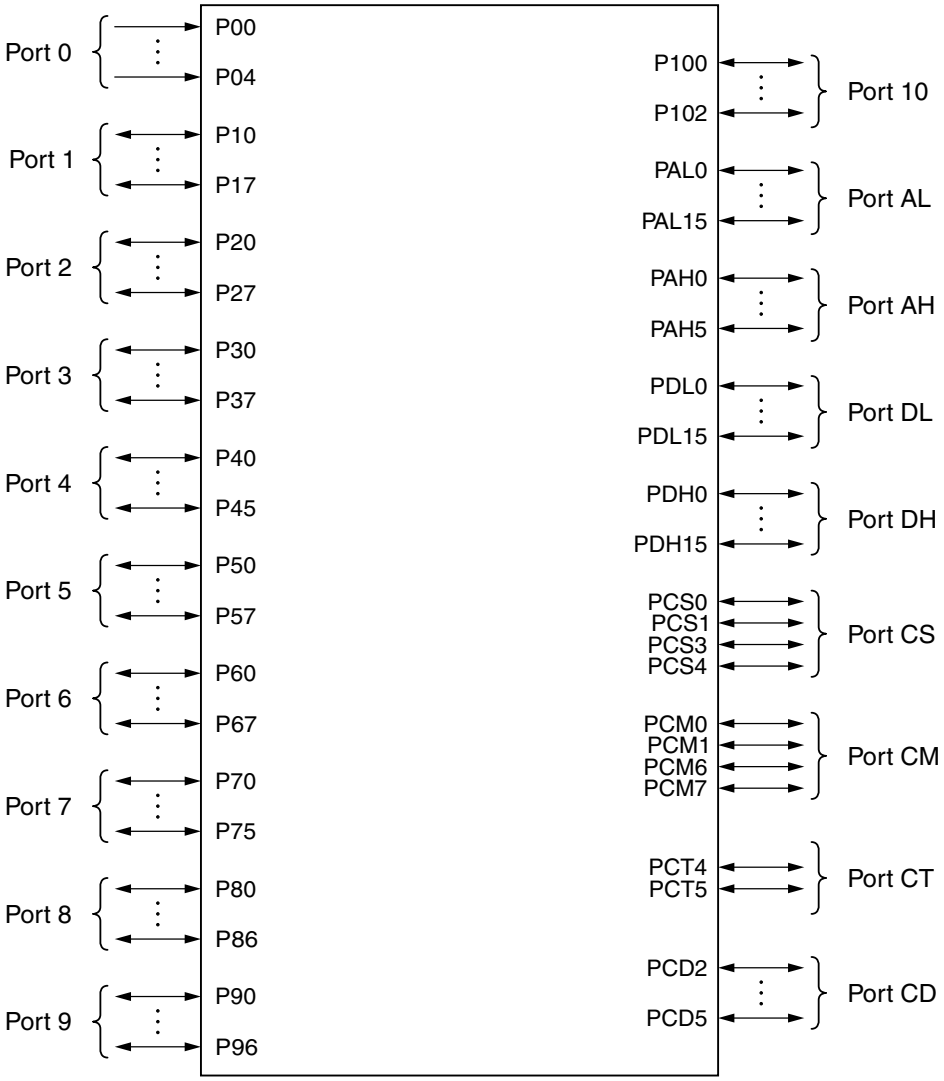
20.1 Features

- Input-only ports: 5
  I/O ports: 136
- Input/Output direction can be specified in 1.bit units
- Noise removal circuit provided for external interrupts and timer inputs
- Edge detect function for external interrupts (rising-, falling-, both edges)
- Security features for port 5 and 6 shared as 3-phase PWM timer outputs
  - Emergency shut off feature
  - Software protection feature

20.2 Port Configuration

The V850E/PH2 incorporates a total of 141 input/output ports (including 5 input-only ports) labelled port 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, AL, AH, DH, DL, CS, CM, CT, and CD. The port configuration is shown in Figure 20-1 below.

Figure 20-1: Port Configuration



20.2.1 Function of each port

The port functions of V850E/PH2 are shown in the table below.
The port type can vary for each individual bit of a port. In addition to their port functions, these pins are also shared with on-chip peripheral I/O pins in control mode. For the port types of each port, refer to Table 20-1 below.

Table 20-1: Port Type and Function Overview

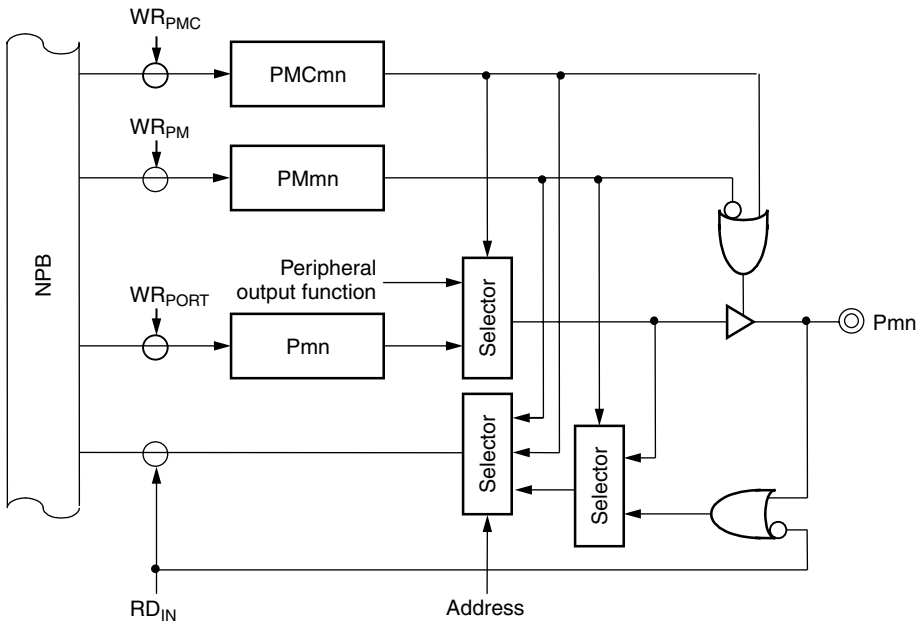
Port Name	Pin Name	Port Function	Function in Control Mode	Port Type
Port 0	P00 to P04	5-bit input only	External interrupt input External A/D conversion start trigger input Emergency shut-off input	3, 15, 15A
Port 1	P10 to P17	8-bit I/O port	Timer I/O (TMP0, TMP1, TMP2, TMP3)	6
Port 2	P20 to P27	8-bit I/O port	Timer I/O (TMP4, TMP5, TMP6, TMP7)	6
Port 3	P30 to P37	8-bit I/O port	Serial interface I/O (UARTC0, UARTC1, AFCAN0, AFCAN1)	1S, 2, 9
Port 4	P40 to P45	6-bit I/O port	Serial interface I/O (CSIB0, CSIB1)	1E, 2, 4C
Port 5	P50 to P57	8-bit I/O port	Timer output (TMR0)	11, 13
Port 6	P60 to P67	8-bit I/O port	Timer I/O (TMR1)	12, 13, 14
Port 7	P70 to P75	6-bit I/O port	Timer I/O (TMT0, TMT1)	6, 8
Port 8	P80 to P86	7-bit I/O port	Serial interface I/O (CSI30)	1S, 2, 4, 5, 7
Port 9	P90 to P96	7-bit I/O port	Serial interface I/O (CSI31)	1S, 2, 4, 5, 7
Port 10	P100 to P102	3-bit I/O port	Timer I/O (TENC1, TMP8, TMR0, TMR1)	6, 10
Port AL	PAL0 to PAL15	16-bit I/O port	External address bus (A0-A15)	1
Port AH	PAH0 to PAH5	6-bit I/O port	External address bus (A16-A21)	1
Port DL	PDL0 to PDL15	16-bit /IO port	External data bus (D0-D15)	4C
Port DH	PDH0 to PDH15	16-bit I/O port	External data bus (D16-D31)	4C
Port CS	PCS0, PCS1, PCS3, PCS4	4-bit I/O port	External bus interface control signal output ($\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$, $\overline{CS4}$)	1
Port CM	PCM0, PCM1, PCM6, PCM7	4-bit I/O port	External bus interface control signal I/O ($\overline{WAIT}$, BCLK, STST, STNXT)	1, 2C
Port CT	PCT4, PCT5	2-bit I/O port	External bus interface control signal output ($\overline{RD}$, $\overline{WR}$)	1
Port CD	PCD2 to PCD5	4-bit I/O port	External bus interface control signal output ($\overline{BEN0}$-$\overline{BEN3}$)	1

20.2.2 Port types

(1) Port type 1

Port type 1 provides a general purpose I/O port with peripheral output function.

Figure 20-2: Port Type 1

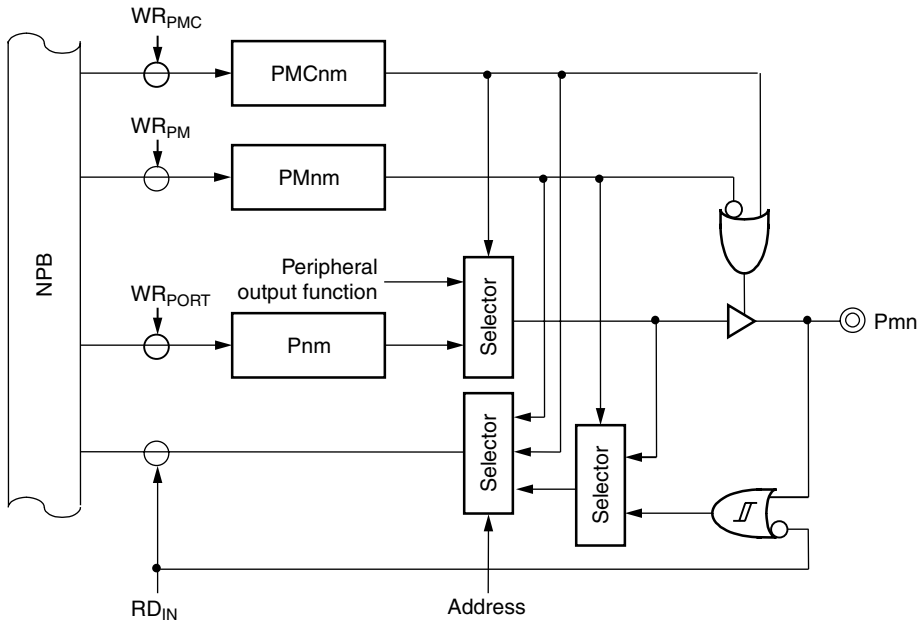


Remark: m: port number
n: port bit number

(2) Port type 1S

Port type 1S provides a general purpose I/O port with peripheral output function. This type is similar to port type 1, but features a Schmitt trigger input buffer characteristic.

Figure 20-3: Port Type 1S

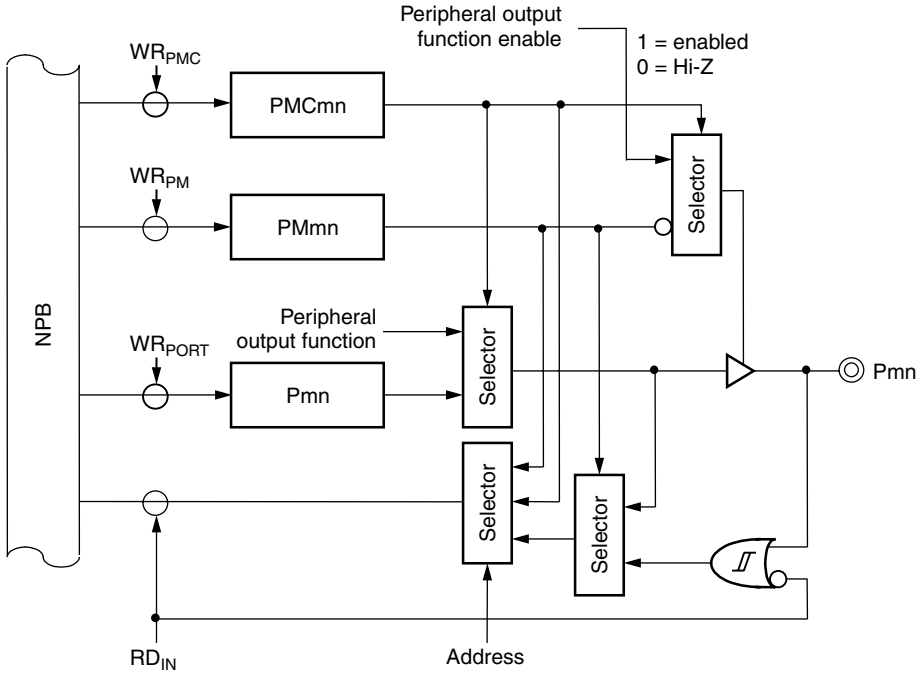


Remark: m: port number
n: port bit number

(3) Port type 1E

Port type 1E provides a general purpose I/O port with peripheral output function. In peripheral function mode a control signal is provided to enable or disable the output.

Figure 20-4: Port Type 1E

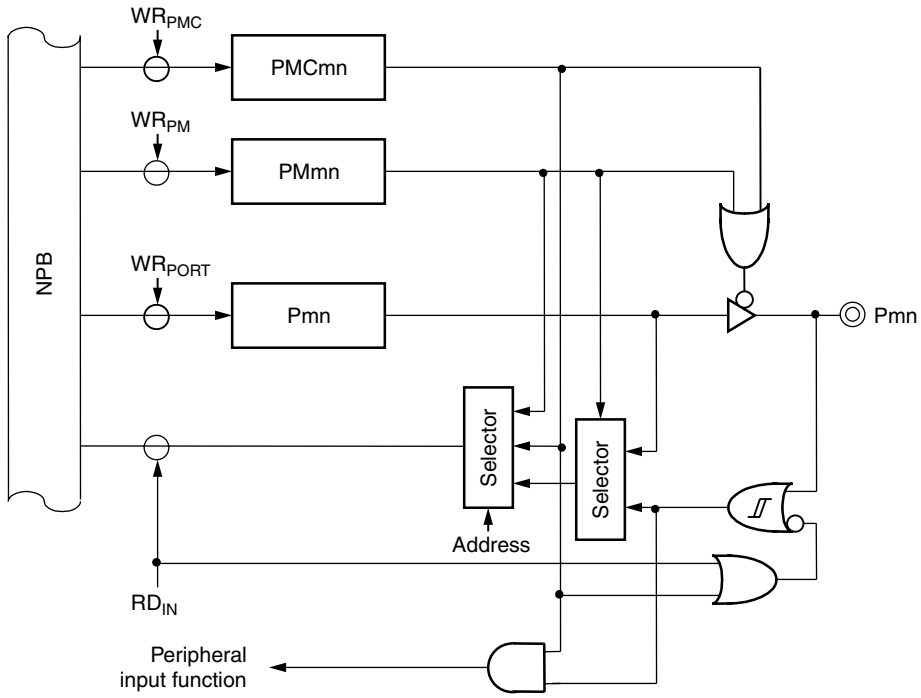


Remark: m: port number
n: port bit number

(4) Port type 2

Port type 2 provides a general purpose I/O port with peripheral input function.

Figure 20-5: Port Type 2

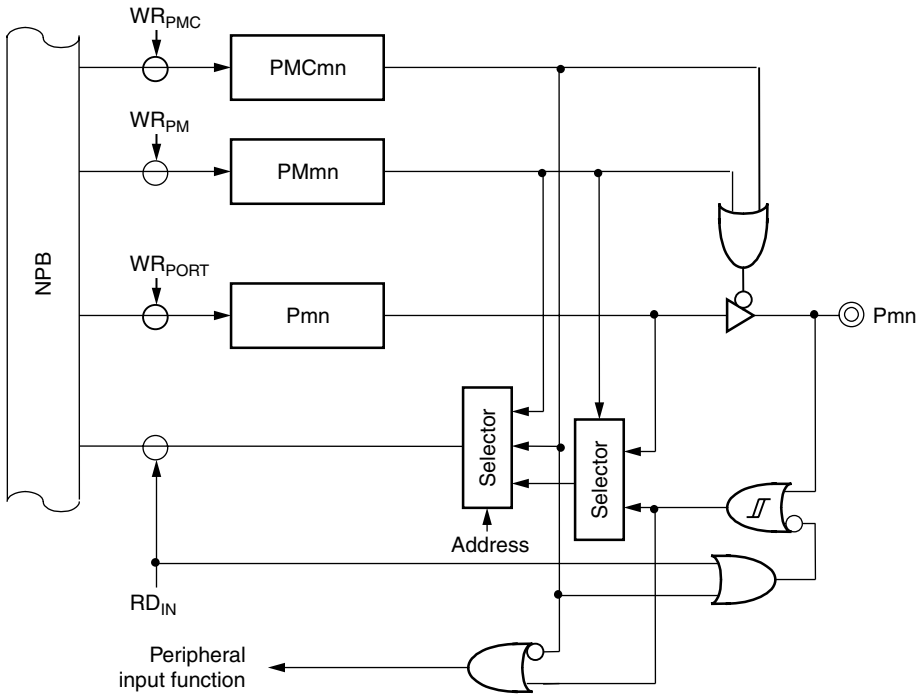


Remark: m: port number
n: port bit number

(5) Port type 2A

Port type 2A provides a general purpose I/O port with peripheral input function. This type is similar as port type 2, but in port mode the peripheral input function is forced to high level.

Figure 20-6: Port Type 2A

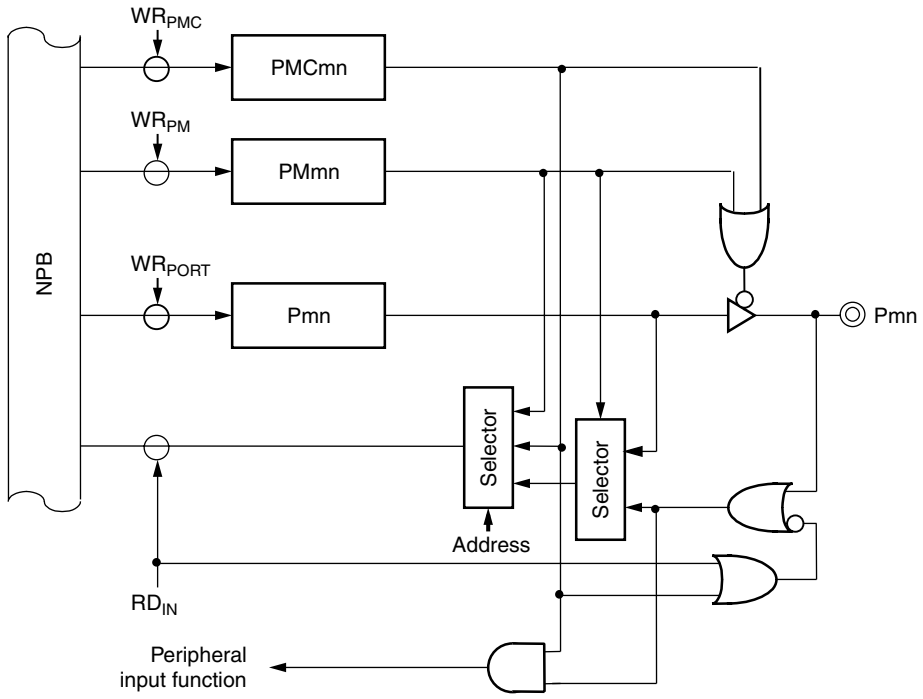


Remark: m: port number
n: port bit number

(6) Port type 2C

Port type 2C provides a general purpose I/O port with peripheral input function. This type is similar to type 2, but features CMOS input buffer characteristic.

Figure 20-7: Port Type 2C

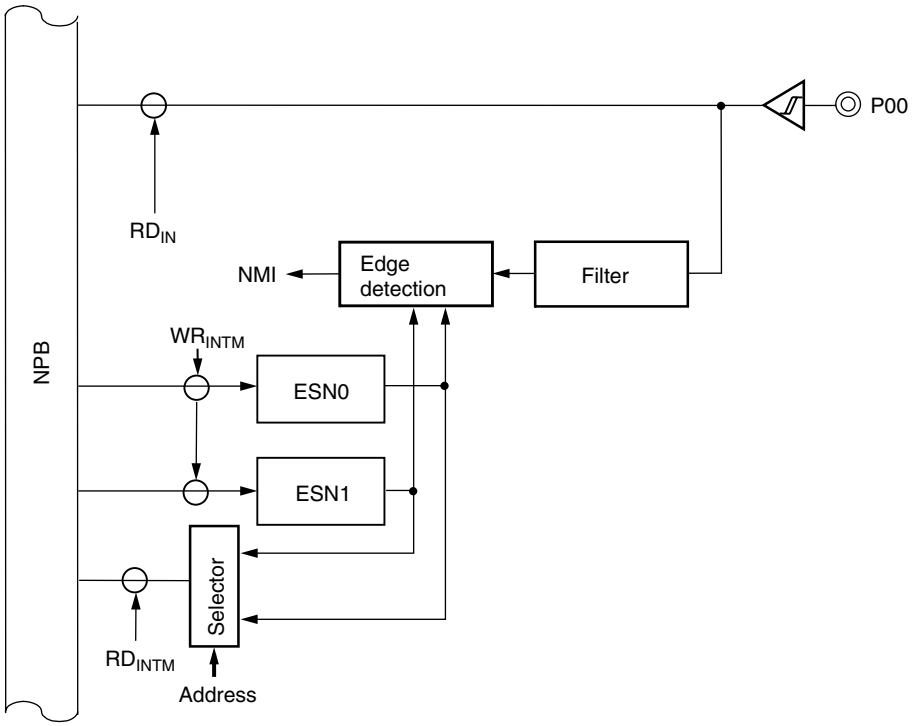


Remark: m: port number
n: port bit number

(7) Port type 3

Port type 3 provides a general purpose input port with NMI interrupt input function.

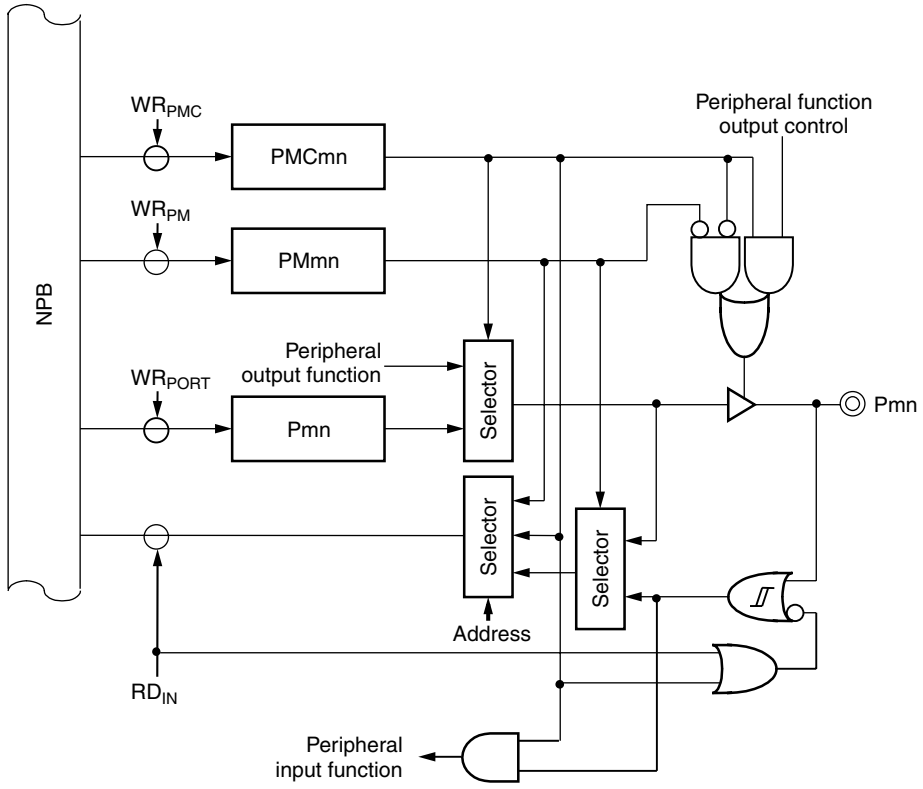
Figure 20-8: Port Type 3



(8) Port type 4

Port type 4 provides a general purpose I/O port with peripheral I/O function. Peripheral output enable is controlled by the corresponding peripheral function.

Figure 20-9: Port Type 4

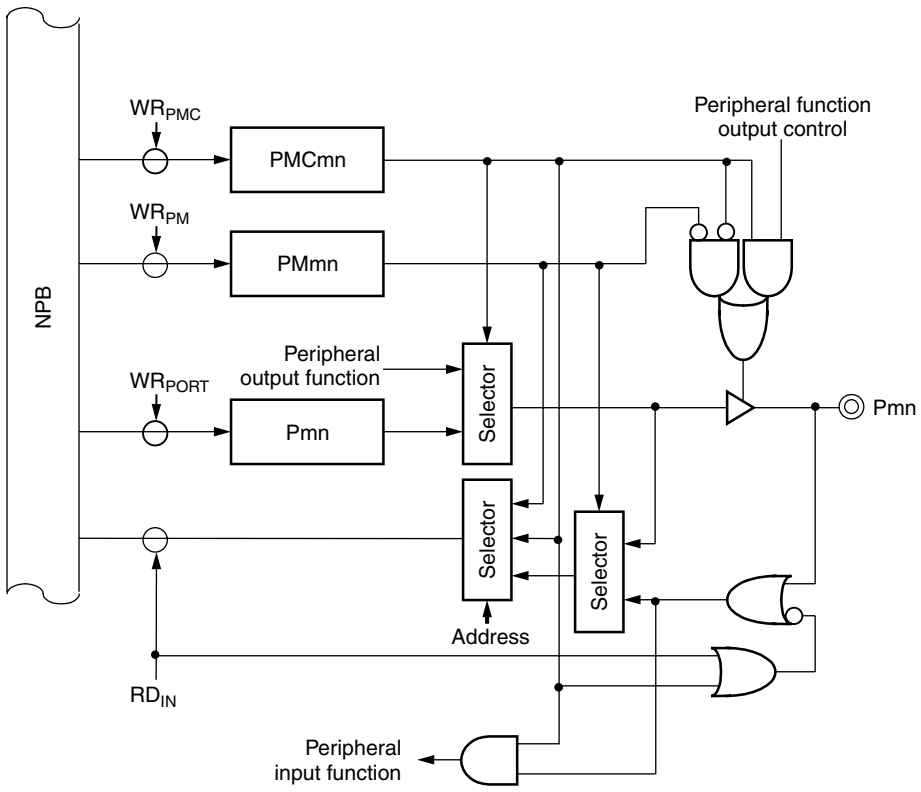


Remark: m: port number
n: port bit number

(9) Port type 4C

Port type 4 provides a general purpose I/O port with peripheral I/O function. Peripheral output enable is controlled by the corresponding peripheral function.

Figure 20-10: Port Type 4C

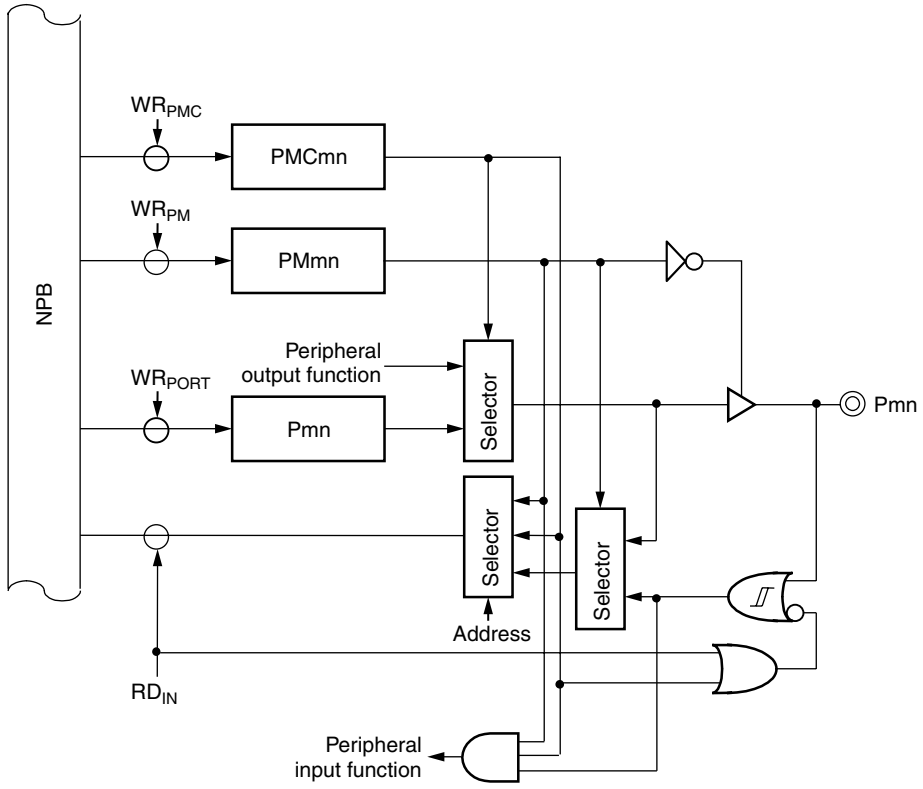


Remark: m: port number
n: port bit number

(10) Port type 5

Port type 5 provides a general purpose I/O port with peripheral I/O function. If the peripheral input function is disabled, the value of the peripheral input signal is fixed to low level.

Figure 20-11: Port Type 5

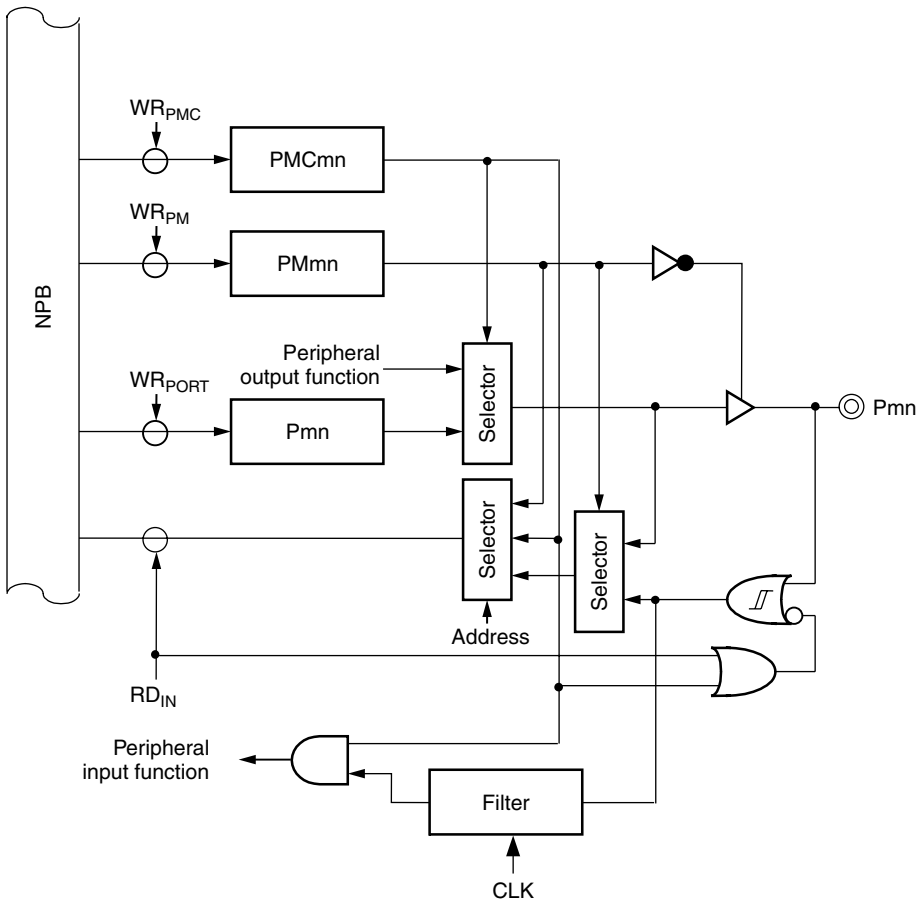


Remark: m: port number
n: port bit number

(11) Port type 6

Port type 6 provides a general purpose I/O port with peripheral output function and digitally filtered peripheral input function.

Figure 20-12: Port Type 6

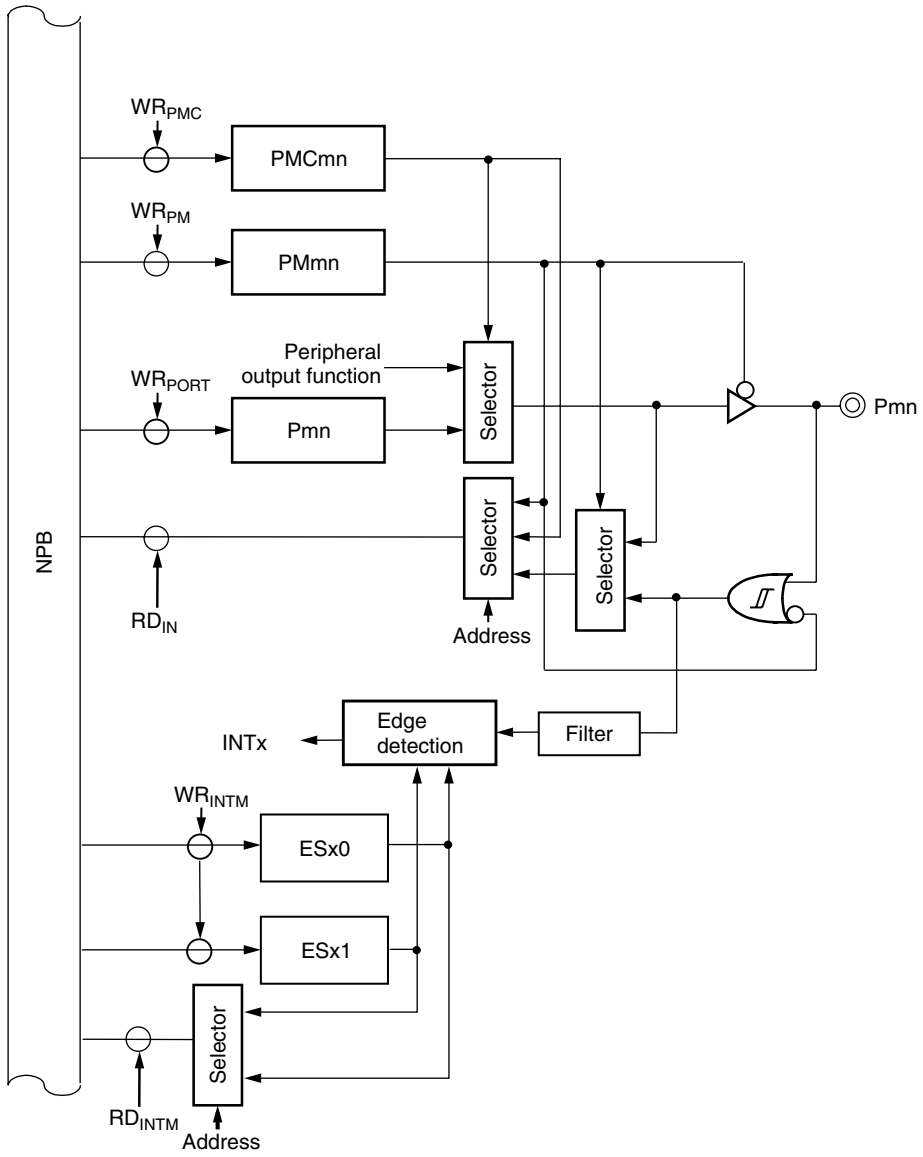


Remark: m: port number
n: port bit number

(12) Port type 7

Port type 7 provides a general purpose I/O port with peripheral output function and external interrupt input capability.

Figure 20-13: Port Type 7

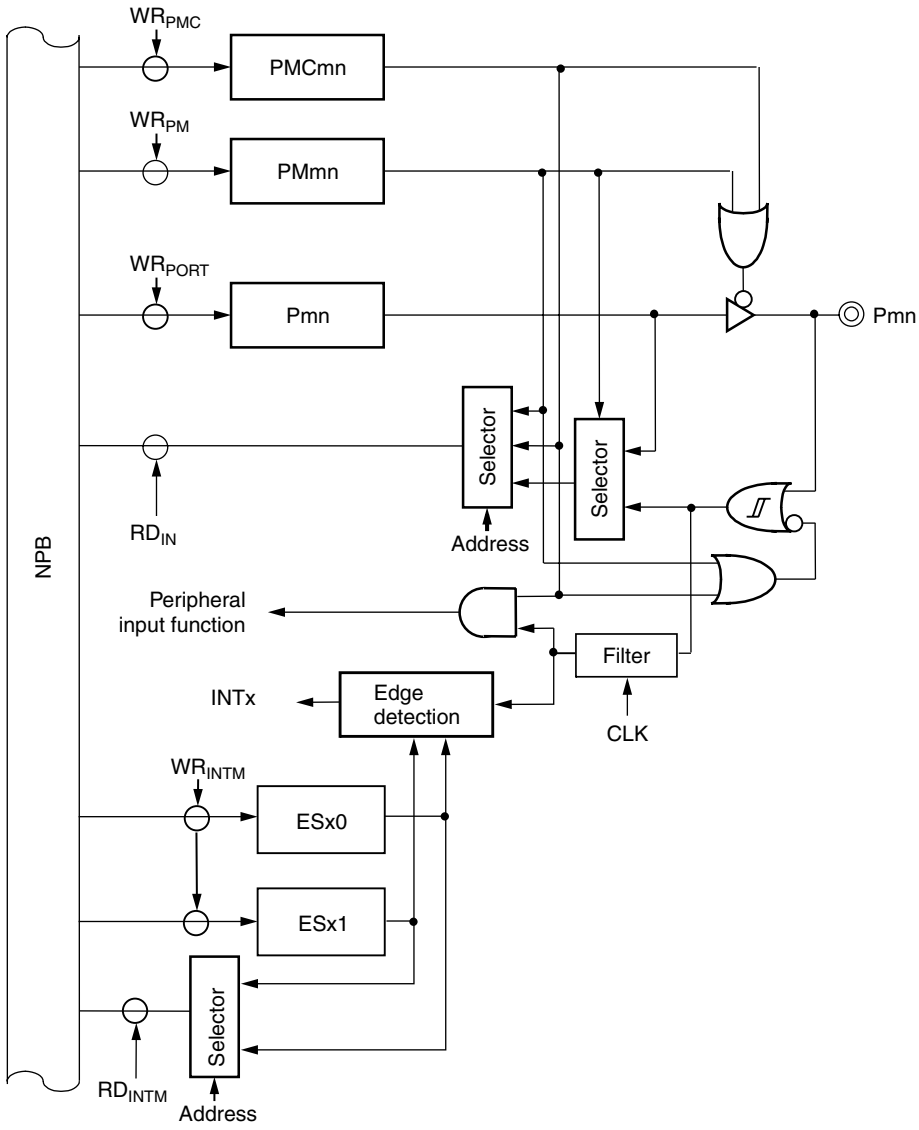


Remark: m: port number
n: port bit number
x: external interrupt number

(13) Port type 8

Port type 8 provides a general purpose I/O port with digitally filtered peripheral input function and external interrupt input capability.

Figure 20-14: Port Type 8



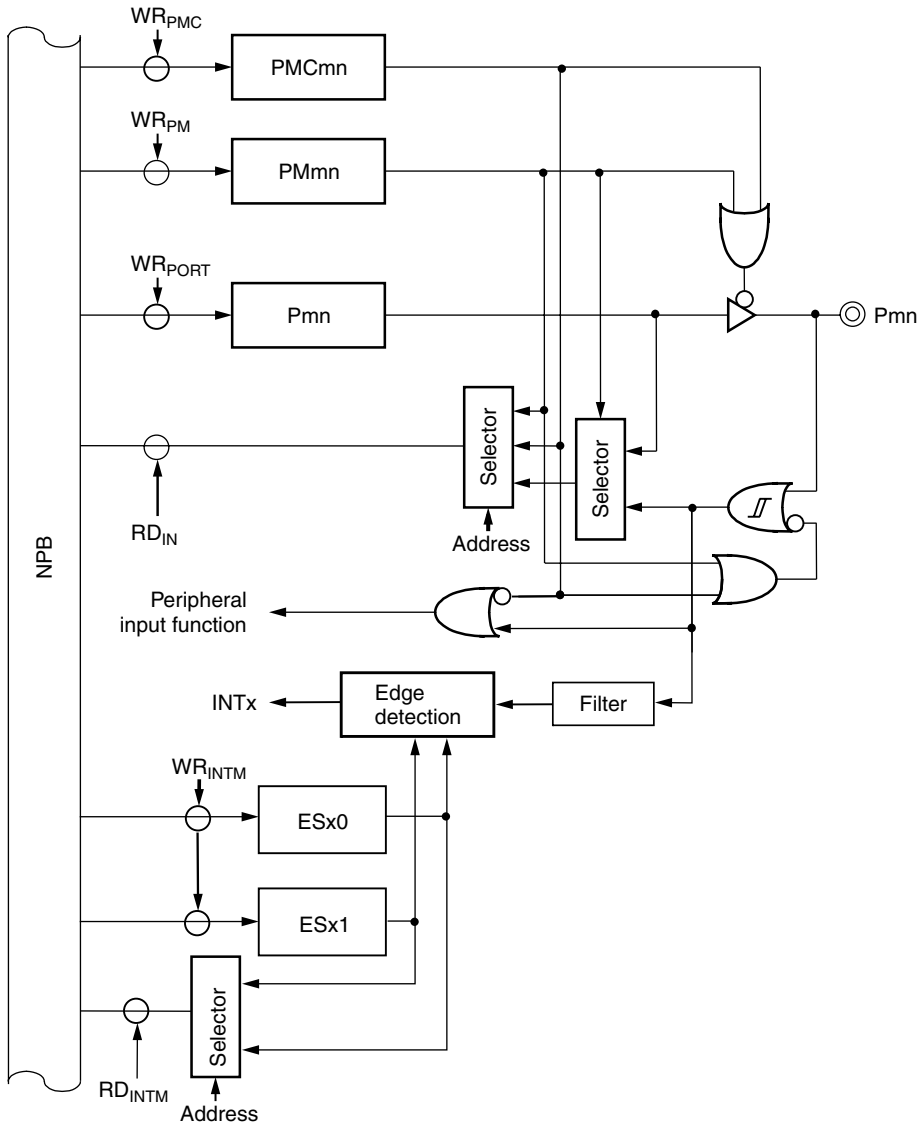
Remark: m: port number
n: port bit number
x: external interrupt number

(14) Port type 9

Port type 9 provides a general purpose I/O port with peripheral input function and external interrupt input capability. This type is similar to the port type 8, but input noise filter is bypassed for peripheral input function.

Remark: The peripheral input signal provided by port type 9 is fixed to high level, if peripheral input function is disabled.

Figure 20-15: Port Type 9

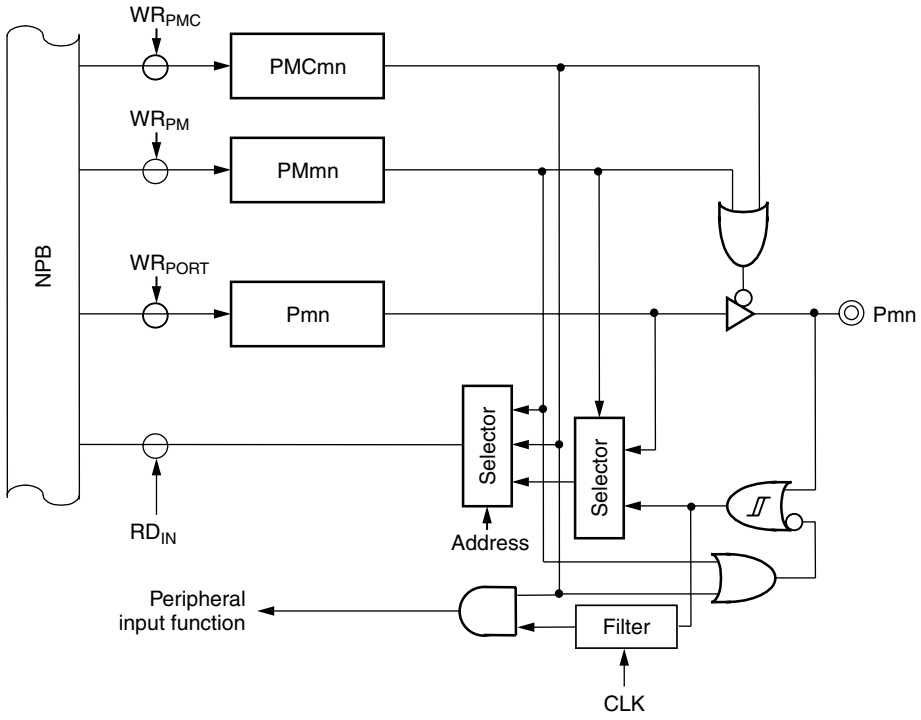


Remark: m: port number
n: port bit number
x: external interrupt number

(15) Port type 10

Port type10 provides a general purpose I/O port with digitally filtered peripheral input function.

Figure 20-16: Port Type 10

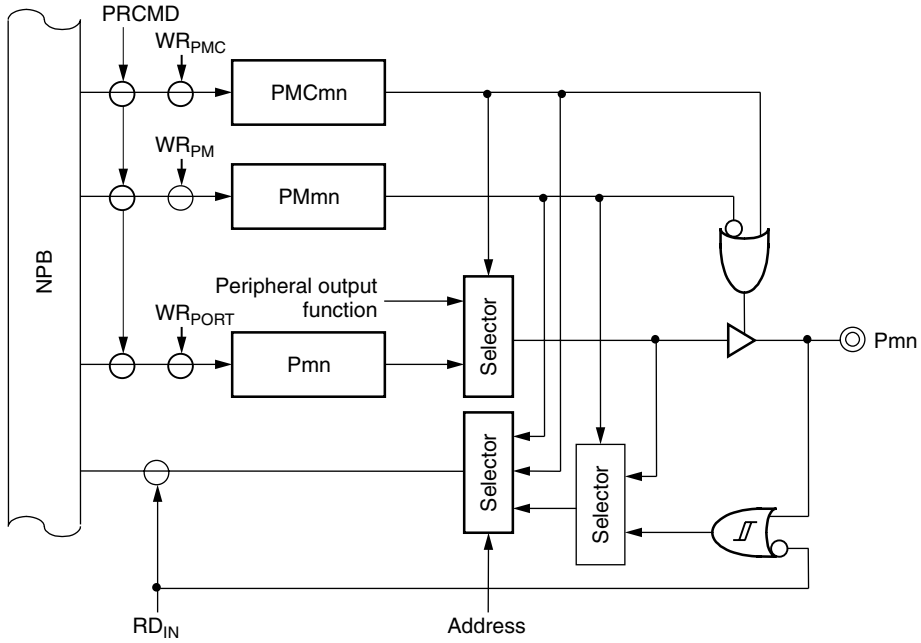


Remark: m: port number
n: port bit number

(16) Port type 11

Port type 11 provides a general purpose I/O port with peripheral output function. This type is similar to the port type 6, but all port registers are write protected against unintended change due to system or software malfunction. Writing to the port registers of type 11 is only possible immediately after a write access to the PRCMD register.

Figure 20-17: Port Type 11

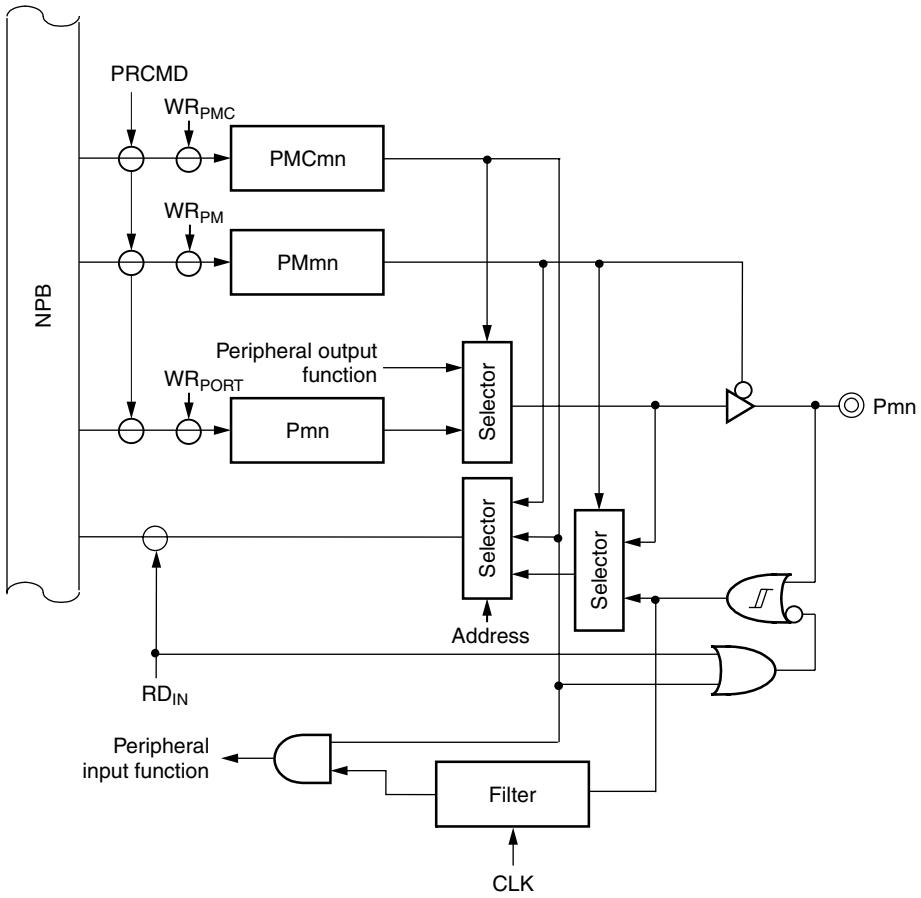


Remark: m: port number
n: port bit number

(17) Port type 12

Port type 12 provides a general purpose I/O port with digitally filtered peripheral input function and peripheral output function. This type is similar to the port logic type 1S, but all port registers are write protected against unintended change due to system or software malfunction. Writing to the port registers of type 12 is only possible immediately after a write access to the PRCMD register.

Figure 20-18: Port Type 12

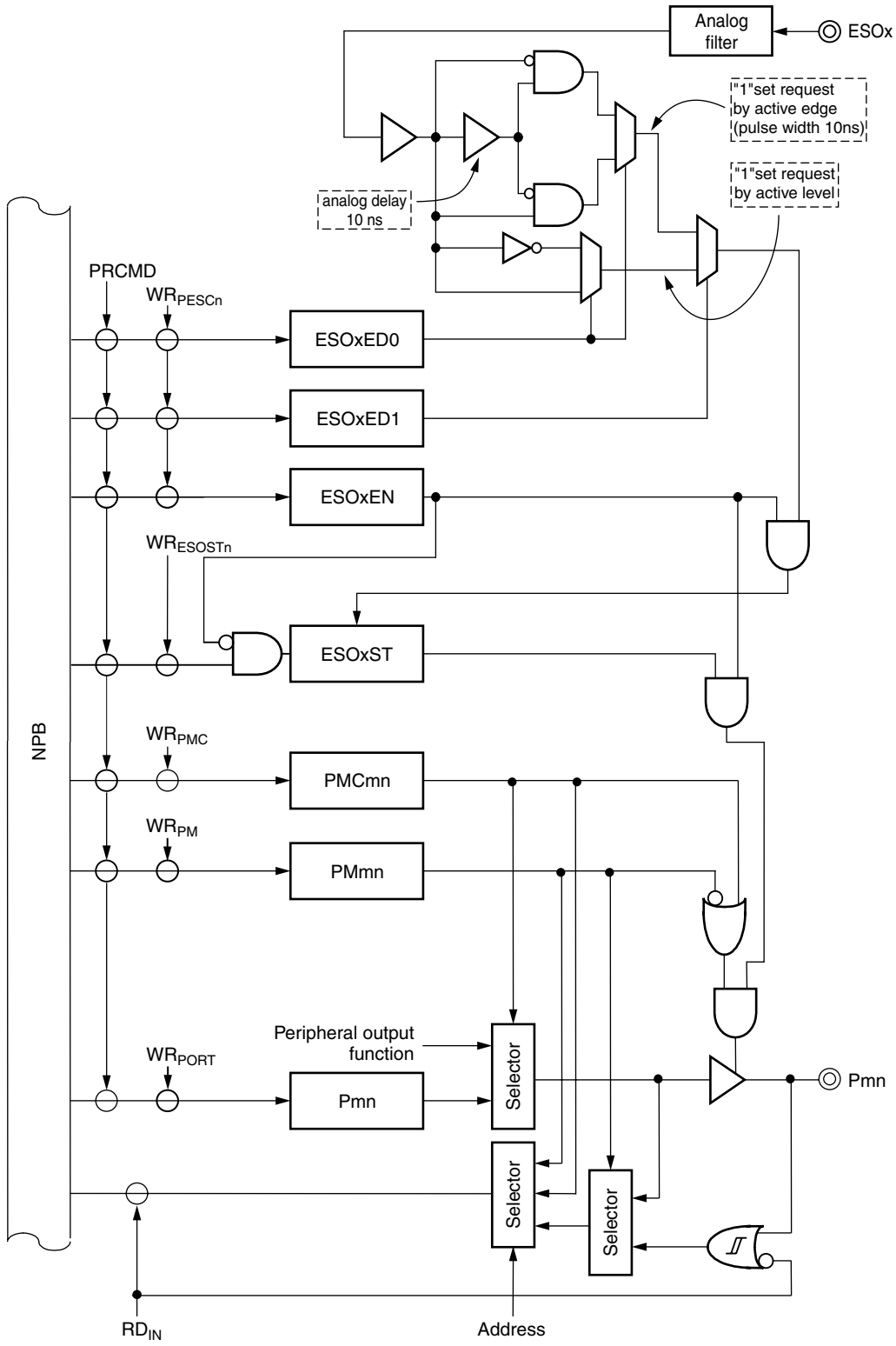


Remark: m: port number
n: port bit number

(18) Port type 13

Port type 13 provides a general purpose I/O port with peripheral output function. This type is similar to the port logic type 11, but the output driver can be shut down immediately by the ESOx input signal (x = 0, 1). All port registers are write protected against unintended change due to system or software malfunction. Writing to the port registers of type 13 is only possible immediately after a write access to the PRCMD register.

Figure 20-19: Port Type 13

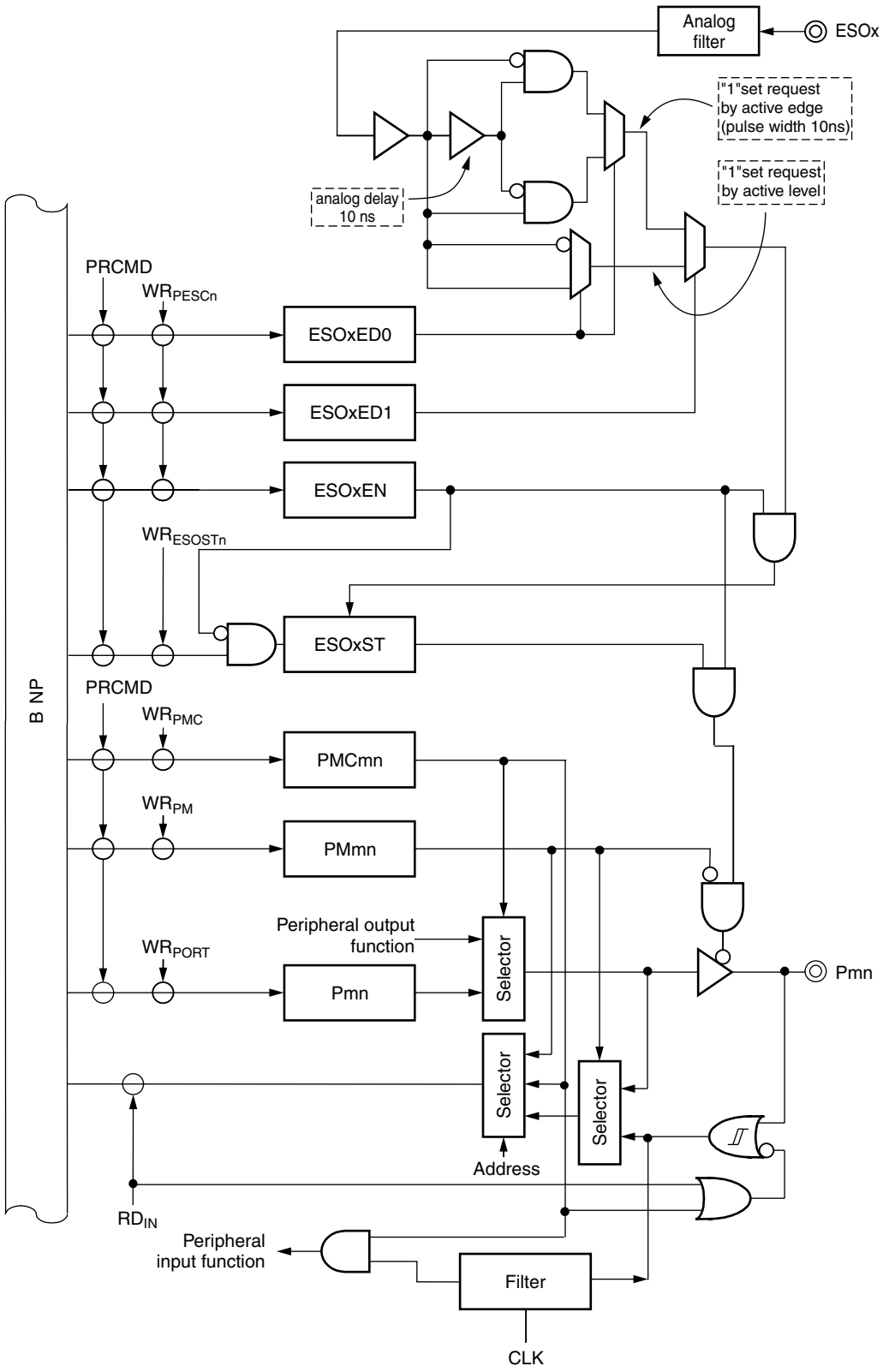


Remark: m: port number
n: port bit number
x: index of ESO signal (x = 0, 1)

(19) Port type 14

Port type 14 provides a general purpose I/O port with digitally filtered peripheral input function and peripheral output function. This type is similar to the port type 12, but the output driver can be shut down immediately by the ESOx input signal (x = 0, 1). All port registers are write protected against unintended change due to system or software malfunction. Writing to the port registers of type 13 is only possible immediately after a write access to the PRCMD register.

Figure 20-20: Port Type 14

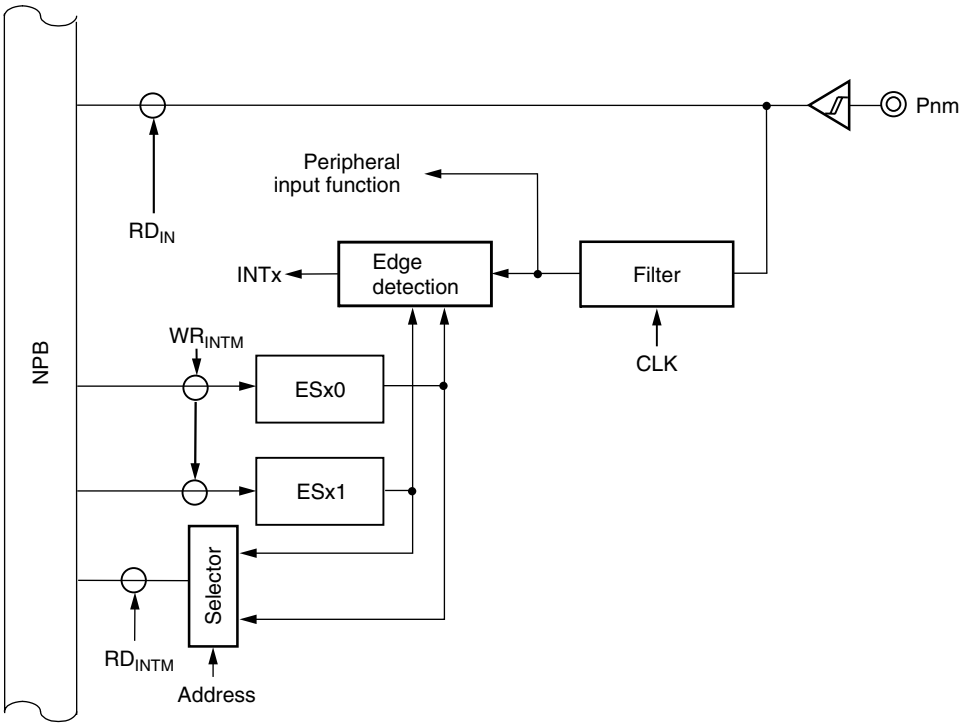


Remark: m: port number
n: port bit number
x: index of ESO signal (x = 0, 1)

(20) Port type 15

Port type 15 provides a general purpose input port with external interrupt input function. This type is similar as port type 3. Difference is the additional filtered peripheral input function support.

Figure 20-21: Port Type 15

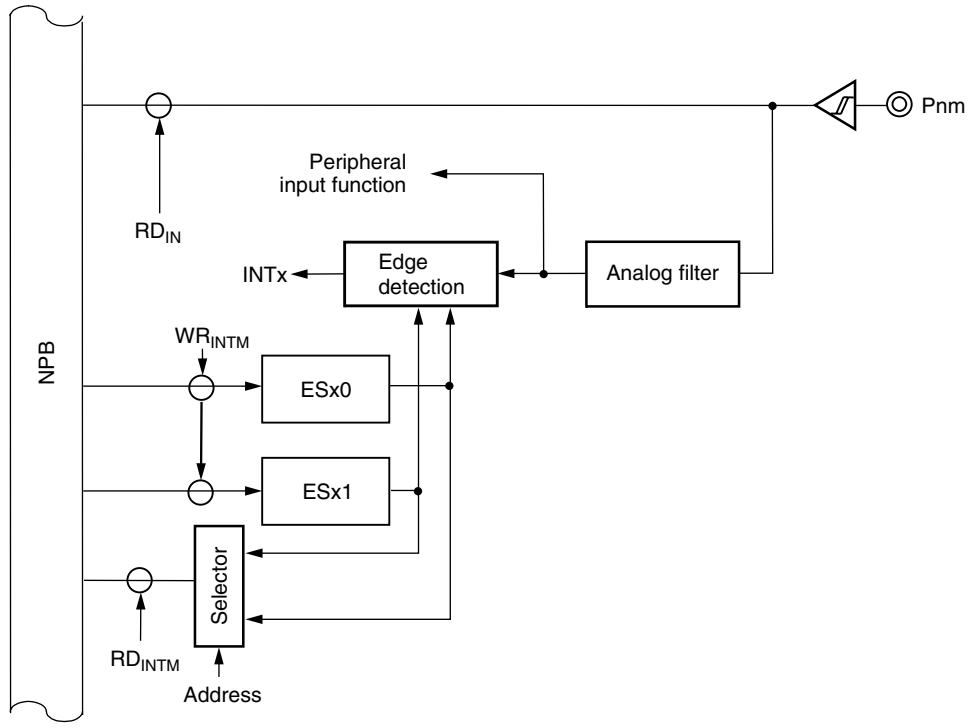


Remark: m: port number
n: port bit number
x: external interrupt number

(21) Port type 15A

Port type 15A provides a general purpose input port with external interrupt input function. This type is similar as port type 15. Difference is the analog filter instead of digital filter.

Figure 20-22: Port Type 15A



Remark: m: port number
n: port bit number
x: external interrupt number

20.2.3 Peripheral registers of I/O ports

The following table lists the peripheral registers related to I/O ports.

Table 20-2: Peripheral Registers of I/O Ports (1/3)

Address	Function Register Name	Symbol	Bit Units for Manipulation			After Reset
			1 Bit	8 Bits	16 Bits	
0xFFFFF000	Port register port AL low byte	PALL	R/W	R/W	-	0x00
0xFFFFF000	Port register port AL	PAL	-	-	R/W	0x0000
0xFFFFF001	Port register port AL high byte	PALH	R/W	R/W	-	0x00
0xFFFFF002	Port register port AH	PAH	R/W	R/W	-	0x00
0xFFFFF004	Port register port DL low byte	PDLL	R/W	R/W	-	0x00
0xFFFFF004	Port register port DL	PDL	-	-	R/W	0x0000
0xFFFFF005	Port register port DL high byte	PDLH	R/W	R/W	-	0x00
0xFFFFF006	Port register port DH low byte	PDHL	R/W	R/W	-	0x00
0xFFFFF006	Port register port DH	PDH	-	-	R/W	0x0000
0xFFFFF007	Port register port DH high byte	PDHH	R/W	R/W	-	0x00
0xFFFFF008	Port register port CS	PCS	R/W	R/W	-	0x00
0xFFFFF00A	Port register port CT	PCT	R/W	R/W	-	0x00
0xFFFFF00C	Port register port CM	PCM	R/W	R/W	-	0x00
0xFFFFF00E	Port register port CD	PCD	R/W	R/W	-	0x00
0xFFFFF020	Port mode register Port mode AL low byte	PMALL	R/W	R/W	-	0xFF
0xFFFFF020	Port mode register Port mode AL	PMAL	-	-	R/W	0xFFFF
0xFFFFF021	Port mode register Port mode AL high byte	PMALH	R/W	R/W	-	0xFF
0xFFFFF022	Port mode register Port mode AH	PMAH	R/W	R/W	-	0xFF
0xFFFFF024	Port mode register Port mode DL low byte	PMDLL	R/W	R/W	-	0xFF
0xFFFFF024	Port mode register Port mode DL	PMDL	-	-	R/W	0xFFFF
0xFFFFF025	Port mode register Port mode DL high byte	PMDLH	R/W	R/W	-	0xFF
0xFFFFF026	Port mode register Port mode DH low byte	PMDHL	R/W	R/W	-	0xFF
0xFFFFF026	Port mode register Port mode DH	PMDH	-	-	R/W	0xFFFF
0xFFFFF027	Port mode register Port mode DH high byte	PMDHH	R/W	R/W	-	0xFF
0xFFFFF028	Port mode register Port mode CS	PMCS	R/W	R/W	-	0xFF
0xFFFFF02A	Port mode register Port mode CT	PMCT	R/W	R/W	-	0xFF
0xFFFFF02C	Port mode register Port mode CM	PMCM	R/W	R/W	-	0xFF
0xFFFFF02E	Port mode register Port mode CD	PMCD	R/W	R/W	-	0xFF
0xFFFFF040	Port mode control register Port mode control AL low byte	PMCALL	R/W	R/W	-	0x00
0xFFFFF040	Port mode control register Port mode control AL	PMCAL	-	-	R/W	0x0000
0xFFFFF041	Port mode control register Port mode control AL high byte	PMCALH	R/W	R/W	-	0x00
0xFFFFF042	Port mode control register Port mode control AH	PMCAH	R/W	R/W	-	0x00
0xFFFFF044	Port mode control register Port mode control DL low byte	PMCDLL	R/W	R/W	-	0x00

Table 20-2: Peripheral Registers of I/O Ports (2/3)

Address	Function Register Name	Symbol	Bit Units for Manipulation			After Reset
			1 Bit	8 Bits	16 Bits	
0xFFFFF044	Port mode control register Port mode control DL	PMCDL	-	-	R/W	0x0000
0xFFFFF045	Port mode control register Port mode control DL high byte	PMCDLH	R/W	R/W	-	0x00
0xFFFFF046	Port mode control register Port mode control DH low byte	PMCDHL	R/W	R/W	-	0x00
0xFFFFF046	Port mode control register Port mode control DH	PMCDH	-	-	R/W	0x0000
0xFFFFF047	Port mode control register Port mode control DH high byte	PMCDHH	R/W	R/W	-	0x00
0xFFFFF048	Port mode control register Port mode control CS	PMCCS	R/W	R/W	-	0x00
0xFFFFF04A	Port mode control register Port mode control CT	PMCCT	R/W	R/W	-	0x00
0xFFFFF04C	Port mode control register Port mode control CM	PMCCM	R/W	R/W	-	0x00
0xFFFFF04E	Port mode control register Port mode control CD	PMCCD	R/W	R/W	-	0x00
0xFFFFF400	Port register port 0	P0	R	R	-	undef.
0xFFFFF402	Port register port 1	P1	R/W	R/W	-	undef.
0xFFFFF404	Port register port 2	P2	R/W	R/W	-	undef.
0xFFFFF406	Port register port 3	P3	R/W	R/W	-	undef.
0xFFFFF408	Port register port 4	P4	R/W	R/W	-	undef.
0xFFFFF40A	Port register port 5	P5	R/W	R/W	-	undef.
0xFFFFF40C	Port register port 6	P6	R/W	R/W	-	undef.
0xFFFFF40E	Port register port 7	P7	R/W	R/W	-	undef.
0xFFFFF410	Port register port 8	P8	R/W	R/W	-	undef.
0xFFFFF412	Port register port 9	P9	R/W	R/W	-	undef.
0xFFFFF414	Port register port 10	P10	R/W	R/W	-	undef.
0xFFFFF422	Port mode register port 1	PM1	R/W	R/W	-	0xFF
0xFFFFF424	Port mode register port 2	PM2	R/W	R/W	-	0xFF
0xFFFFF426	Port mode register port 3	PM3	R/W	R/W	-	0xFF
0xFFFFF428	Port mode register port 4	PM4	R/W	R/W	-	FFH
0xFFFFF42A	Port mode register port 5	PM5	R/W	R/W	-	0xFF
0xFFFFF42C	Port mode register port 6	PM6	R/W	R/W	-	0xFF
0xFFFFF42E	Port mode register port 7	PM7	R/W	R/W	-	FFH
0xFFFFF430	Port mode register port 8	PM8	R/W	R/W	-	0xFF
0xFFFFF432	Port mode register port 9	PM9	R/W	R/W	-	FFH
0xFFFFF434	Port mode register port 10	PM10	R/W	R/W	-	FFH
0xFFFFF442	Port mode control register port 1	PMC1	R/W	R/W	-	0x00
0xFFFFF444	Port mode control register port 2	PMC2	R/W	R/W	-	0x00
0xFFFFF446	Port mode control register port 3	PMC3	R/W	R/W	-	0x00
0xFFFFF448	Port mode control register port 4	PMC4	R/W	R/W	-	0x00

Table 20-2: Peripheral Registers of I/O Ports (3/3)

Address	Function Register Name	Symbol	Bit Units for Manipulation			After Reset
			1 Bit	8 Bits	16 Bits	
0xFFFFF44A	Port mode control register port 5	PMC5	R/W	R/W	-	0x00
0xFFFFF44C	Port mode control register port 6	PMC6	R/W	R/W	-	0x00
0xFFFFF44E	Port mode control register port 7	PMC7	R/W	R/W	-	0x00
0xFFFFF450	Port mode control register port 8	PMC8	R/W	R/W	-	0x00
0xFFFFF452	Port mode control register port 9	PMC9	R/W	R/W	-	0x00
0xFFFFF454	Port mode control register port 10	PMC10	R/W	R/W	-	0x00

20.2.4 Peripheral registers of valid edge control

The following table lists the peripheral registers related to valid edge control.

Table 20-3: Peripheral Registers of Valid Edge Control

Address	Function Register Name	Symbol	Bit Units for Manipulation			After Reset
			1 Bit	8 Bits	16 Bits	
0xFFFFF880	Interrupt mode register 0	INTM0	R/W	R/W	-	0x00
0xFFFFF882	Interrupt mode register 1	INTM1	R/W	R/W	-	0x00
0xFFFFF884	Interrupt mode register 2	INTM2	R/W	R/W	-	0x00
0xFFFFF886	Interrupt mode register 3	INTM3	R/W	R/W	-	0x00
0xFFFFF888	Port 5 emergency shut off control register	PESC5	R/W	R/W	-	0x00
0xFFFFF88A	Port 5 emergency shut off control register	ESOST5	R/W	R/W	-	0x00
0xFFFFF88C	Port 6 emergency shut off status register	PESC6	R/W	R/W	-	0x00
0xFFFFF88E	Port 6 emergency shut off status register	ESOST6	R/W	R/W	-	0x00

20.3 Port Pin Functions

20.3.1 Port 0

Port 0 is a 5-bit input only port.

(1) Functions

- Input data can be read in 1-bit units by using the port register 0 (P0).
- The alternate functions shared with the input port functionality of port 0 are always enabled.

Table 20-4: Alternate Function Pins and Port Types of Port 0

Port		Alternate Function	Remark	Port Type
Port 0	P00	NMI	Non maskable interrupt	3
	P01	INTP0, ESO0	External interrupt request input, Emergency output shut off input (TMR0)	15A
	P02	INTP1, ESO1	External interrupt request input, Emergency output shut off input (TMR1)	
	P03	INTP2, ADTRG0	External interrupt request input, External A/D conversion start trigger (ADC0)	15
	P04	INTP3, ADTRG1	External interrupt request input, External A/D conversion start trigger (ADC1)	

(2) Control registers

(a) Port register 0 (P0)

The port register 0 (P0) is an 8-bit register that reflects the input levels of port pins P00 to P04.
This register is read-only in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-23: Port Register 0 (P0)

After reset: Undefined R Address: FFFFF400H

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

P0n	Input Data Control of Pin P0n
0	Low level is input
1	High level is input

Remark: n = 0 to 4

20.3.2 Port 1

Port 1 is an 8-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 1 (P1).
- Input or output mode can be set in 1-bit units by using the port mode register 1 (PM1).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 1 (PMC1).

Table 20-5: Alternate Function Pins and Port Types of Port 1

Port		Alternate Function	Remark	Port Type
Port 1	P10	TIP00, TEVTP1, TOP00	Timer input (TMP0/TMP1) Timer output (TMP0)	6
	P11	TIP01, TTRGP1, TOP01	Timer input (TMP0/TMP1) Timer output (TMP0)	
	P12	TIP10, TTRGP0, TOP10	Timer input (TMP0/TMP1) Timer output (TMP1)	
	P13	TIP11, TEVTP0, TOP11	Timer input (TMP0/TMP1) Timer output (TMP1)	
	P14	TIP20, TEVTP3, TOP20	Timer input (TMP2/TMP3) Timer output (TMP2)	
	P15	TIP21, TTRGP3, TOP21	Timer input (TMP2/TMP3) Timer output (TMP2)	
	P16	TIP30, TTRGP2, TOP30	Timer input (TMP2/TMP3) Timer output (TMP3)	
	P17	TIP31, TEVTP2, TOP31	Timer input (TMP2/TMP3) Timer output (TMP3)	

(2) Control registers

(a) Port register 1 (P1)

The P1 register 1 is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P10 to P17.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-24: Port Register 1 (P1)

After reset: Undefined R/W Address: FFFFF402H

	7	6	5	4	3	2	1	0
P1	P17	P16	P15	P14	P13	P12	P11	P10

P1n	Input/Output Data Control of Pin P1n
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

Remark: n = 0 to 7

(b) Port mode register 1 (PM1)

The PM1 register is an 8-bit register that specifies the input or output mode.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-25: Port Mode Register 1 (PM1)

After reset: FFH R/W Address: FFFFF422H

	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	Input/Output Mode Control of Pin P1n (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 7

(c) Port mode control register 1 (PMC1)

The PMC1 register is an 8-bit register that specifies the port mode or control mode (alternate function).
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-26: Port Mode Control Register 1 (PMC1) (1/2)

After reset: 00H R/W Address: FFFFF442H

	7	6	5	4	3	2	1	0
PMC1	PMC17	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10

PMC17	Port/Control Mode Specification of Pin P17
0	I/O port mode
1	Control mode (alternate function) PM17: 0 – TOP31 output mode; 1 – TIP31, TEVTP2 input mode

PMC16	Port/Control Mode Specification of Pin P16
0	I/O port mode
1	Control mode (alternate function) PM16: 0 – TOP30 output mode; 1 – TIP30, TTRGP2 input mode

PMC15	Port/Control Mode Specification of Pin P15
0	I/O port mode
1	Control mode (alternate function) PM15: 0 – TOP21 output mode; 1 – TIP21, TTRGP3 input mode

PMC14	Port/Control Mode Specification of Pin P14
0	I/O port mode
1	Control mode (alternate function) PM14: 0 – TOP20 output mode; 1 – TIP20, TEVTP3 input mode

Figure 20-26: Port Mode Control Register 1 (PMC1) (2/2)

PMC13	Port/Control Mode Specification of Pin P13
0	I/O port mode
1	Control mode (alternate function) PM13: 0 = TOP11 output mode; 1 = TIP11, TEVTP0 input mode

PMC12	Port/Control Mode Specification of Pin P12
0	I/O port mode
1	Control mode (alternate function) PM12: 0 = TOP10 output mode; 1 = TIP10, TTRGP0 input mode

PMC11	Port/Control Mode Specification of Pin P11
0	I/O port mode
1	Control mode (alternate function) PM11: 0 = TOP01 output mode; 1 = TIP01, TTRGP1 input mode

PMC10	Port/Control Mode Specification of Pin P10
0	I/O port mode
1	Control mode (alternate function) PM10: 0 = TOP00 output mode; 1 = TIP00, TEVTP1 input mode

20.3.3 Port 2

Port 2 is an 8-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 2 (P2).
- Input or output mode can be set in 1-bit units by using the port mode register 2 (PM2).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 2 (PMC2).

Table 20-6: Alternate Function Pins and Port Types of Port 2

Port		Alternate Function	Remark	Port Type
Port 2	P20	TIP40, TEVTP5, TOP40	Timer input (TMP4/TMP5) Timer output (TMP4 output)	6
	P21	TIP41, TTRGP5, TOP41	Timer input (TMP4/TMP5) Timer output (TMP4 output)	
	P22	TIP50, TTRGP4, TOP50	Timer input (TMP4/TMP5) Timer output (TMP5 output)	
	P23	TIP51, TEVTP4, TOP51	Timer input (TMP4/TMP5) Timer output (TMP5 output)	
	P24	TIP60, TEVTP7, TOP60	Timer input (TMP6/TMP7) Timer output (TMP6 output)	
	P25	TIP61, TTRGP7, TOP61	Timer input (TMP6/TMP7) Timer output (TMP6 output)	
	P26	TIP70, TTRGP6, TOP70	Timer input (TMP6TMP7) Timer output (TMP7 output)	
	P27	TIP71, TEVTP6, TOP71	Timer input (TMP6/TMP7) Timer output (TMP7 output)	

(2) Control registers

(a) Port register 2 (P2)

The P2 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P20 to P27.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-27: Port Register 2 (P2)

After reset: Undefined R/W Address: FFFFF404H

	7	6	5	4	3	2	1	0
P2	P27	P26	P25	P24	P23	P22	P21	P20

P2n	Input/Output Data Control of Pin P2n
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

Remark: n = 0 to 7

(b) Port mode register 2 (PM2)

The PM2 register is an 8-bit register that specifies the input or output mode.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-28: Port Mode Register 2 (PM2)

After reset: FFH R/W Address: FFFFF424H

	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

PM2n	Input/Output Mode Control of Pin P2n (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 7

PM2 is an 8 bit read/write register. It is the port mode register of Port 2. PM2 determines the input or output direction of the respective port pin.

(c) Port mode control register 2 (PMC2)

The PMC2 register is an 8-bit register that specifies the port mode or control mode (alternate function).
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-29: Port Mode Control Register 2 (PMC2) (1/2)

After reset: 00H R/W Address: FFFFF444H

	7	6	5	4	3	2	1	0
PMC2	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20

PMC27	Port/Control Mode Specification of Pin P27
0	I/O port mode
1	Control mode (alternate function) PM27: 0 – TOP71 output mode; 1 – TIP71, TEVTP6 input mode

PMC26	Port/Control Mode Specification of Pin P26
0	I/O port mode
1	Control mode (alternate function) PM26: 0 – TOP70 output mode; 1 – TIP70, TTRGP6 input mode

PMC25	Port/Control Mode Specification of Pin P25
0	I/O port mode
1	Control mode (alternate function) PM25: 0 – TOP61 output mode; 1 – TIP61, TTRGP7 input mode

PMC24	Port/Control Mode Specification of Pin P24
0	I/O port mode
1	Control mode (alternate function) PM24: 0 – TOP60 output mode; 1 – TIP60, TEVTP7 input mode

Figure 20-29: Port Mode Control Register 1 (PMC2) (2/2)

PMC23	Port/Control Mode Specification of Pin P23
0	I/O port mode
1	Control mode (alternate function) PM23: 0 = TOP51 output mode; 1 = TIP51, TEVTP4 input mode

PMC22	Port/Control Mode Specification of Pin P22
0	I/O port mode
1	Control mode (alternate function) PM22: 0 = TOP50 output mode; 1 = TIP50, TTRGP4 input mode

PMC21	Port/Control Mode Specification of Pin P21
0	I/O port mode
1	Control mode (alternate function) PM21: 0 = TOP41 output mode; 1 = TIP41, TTRGP5 input mode

PMC20	Port/Control Mode Specification of Pin P20
0	I/O port mode
1	Control mode (alternate function) PM20: 0 = TOP40 output mode; 1 = TIP40, TEVTP5 input mode

20.3.4 Port 3

Port 3 is an 8-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 3 (P3).
- Input or output mode can be set in 1-bit units by using the port mode register 3 (PM3).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 3 (PMC3).
- The external interrupt request inputs shared with the input port functionality of port 3 are always enabled in input port mode.

Table 20-7: Alternate Function Pins and Port Types of Port 3

Port		Alternate Function	Remark	Port Type
Port 3	P30	RXDC0, INTP4	Serial interface (UARTC0) input External interrupt request input	9
	P31	TXDC0	Serial interface (UARTC0) output	1S
	P32	RXDC1, INTP5	Serial interface (UARTC1) input External interrupt request input	9
	P33	TXDC1	Serial interface (UARTC1) output	1S
	P34	FCRXD0	FCAN0 input	2A
	P35	FCTXD0	FCAN0 output	1S
	P36	FCRXD1	FCAN1 input	2A
	P37	FCTXD1	FCAN1 output	1S

(2) Control registers

(a) Port register 3 (P3)

The P3 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P30 to P37.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-30: Port Register 3 (P3)

After reset: Undefined R/W Address: FFFFF406H

	7	6	5	4	3	2	1	0
P3	P37	P36	P35	P34	P33	P32	P31	P30

P3n	Input/Output Data Control of Pin P3n
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

(b) Port mode register 3 (PM3)

The PM3 register is an 8-bit register that specifies the input or output mode.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-31: Port Mode Register 3 (PM3)

After reset: FFH R/W Address: FFFFF426H

	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	Input/Output Mode Control of Pin P3n (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 7

(c) Port mode control register 3 (PMC3)

The PMC3 register is an 8-bit register that specifies the port mode or control mode (alternate function).
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-32: Port Mode Control Register 3 (PMC3) (1/2)

After reset: 00H R/W Address: FFFFF446H

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

PMC37	Port/Control Mode Specification of Pin P37
0	I/O port mode
1	FCTXD1 output mode

PMC36	Port/Control Mode Specification of Pin P36
0	I/O port mode**Note**
1	FCRXD1 input mode

PMC35	Port/Control Mode Specification of Pin P35
0	I/O port mode
1	FCTXD0 output mode

PMC34	Port/Control Mode Specification of Pin P34
0	I/O port mode**Note**
1	FCRXD0 input mode

PMC33	Port/Control Mode Specification
0	I/O port mode
1	TXDC1 output mode

Note: If this pin is set to port mode, the corresponding peripheral input signal (alternate function) is forced to high level internally.

Figure 20-32: Port Mode Control Register 3 (PMC3) (2/2)

PMC33	Port/Control Mode Specification of Pin P33
0	I/O port mode
1	TXDC1 output mode

PMC32	Port/Control Mode Specification of Pin P32
0	I/O port mode[Note] PM32 = 0: Output mode PM32 = 1: Input mode, External interrupt request input mode (INTP5)
1	RXDC1 input mode, External interrupt request input mode (INTP5)

PMC31	Port/Control Mode Specification of Pin P31
0	I/O port mode
1	TXDC0 output mode

PMC30	Port/Control Mode Specification of Pin P30
0	I/O port mode[Note] PM30 = 0: Output mode PM30 = 1: Input mode, External interrupt request input mode (INTP4)
1	RXDC0 input mode, External interrupt request input mode (INTP4)

Note: If this pin is set to port mode, the corresponding peripheral input signal (alternate function) is forced to high level internally.

20.3.5 Port 4

Port 4 is a 6-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 4 (P4).
- Input or output mode can be set in 1-bit units by using the port mode register 4 (PM4).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 4 (PMC4).

Table 20-8: Alternate Function Pins and Port Types of Port 4

Port		Alternate Function	Remark	Port Type
Port 4	P40	SIB0	Serial interface (CSIB0) input	2
	P41	SOB0	Serial interface (CSIB0) output	1E
	P42	SCKB0	Serial interface (CSIB0) I/O	4C
	P43	SIB1	Serial interface (CSIB1) input	2
	P44	SOB1	Serial interface (CSIB1) output	1E
	P45	SCKB1	Serial interface (CSIB1 I/O	4C

(2) Control registers

(a) Port register 4 (P4)

The P4 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P40 to P45.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-33: Port Register 4 (P4)

After reset: Undefined R/W Address: FFFFF408H

	7	6	5	4	3	2	1	0
P4	0	0	P45	P44	P43	P42	P41	P40

P4n	Input/Output Data Control of Pin P4n
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

(b) Port mode register 4 (PM4)

The PM4 register is an 8-bit register that specifies the input or output mode.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-34: Port Mode Register 4 (PM4)

After reset: FFH R/W Address: FFFFF428H

	7	6	5	4	3	2	1	0
PM4	1	1	PM45	PM44	PM43	PM42	PM41	PM40

PM4n	Input/Output Mode Control of Pin P4n (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 5

(c) Port mode control register 4 (PMC4)

The PMC4 register is an 8-bit register that specifies the port mode or control mode (alternate function).
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-35: Port Mode Control Register 4 (PMC4)

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	0	0	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40

PMC45	Port/Control Mode Specification of Pin P45
0	I/O port mode
1	SCKB1 I/O mode (input or output mode controlled by CSIB1)

PMC44	Port/Control Mode Specification of Pin P44
0	I/O port mode
1	SOB1 output mode

PMC43	Port/Control Mode Specification of Pin P43
0	I/O port mode
1	SIB1input mode

PMC42	Port/Control Mode Specification of Pin P42
0	I/O port mode
1	SCKB0 I/O mode (input or output mode controlled by CSIB0)

PMC41	Port/Control Mode Specification of Pin P41
0	I/O port mode
1	SOB0 output mode

PMC40	Port/Control Mode Specification of Pin P40
0	I/O port mode
1	SIB0 input mode

20.3.6 Port 5

Port 5 is an 8-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 5 (P5).
- Input or output mode can be set in 1-bit units by using the port mode register 5 (PM5).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 5 (PMC5).
- Emergency shut off by ES0 input signal of output buffers P51 to P56 can be controlled by port emergency shut off control register 5 (PESC5) and emergency shut off status register 5 (ESOST5).
- Security feature to protect the timer output signals of TMR0 from unintended CPU interference. Registers P5, PM5, PMC5, PESC5 and ESOST5 can only be written in a special sequence.

Table 20-9: Alternate Function Pins and Port Types of Port 5

Port		Alternate Function	Remark	Port Type
Port 5	P50	TOR00	Timer output (TMR0)	11
	P51	TOR01		13
	P52	TOR02		
	P53	TOR03		
	P54	TOR04		
	P55	TOR05		
	P56	TOR06		
	P57	TOR07		11

(2) Control registers

(a) Port register 5 (P5)

The P5 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P50 to P57.
Writing to the P5 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the P5 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register P5. For details refer to **3.4.8 Specific registers**.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-36: Port Register 5 (P5)

After reset: Undefined R/W Address: FFFFF40AH

	7	6	5	4	3	2	1	0
P5	P57	P56	P55	P54	P53	P52	P51	P50

P5n	Input/Output Data Control of Pin P5n
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

(b) Port mode register 5 (PM5)

The PM5 register is an 8-bit register that specifies the input or output mode.
Writing to the PM5 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PM5 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PM5. For details refer to **3.4.8 Specific registers**.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-37: Port Mode Register 5 (PM5)

After reset: FFH R/W Address: FFFFF42AH

	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	Input/Output Mode Control of Pin P5n (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 7

(c) Port mode control register 5 (PMC5)

The PMC5 register is an 8-bit register that specifies the port mode or control mode (alternate function).
Writing to the PMC5 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PMC5 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PMC5. For details refer to **3.4.8 "Specific registers" on page 127**.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-38: Port Mode Control Register 5 (PMC5))

After reset: 00H R/W Address: FFFFF44AH

	7	6	5	4	3	2	1	0
PMC5	PMC57	PMC56	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50

PMC5n	Port/Control Mode Specification of Pin P5n
0	I/O port mode
1	TOR0n output mode

Remark: n = 0 to 7

(d) Port emergency shut off control register 5 (PESC5)

The PESC5 register is an 8-bit register that controls the emergency shut off behaviour of output buffers of ports P51 to P56.
Writing to the PESC5 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PESC5 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PESC5. For details refer to **3.4.8 "Specific registers" on page 127**.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-39: Port Emergency Shut Off Control Register 5 (PESC5)

After reset: 00H R/W Address: FFFFF888H

	7	6	5	4	3	2	1	0
PESC5	0	0	0	0	ESO0EN	0	ESO0ED1	ESO0EDO0

ESO0EN	Emergency Output Shut Off Enable Control
0	Emergency shut off control by ESO0 input disabled
1	Emergency shut off control by ESO0 input enabled

ESO0ED1	ESO0ED0	Valid Edge Specification of Emergency Shut Off Input (ESO0)
0	0	Falling edge
0	1	Rising edge
1	0	Low level
1	1	High level

Caution: State of the edge detection control bits ESO0ED1 and ESO0ED0 must not be changed while ESO0EN is set (1). Otherwise the output shut off function may be unintentionally triggered or a trigger event may be lost.

Remarks: 1. The output buffers of ports P51 to P56 are forcibly disabled (high impedance output) as long as ESO0EN and ESO0ST are set to 1.

2. Setup of the emergency shut off function must be performed in the following sequence. Otherwise the output shut off function may be unintentionally triggered or a trigger event may be lost.

<1> power on (All registers are reset)
<2> PRCMD write (write protect released)
<3> clear ESO0EN bit to 0
<4> PRCMD write (write protect released)
<5> clear ESO0ST bit of ESOST5 register to 0
<6> PRCMD write (write protect released)
<7> set ESO0ED0, ESO0ED1 bits
<8> PRCMD write (write protect released)
<9> set ESO0EN bit to 1

(e) Port emergency shut off status register 5 (ESOST5)

The ESOST5 register is an 8-bit register that indicates the emergency status control mode (alternate function).
Writing to the ESOST5 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the ESOST5 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register ESOST5. For details refer to **3.4.8 Specific registers**.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-40: Port Emergency Shut Off Status Register 5 (ESOST5))

After reset: 00H R/W Address: FFFFF88AH

	7	6	5	4	3	2	1	0
ESOST5	ESO0ST	0	0	0	0	0	0	0

ESO0ST	Emergency Shut Off Status
0	No emergency shut off was triggered
1	Emergency shut off of output ports P51 to P56 triggered by ESO0 input

Remarks: 1. Writing the emergency shut off status flag (ESO0ST) is only possible, if the ES0EN bit of the PESC5 register is cleared (0).

2. The ESO0ST flag can only be cleared by CPU to 0. Setting the ESO0ST flag to 1 is not possible.

3. The output buffers of P51 to P56 are forcibly disabled as long as ESO0EN and ESO0ST of the PESC5 register are set to 1.

20.3.7 Port 6

Port 6 is an 8-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 6 (P6).
- Input or output mode can be set in 1-bit units by using the port mode register 6 (PM6).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 6 (PMC6).
- Emergency shut off by ES0 input signal of output buffers P61 to P66 can be controlled by port emergency shut off control register 6 (PESC6) and emergency shut off status register 6 (ESOST6).
- Security feature to protect the timer output signals of TMR0 from unintended CPU interference. Registers P6, PM6, PMC6, PESC6 and ESOST6 can only be written in a special sequence.

Table 20-10: Alternate Function Pins and Port Types of Port 6

Port		Alternate Function	Remark	Port Type
Port 6	P60	TOR10, TTRGR1	Timer I/O (TMR1)	12
	P61	TOR11, TIR10		14
	P62	TOR12, TIR11		
	P63	TOR13, TIR12		
	P64	TOR14, TIR13		
	P65	TOR15	Timer output (TMR1)	13
	P66	TOR16		
	P67	TOR17, TEVTR1	Timer I/O (TMR1)	12

(2) Control registers

(a) Port register 6 (P6)

The P6 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P60 to P67.
Writing to the P6 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the P6 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register P6. For details refer to **3.4.8 Specific registers**.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-41: Port Register 6 (P6)

After reset: Undefined R/W Address: FFFFF40CH

	7	6	5	4	3	2	1	0
P6	P67	P66	P65	P64	P63	P62	P61	P60

P6n	Input/Output Data Control of Pin P6n
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

(b) Port mode register 6 (PM6)

The PM6 register is an 8-bit register that specifies the input or output mode.
Writing to the PM6 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PM6 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PM6. For details refer to **3.4.8 Specific registers**.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-42: Port Mode Register 6 (PM6)

After reset: FFH R/W Address: FFFFF42CH

	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60

PM6n	Input/Output Mode Control of Pin P6n (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 7

(c) Port mode control register 6 (PMC6)

The PMC6 register is an 8-bit register that specifies the port mode or control mode (alternate function).
Writing to the PMC6 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PMC6 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PMC6. For details refer to **3.4.8 Specific registers**.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-43: Port Mode Control Register 6 (PMC6) (1/2)

After reset: 00H R/W Address: FFFFF44CH

	7	6	5	4	3	2	1	0
PMC6	PMC67	PMC66	PMC65	PMC64	PMC63	PMC62	PMC61	PMC60

PMC67	Port/Control Mode Specification of Pin P67
0	I/O port mode
1	Control mode

PM67	Function
0	TOR17 output mode
1	TTEVTR1 input mode

PMC66	Port/Control Mode Specification of Pin P66
0	I/O port mode
1	TOR16 output mode

PMC65	Port/Control Mode Specification of Pin P65
0	I/O port mode
1	TOR15 output mode

PMC64	Port/Control Mode Specification of Pin P64
0	I/O port mode
1	Control mode

PM64	Function
0	TOR14 output mode
1	TIR13 input mode

Figure 20-43: Port Mode Control Register 6 (PMC6) (2/2)

PMC63	Port/Control Mode Specification of Pin P63
0	I/O port mode
1	Control mode

PM63	Function
0	TOR13 output mode
1	TIR12 input mode

PMC62	Port/Control Mode Specification of Pin P62
0	I/O port mode
1	Control mode

PM62	Function
0	TOR12 output mode
1	TIR11 input mode

PMC61	Port/Control Mode Specification of Pin P61
0	I/O port mode
1	Control mode

PM61	Function
0	TOR11 output mode
1	TIR10 input mode

PMC60	Port/Control Mode Specification of Pin P60
0	I/O port mode
1	Control mode

PM63	Function
0	TOR10 output mode
1	TTRGR1 input mode

Remark: n = 0 to 7

(d) Port emergency shut off control register 6 (PESC6)

The PESC6 register is an 8-bit register that controls the emergency shut off behaviour of output buffers of ports P61 to P66.
Writing to the PESC6 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PESC6 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PESC6. For details refer to **3.4.8 Specific registers**.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-44: Port Emergency Shut Off Control Register 6 (PESC6)

After reset: 00H R/W Address: FFFFF88CH

	7	6	5	4	3	2	1	0
PESC6	0	0	0	0	ESO0EN	0	ESO0ED1	ESO0EDO0

ESO1EN	Emergency Output Shut Off Enable Control
0	Emergency shut off control by ESO1 input disabled
1	Emergency shut off control by ESO1 input enabled

ESO1ED1	ESO1ED0	Valid Edge Specification of Emergency Shut Off Input (ESO1)
0	0	Falling edge
0	1	Rising edge
1	0	Low level
1	1	High level

Caution: State of the edge detection control bits ESO1ED1 and ESO1ED0 must not be changed while ESO1EN is set (1). Otherwise the output shut off function may be unintentionally triggered or a trigger event may be lost.

Remarks: 1. The output buffers of ports P61 to P66 are forcibly disabled (high impedance output) as long as ESO1EN and ESO1ST are set to 1.

2. Setup of the emergency shut off function must be performed in the following sequence. Otherwise the output shut off function may be unintentionally triggered or a trigger event may be lost.

<1> power on (All registers are reset)
<2> PRCMD write (write protect released)
<3> clear ESO1EN bit to 0
<4> PRCMD write (write protect released)
<5> clear ESO1ST bit of ESOST6 register to 0
<6> PRCMD write (write protect released)
<7> set ESO1ED0, ESO1ED1 bits
<8> PRCMD write (write protect released)
<9> set ESO1EN bit to 1

(e) Port emergency shut off status register 6 (ESOST6)

The ESOST6 register is an 8-bit register that indicates the emergency status control mode (alternate function).
Writing to the ESOST6 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the ESOST6 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register ESOST6. For details refer to **3.4.8 "Specific registers" on page 127**.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-45: Port Emergency Shut Off Status Register 6 (ESOST6))

After reset: 00H R/W Address: FFFFF88EH

	7	6	5	4	3	2	1	0
ESOST6	ESO1ST	0	0	0	0	0	0	0

ESO1ST	Emergency Shut Off Status
0	No emergency shut off was triggered
1	Emergency shut off of output ports P61 to P66 triggered by ESO1 input

Remarks: 1. Writing the emergency shut off status flag (ESO1ST) is only possible, if the ES1EN bit of the PESC6 register is cleared (0).

2. The ESO1ST flag can only be cleared by CPU to 0. Setting the ESO1ST flag to 1 is not possible.

3. The output buffers of P61 to P66 are forcibly disabled as long as ESO1EN and ESO1ST of the PESC6 register are set to 1.

20.3.8 Port 7

Port 7 is an 8-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 7 (P7).
- Input or output mode can be set in 1-bit units by using the port mode register 7 (PM7).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 7 (PMC7).
- The external interrupt request input shared with the input port functionality of port 7 is always enabled in input port mode.

Table 20-11: Alternate Function Pins and Port Types of Port 7

Port		Alternate Function	Remark	Port Type
Port 7	P70	TIT00, TEVT1, TOT00	Timer input (TMT0, TMT1) Timer output (TMT0)	6
	P71	TIT01, TTRGT1, TOT01	Timer input (TMT0, TMT1 Timer output (TMT0)	
	P72	TECRT0, INTP12	Timer input (TMT0), External interrupt request input	8
	P73	TIT10, TTRGT0, TOT10	Timer input (TMT0, TMT1) Timer output (TMT1)	6
	P74	TIT11, TEVT0, TOT11	Timer input (TMT0, TMT1 Timer output (TMT1)	
	P75	TECRT1, AFO	Timer input (TMT1) Auxiliary frequency output	

(2) Control registers

(a) Port register 7 (P7)

The P7 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P70 to P75.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-46: Port Register 7 (P7)

After reset: Undefined R/W Address: FFFFF40EH

	7	6	5	4	3	2	1	0
P7	0	0	P75	P74	P73	P72	P71	P70

P7n	Input/Output Data Control of Pin P7n
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

(b) Port mode register 4 (PM7)

The PM7 register is an 8-bit register that specifies the input or output mode.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-47: Port Mode Register 7 (PM7)

After reset: FFH R/W Address: FFFFF42EH

	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	Input/Output Mode Control of Pin P7n (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 5

(c) Port mode control register 4 (PMC7)

The PMC7 register is an 8-bit register that specifies the port mode or control mode (alternate function).
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-48: Port Mode Control Register 7 (PMC7) (1/2)

After reset: 00H R/W Address: FFFFF44EH

	7	6	5	4	3	2	1	0
PMC7	0	0	PMC75	PMC74	PMC73	PMC72	PMC71	PMC70

PMC75	Port/Control Mode Specification of Pin P75
0	I/O port mode
1	Control mode

PM75	Function
0	AFO output mode
1	TECRT1 input mode

PMC74	Port/Control Mode Specification of Pin P74
0	I/O port mode
1	Control mode

PM74	Function
0	TOT11 output mode
1	TIT11 input mode, TEVTT0 input mode

PMC73	Port/Control Mode Specification of Pin P73
0	I/O port mode
1	Control mode

PM73	Function
0	TOT10 output mode
1	TIT10 input mode, TTRGT0 input mode

Figure 20-48: Port Mode Control Register 7 (PMC7) (2/2)

PMC72	Port/Control Mode Specification of Pin P72
0	I/O port mode PM72 = 0: Output mode PM72 = 1: Input mode, External interrupt request input mode (INTP12)
1	TECRT0 input mode External interrupt request input mode (INTP12)

PMC71	Port/Control Mode Specification of Pin P71
0	I/O port mode
1	Control mode PM71 = 0: TOT01 output mode PM71 = 1: TIT01, TTRGT1 input mode

PMC70	Port/Control Mode Specification of Pin P70
0	I/O port mode
1	Control mode PM70 = 0: TOT00 output mode PM70 = 1: TIT00, TEVTT1 input mode

20.3.9 Port 8

Port 8 is a 7-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 8 (P8).
- Input or output mode can be set in 1-bit units by using the port mode register 8 (PM8).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 8 (PMC8).
- The external interrupt request inputs shared with the input port functionality of port 8 are always enabled in input port mode.

Table 20-12: Alternate Function Pins and Port Types of Port 8

Port		Alternate Function	Remark	Port Type
Port 8	P80	SI30	Serial interface (CSI30) input	2
	P81	SO30	Serial interface (CSI30) output	1S
	P82	SCK30	Serial interface (CSI30) I/O	4
	P83	SCS300, INTP6	Serial interface (CSI30) output, External interrupt request input	7
	P84	SCS301, INTP7	Serial interface (CSI30) output, External interrupt request input	
	P85	SCS302, INTP8	Serial interface (CSI30) output, External interrupt request input	
	P86	SCS303, SSB0	Serial interface (CSI30) output, Serial interface (CSIB0) input	5

(2) Control registers

(a) Port register 8 (P8)

The P8 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P80 to P86.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-49: Port Register 8 (P8)

After reset: Undefined R/W Address: FFFFF410H

	7	6	5	4	3	2	1	0
P8	0	P86	P85	P84	P83	P82	P81	P80

P8n	Input/Output Data Control of Pin P8n
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

(b) Port mode register 8 (PM8)

The PM8 register is an 8-bit register that specifies the input or output mode.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-50: Port Mode Register 8 (PM8)

After reset: FFH R/W Address: FFFFF430H

	7	6	5	4	3	2	1	0
PM8	1	PM86	PM85	PM84	PM83	PM82	PM81	PM80

PM8n	Input/Output Mode Control of Pin P8n (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 6

(c) Port mode control register 8 (PMC8)

The PMC8 register is an 8-bit register that specifies the port mode or control mode (alternate function).
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-51: Port Mode Control Register 8 (PMC8) (1/2)

After reset: 00H R/W Address: FFFFF450H

	7	6	5	4	3	2	1	0
PMC8	0	PMC86	PMC85	PMC84	PMC83	PMC82	PMC81	PMC80

PMC86	Port/Control Mode Specification of Pin P86
0	I/O port mode
1	Control mode

PM86	Function
0	SCS303 output mode
1	SSB0 input mode

PMC85	Port/Control Mode Specification of Pin P85
0	I/O port mode

PM85	Function
0	Output mode
1	Input mode, External interrupt request input mode (INTP8)

1	Control mode

PM85	Function
0	SCS302 output mode
1	External interrupt request input mode (INTP8)

PMC84	Port/Control Mode Specification of Pin P84
0	I/O port mode

PM84	Function
0	Output mode
1	Input mode, External interrupt request input mode (INTP7)

1	Control mode

PM84	Function
0	SCS301 output mode
1	External interrupt request input mode (INTP7)

Figure 20-51: Port Mode Control Register 8 (PMC8) (2/2)

PMC83	Port/Control Mode Specification of Pin P83
0	I/O port mode PM83 = 0: Output PM83 = 1: Input, External interrupt request input mode (INTP6)
1	Control mode PM83 = 0: SCS300 output mode PM83 = 1: External interrupt request input mode (INTP6)

PMC82	Port/Control Mode Specification of Pin P82
0	I/O port mode
1	SCK30 I/O mode

PMC81	Port/Control Mode Specification of Pin P81
0	I/O port mode
1	SO30 output mode

PMC80	Port/Control Mode Specification of Pin P80
0	I/O port mode
1	SI30 input mode

20.3.10 Port 9

Port 9 is a 7-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 9 (P9).
- Input or output mode can be set in 1-bit units by using the port mode register 9 (PM9).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 9 (PMC9).
- The external interrupt request inputs shared with the input port functionality of port 9 are always enabled in input port mode.

Table 20-13: Alternate Function Pins and Port Types of Port 9

Port		Alternate Function	Remark	Port Type
Port 9	P90	SI31	Serial interface (CSI31) input	2
	P91	SO31	Serial interface (CSI31) output	1S
	P92	SCK31	Serial interface (CSI31) I/O	4
	P93	SCS310, INTP9	Serial interface (CSI31) output, External interrupt request input	7
	P94	SCS311, INTP10	Serial interface (CSI31) output, External interrupt request input	
	P95	SCS312, INTP11	Serial interface (CSI31) output, External interrupt request input	
	P96	SCS313, SSB1	Serial interface (CSI31) output, Serial interface (CSIB1) input	5

(2) Control registers

(a) Port register 9 (P9)

The P9 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P90 to P96.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-52: Port Register 9 (P9)

After reset: Undefined R/W Address: FFFFF412H

	7	6	5	4	3	2	1	0
P9	0	P96	P95	P94	P93	P92	P91	P90

P9n	Input/Output Data Control of Pin P9n
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

(b) Port mode register 9 (PM9)

The PM9 register is an 8-bit register that specifies the input or output mode.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-53: Port Mode Register 9 (PM9)

After reset: FFH R/W Address: FFFFF432H

	7	6	5	4	3	2	1	0
PM9	1	PM96	PM95	PM94	PM93	PM92	PM91	PM90

PM9n	Input/Output Mode Control of Pin P9n (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 6

(c) Port mode control register 9 (PMC9)

The PMC9 register is an 8-bit register that specifies the port mode or control mode (alternate function).
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-54: Port Mode Control Register 9 (PMC9) (1/2)

After reset: 00H R/W Address: FFFFF452H

	7	6	5	4	3	2	1	0
PMC9	0	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

PMC96	Port/Control Mode Specification of Pin P96
0	I/O port mode
1	Control mode

PM96	Function
0	SCS313 output mode
1	SSB1 input mode

PMC95	Port/Control Mode Specification of Pin P95
0	I/O port mode

PM95	Function
0	Output mode
1	Input mode, External interrupt request input mode (INTP11)

1	Control mode

PM95	Function
0	SCS312 output mode
1	External interrupt request input mode (INTP11)

PMC94	Port/Control Mode Specification of Pin P94
0	I/O port mode

PM94	Function
0	Output mode
1	Input mode, External interrupt request input mode (INTP10)

1	Control mode

PM94	Function
0	SCS311 output mode
1	External interrupt request input mode (INTP10)

Figure 20-54: Port Mode Control Register 9 (PMC9) (2/2)

<table>
<tr><th>PMC93</th><th>Port/Control Mode Specification of Pin P93</th></tr>
<tr><td>0</td><td>I/O port mode
<table>
<tr><th>PM93</th><th>Function</th></tr>
<tr><td>0</td><td>Output mode</td></tr>
<tr><td>1</td><td>Input mode,
External interrupt request input mode (INTP9)</td></tr>
</table></td></tr>
<tr><td>1</td><td>Control mode
<table>
<tr><th>PM93</th><th>Function</th></tr>
<tr><td>0</td><td>SCS310 output mode</td></tr>
<tr><td>1</td><td>External interrupt request input mode (INTP9</td></tr>
</table></td></tr>
</table>

PMC92	Port/Control Mode Specification of Pin P92
0	I/O port mode
1	SCK31 I/O mode

PMC91	Port/Control Mode Specification of Pin P91
0	I/O port mode
1	SO31 output mode

PMC90	Port/Control Mode Specification of Pin P90
0	I/O port mode
1	SI31 input mode

20.3.11 Port 10

Port 10 is a 3-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 10 (P10).
- Input or output mode can be set in 1-bit units by using the port mode register 10 (PM10).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 10 (PMC10).

Table 20-14: Alternate Function Pins and Port Types of Port 10

Port		Alternate Function	Remark	Port Type
Port 10	P100	TCLR0, TICC00, TOP81	Timer input (ITENC0) Timer output (TMP8)	6
	P101	TCUD0, TICC01	Timer input (ITENC0)	10
	P102	TIUD0, TO1	Timer input (ITENC0), Timer output (ITENC0)	6

(2) Control registers

(a) Port register 10 (P10)

The P10 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P100 to P105.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-55: Port Register 10 (P10)

After reset: Undefined R/W Address: FFFFF414H

	7	6	5	4	3	2	1	0
P10	0	0	0	0	0	P102	P101	P100

P10n	Input/Output Data Control of Pin P10n
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

(b) Port mode register 10 (PM10)

The PM10 register is an 8-bit register that specifies the input or output mode.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-56: Port Mode Register 10 (PM10)

After reset: FFH R/W Address: FFFFF434H

	7	6	5	4	3	2	1	0
PM10	1	1	1	1	1	PM102	PM101	PM100

PM10n	Input/Output Mode Control of Pin P10n (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 2

(c) Port mode control register 10 (PMC10)

The PMC10 register is an 8-bit register that specifies the port mode or control mode (alternate function).
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-57: Port Mode Control Register 10 (PMC10)

After reset: 00H R/W Address: FFFFF454H

	7	6	5	4	3	2	1	0
PMC10	0	0	0	0	0	PMC102	PMC101	PMC100

PMC102	Port/Control Mode Specification of Pin P102
0	I/O port mode
1	Control mode

PM102	Function
0	TOP81 output mode
1	TCLR0 input mode, TICC0 input mode

PMC101	Port/Control Mode Specification of Pin P101
0	I/O port mode
1	TCUD0 input mode, TICC01 input mode

PMC100	Port/Control Mode Specification of Pin P100
0	I/O port mode
1	Control mode

PM100	Function
0	TO1 output mode
1	TIUD0 input mode

20.3.12 Port AL

Port AL is a 16-bit I/O port that can be set to input or output mode in 1-bit units. When the higher 8 bits of port AL are used as port ALH (PALH) and the lower 8 bits as port ALL (PALL), port AL becomes two 8-bit ports that can be set in the input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register AL (PAL).
- Input or output mode can be set in 1-bit units by using the port mode register AL (PMAL).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register AL (PMCAL).

Table 20-15: Alternate Function Pins and Port Types of Port AL

Port		Alternate Function	Remark	Port Type
Port AL	PAL0 to PAL15	A0 to A15	External address bus	1

Caution: In single-chip mode 1 and in ROM-less mode this port has external address bus function only. Reprogramming of port AL to port mode is not possible in these modes. Reading and writing of the port register PAL and port mode register PMAL is possible but has no effect. Reading of the port mode control register PMCAL is possible and the result is always FFFFH. Writing of the port mode control register PMCAL is not possible.

(2) Control registers

(a) Port register AL (PAL)

The PAL register is a 16-bit register that controls reading the pin levels and writing the output levels of port pins PAL0 to PAL15.
This register can be read or written in 16-bit units.
If the higher 8 bits of the PAL register are used as PALH register, and the lower 8 bits as the PALL register, however, these registers can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-58: Port Register AL(PAL)

After reset: Undefined R/W Address: FFFFF000H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAL	PAL15	PAL14	PAL13	PAL12	PAL11	PAL10	PAL9	PAL8	PAL7	PAL6	PAL5	PAL4	PAL3	PAL2	PAL1	PAL0
	PALH								PALL							

After reset: Undefined R/W Address: FFFFF000H

	7	6	5	4	3	2	1	0
PALL	PAL7	PAL6	PAL5	PAL4	PAL3	PAL2	PAL1	PAL0

After reset: Undefined R/W Address: FFFFF001H

	7	6	5	4	3	2	1	0
PALH	PAL15	PAL14	PAL13	PAL12	PAL11	PAL10	PAL9	PAL8

PALn	Input/Output Data Control of Pin PALn
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

Remark: n = 0 to 15

(b) Port mode register AL (PMAL)

The PMAL register is a 16-bit register that specifies the input or output mode of port pins PAL0 to PAL15.
This register can be read or written in 16-bit units.
If the higher 8 bits of the PMAL register are used as PMALH register, and the lower 8 bits as the PMALL register, however, these registers can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFFFH.

Figure 20-59: Port Mode Register AL(PMAL)

After reset: FFFFH R/W Address: FFFFF020H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMAL	PMAL15	PMAL14	PMAL13	PMAL12	PMAL11	PMAL10	PMAL9	PMAL8	PMAL7	PMAL6	PMAL5	PMAL4	PMAL3	PMAL2	PMAL1	PMAL0
	PMALH								PMALL							

After reset: FFH R/W Address: FFFFF020H

	7	6	5	4	3	2	1	0
PMALL	PMAL7	PMAL6	PMAL5	PMAL4	PMAL3	PMAL2	PMAL1	PMAL0

After reset: FFH R/W Address: FFFFF021H

	7	6	5	4	3	2	1	0
PMALH	PMAL15	PMAL14	PMAL13	PMAL12	PMAL11	PMAL10	PMAL9	PMAL8

PMALn	Input/Output Mode Control of Pin PALn (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 15

(c) Port AL mode control register

The PMCAL register is a 16-bit read/write register that specifies the port mode or control mode (alternate function) of Port AL.
This register can be read or written in 16-bit units.
If the higher 8 bits of the PMCAL register are used as PMCALH register, and the lower 8 bits as the PMCALL register, however, these registers can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 0000H in single-chip mode 0, and to FFFFH in ROM-less mode and single-chip mode 1.

Figure 20-60: Port Mode Control Register AL (PMCAL)

After reset[Note 1]: 0000H/ FFFFH R/W[Note 2] Address: FFFFF040H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMCAL	PMCAL15	PMCAL14	PMCAL13	PMCAL12	PMCAL11	PMCAL10	PMCAL9	PMCAL8	PMCAL7	PMCAL6	PMCAL5	PMCAL4	PMCAL3	PMCAL2	PMCAL1	PMCAL0
	PMCALH								PMCALL							

After reset[Note 1]: 00H/FFH R/W[Note 2] Address: FFFFF040H

	7	6	5	4	3	2	1	0
PMCALL	PMCAL7	PMCAL6	PMCAL5	PMCAL4	PMCAL3	PMCAL2	PMCAL1	PMCAL0

After reset[Note 1]: 00H/FFH R/W[Note 2] Address: FFFFF041H

	7	6	5	4	3	2	1	0
PMCALH	PMCAL15	PMCAL14	PMCAL13	PMCAL12	PMCAL11	PMCAL10	PMCAL9	PMCAL8

PMCALn	Port/Control Mode Specification of Pin PALn
0	I/O port mode
1	External address bus output mode (A15 to A0)

Notes: 1. In single-chip mode 0: 0000H, or 00H respectively
In single-chip mode 1 and ROM-less mode: FFFFH, or FFH respectively

2. In the single-chip mode 1 or in the ROM-Less mode, this register can not be written. Reading is possible and returns FFFFH, or FFH respectively.

Remark: n = 0 to 15

20.3.13 Port AH

Port AH is a 6-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register AH (PAH).
- Input or output mode can be set in 1-bit units by using the port mode register AH (PMAH).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register AH (PMCAH).

Table 20-16: Alternate Function Pins and Port Types of Port AH

Port		Alternate Function	Remark	Port Type
Port AH	PAH0 to PAH5	A16 to A21	External address bus	1

Caution: In single-chip mode 1 and in ROM-less mode this port has external address bus function only. Reprogramming of port AH to port mode is not possible in these modes. Reading and writing of the port register PAH and port mode register PMAH is possible but has no effect. Reading of the port mode control register PMCAH is possible and the result is always 3FH. Writing of the port mode control register PMCAH is not possible.

(2) Control registers

(a) Port register AH (PAH)

The PAH register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins PAH0 to PAH5.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-61: Port Register AH (PAH)

After reset: Undefined R/W Address: FFFFF002H

	7	6	5	4	3	2	1	0
PAH	0	0	PAH5	PAH4	PAH3	PAH2	PAH1	PAH0

PAHn	Input/Output Data Control of Pin PAHn
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

Remark: n = 0 to 5

(b) Port mode register AH (PMAH)

The PMAH register is an 8-bit register that specifies the input or output mode of port pins PAL0 to PAL15.
This register can be read or written in 8-bit units.
Reset input sets this register to FFH.

Figure 20-62: Port Mode Register AH (PMAH)

After reset: FFH R/W Address: FFFFF022H

	7	6	5	4	3	2	1	0
PMAH	1	1	PMAH5	PMAH4	PMAH3	PMAH2	PMAH1	PMAH0

PMAHn	Input/Output Mode Control of Pin PAHn (in Port Mode)
0	Output mode
1	Input mode

(c) Port mode control register AH (PMCAH)

The PMCAHL register is an 8-bit register that specifies the port mode or control mode (alternate function) of Port AL.
This register can be read or written in 8-bit units.
Reset input sets this register to 00H in single-chip mode 0, and to 3FH in ROM-less mode and single-chip mode 1.

Figure 20-63: Port Mode Control Register AH (PMCAH)

After reset[Note 1]: 00H/3FH R/W[Note 2] Address: FFFFF042H

	7	6	5	4	3	2	1	0
PMCAH	0	0	PMCAH5	PMCAH4	PMCAH3	PMCAH2	PMCAH1	PMCAH0

PMCAHn	Port/Control Mode Specification of Pin PAHn
0	I/O port mode
1	External memory address bus output mode (A21 to A16)

Notes: 1. In single-chip mode 0: 00H
In single-chip mode 1 and ROM-less mode: 3FH

2. In the single-chip mode 1 or in the ROM-less mode, this register can not be written. Reading is possible and returns 3FH.

Remark: n = 0 to 5

20.3.14 Port DL

Port DL is a 16-bit I/O port that can be set to input or output mode in 1-bit units. When the higher 8 bits of port DL are used as port DLH (PDLH) and the lower 8 bits as port DLL (PDLL), port DL becomes two 8-bit ports that can be set in the input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register DL (PDL).
- Input or output mode can be set in 1-bit units by using the port mode register DL (PMDL).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register DL (PMCDL).

Table 20-17: Alternate Function Pins and Port Types of Port DL

Port		Alternate Function	Remark	Port Type
Port DL	PDL0 to PDL15	D0 to D15	External data bus	4C

Caution: In single-chip mode 1 and in ROM-less mode this port has external data bus function only. Reprogramming of port DL to port mode is not possible in these modes. Reading and writing of the port register PDL and port mode register PMDL is possible but has no effect. Reading of the port mode control register PMCDL is possible and the result is always FFFFH. Writing of the port mode control register PMCDL is not possible.

(2) Control registers

(a) Port register DL (PDL)

The PDL register is a 16-bit register that controls reading the pin levels and writing the output levels of port pins PDL0 to PDL15.
This register can be read or written in 16-bit units.
If the higher 8 bits of the PDL register are used as PDLH register, and the lower 8 bits as the PDLL register, however, these registers can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-64: Port Register DL(PDL)

After reset: Undefined R/W Address: FFFFF004H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDL	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0
	PDLH								PDLL							

After reset: Undefined R/W Address: FFFFF004H

	7	6	5	4	3	2	1	0
PDLL	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

After reset: Undefined R/W Address: FFFFF005H

	7	6	5	4	3	2	1	0
PDLH	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8

PDLn	Input/Output Data Control of Pin PDLn
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

Remark: n = 0 to 15

(b) Port mode register DL (PMDL)

The PMDL register is a 16-bit register that specifies the input or output mode of port pins PDL0 to PDL15.
This register can be read or written in 16-bit units.
If the higher 8 bits of the PMDL register are used as PMDLH register, and the lower 8 bits as the PMDLL register, however, these registers can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFFFH.

Figure 20-65: Port Mode Register DL(PMDL)

After reset: FFFFH R/W Address: FFFFF024H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMDL	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0
	PMDLH								PMDLL							

After reset: FFH R/W Address: FFFFF024H

	7	6	5	4	3	2	1	0
PMDLL	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

After reset: FFH R/W Address: FFFFF025H

	7	6	5	4	3	2	1	0
PMDLH	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8

PMDLn	Input/Output Mode Control of Pin PDLn (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 15

(c) Port mode control register DL (PMCDL)

The PMCDL register is a 16-bit read/write register that specifies the port mode or control mode (alternate function) of Port DL.
This register can be read or written in 16-bit units.
If the higher 8 bits of the PMCDL register are used as PMCDLH register, and the lower 8 bits as the PMCDLL register, however, these registers can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 0000H in single-chip mode 0, and to FFFFH in ROM-less mode and single-chip mode 1.

Figure 20-66: Port Mode Control Register DL (PMCDL)

After reset[Note 1]: 0000H/ FFFFH R/W[Note 2] Address: FFFFF044H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMCDL	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0
	PMCDLH								PMCDLL							

After reset[Note 1]: 00H/FFH R/W[Note 2] Address: FFFFF044H

	7	6	5	4	3	2	1	0
PMCDLL	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0

After reset[Note 1]: 00H/FFH R/W[Note 2] Address: FFFFF045H

	7	6	5	4	3	2	1	0
PMCDLH	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8

PMCDLn	Port/Control Mode Specification of Pin PDLn
0	I/O port mode
1	External data bus output mode (D15 to D0)

Notes: 1. In single-chip mode 0: 0000H, or 00H respectively
In single-chip mode 1 and ROM-less mode: FFFFH, or FFH respectively

2. In the single-chip mode 1 or in the ROM-less mode, this register can not be written. Reading is possible and returns FFFFH, or FFH respectively.

Remark: n = 0 to 15

20.3.15 Port DH

Port DH is a 16-bit I/O port that can be set to input or output mode in 1-bit units. When the higher 8 bits of port DH are used as port DHH (PDHH) and the lower 8 bits as port DHL (PDHL), port DH becomes two 8-bit ports that can be set in the input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register DH (PDH).
- Input or output mode can be set in 1-bit units by using the port mode register DH (PMDH).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register DH (PMCDH).

Table 20-18: Alternate Function Pins and Port Types of Port DH

Port		Alternate Function	Remark	Port Type
Port DH	PDH0 to PDH15	D16 to D31	External data bus	4C

Caution: In single-chip mode 1 and in ROM-less mode this port has external data bus function only. Reprogramming of port DH to port mode is not possible in these modes. Reading and writing of the port register PDH and port mode register PMDH is possible but has no effect. Reading of the port mode control register PMCDH is possible and the result is always FFFFH. Writing of the port mode control register PMCDH is not possible.

(2) Control registers

(a) Port register DH (PDH)

The PDH register is a 16-bit register that controls reading the pin levels and writing the output levels of port pins PDH0 to PDH15.
This register can be read or written in 16-bit units.
If the higher 8 bits of the PDH register are used as PDHH register, and the lower 8 bits as the PDHL register, however, these registers can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-67: Port Register DH(PDH)

After reset: Undefined R/W Address: FFFFF006H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDH	PDH15	PDH14	PDH13	PDH12	PDH11	PDH10	PDH9	PDH8	PDH7	PDH6	PDH5	PDH4	PDH3	PDH2	PDH1	PDH0
	PDHH								PDHL							

After reset: Undefined R/W Address: FFFFF006H

	7	6	5	4	3	2	1	0
PDHL	PDH7	PDH6	PDH5	PDH4	PDH3	PDH2	PDH1	PDH0

After reset: Undefined R/W Address: FFFFF007H

	7	6	5	4	3	2	1	0
PDHH	PDH15	PDH14	PDH13	PDH12	PDH11	PDH10	PDH9	PDH8

PDHn	Input/Output Data Control of Pin PDHn
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

Remark: n = 0 to 15

(b) Port mode register DH (PMDH)

The PMDH register is a 16-bit register that specifies the input or output mode of port pins PDH0 to PDH15.
This register can be read or written in 16-bit units.
If the higher 8 bits of the PMDH register are used as PMDHH register, and the lower 8 bits as the PMDHL register, however, these registers can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFFFH.

Figure 20-68: Port Mode Register DH(PMDH)

After reset: FFFFH R/W Address: FFFFF026H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMDH	PMDH15	PMDH14	PMDH13	PMDH12	PMDH11	PMDH10	PMDH9	PMDH8	PMDH7	PMDH6	PMDH5	PMDH4	PMDH3	PMDH2	PMDH1	PMDH0
	PMDHH								PMDHL							

After reset: FFH R/W Address: FFFFF026H

	7	6	5	4	3	2	1	0
PMDHL	PMDH7	PMDH6	PMDH5	PMDH4	PMDH3	PMDH2	PMDH1	PMDH0

After reset: FFH R/W Address: FFFFF027H

	7	6	5	4	3	2	1	0
PMDHH	PMDH15	PMDH14	PMDH13	PMDH12	PMDH11	PMDH10	PMDH9	PMDH8

PMDHn	Input/Output Mode Control of Pin PDHn (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0 to 15

(c) Port mode control register DH (PMCDH)

The PMCDH register is a 16-bit read/write register that specifies the port mode or control mode (alternate function) of Port DH.
This register can be read or written in 16-bit units.
If the higher 8 bits of the PMCDH register are used as PMCDHH register, and the lower 8 bits as the PMCDHL register, however, these registers can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 0000H in single-chip mode 0, and to FFFFH in ROM-less mode and single-chip mode 1.

Figure 20-69: Port Mode Control Register DH (PMCDH)

After reset[Note 1]: 0000H/ FFFFH R/W[Note 2] Address: FFFFF046H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMCDH	PMCDH15	PMCDH14	PMCDH13	PMCDH12	PMCDH11	PMCDH10	PMCDH9	PMCDH8	PMCDH7	PMCDH6	PMCDH5	PMCDH4	PMCDH3	PMCDH2	PMCDH1	PMCDH0
	PMCDHH								PMCDHL							

After reset[Note 1]: 00H/FFH R/W[Note 2] Address: FFFFF046H

	7	6	5	4	3	2	1	0
PMCDHL	PMCDH7	PMCDH6	PMCDH5	PMCDH4	PMCDH3	PMCDH2	PMCDH1	PMCDH0

After reset[Note 1]: 00H/FFH R/W[Note 2] Address: FFFFF047H

	7	6	5	4	3	2	1	0
PMCDHH	PMCDH15	PMCDH14	PMCDH13	PMCDH12	PMCDH11	PMCDH10	PMCDH9	PMCDH8

PMCDHn	Port/Control Mode Specification of Pin PDHn
0	I/O port mode
1	External memory address bus output mode (D31 to D16)

Notes: 1. In single-chip mode 0: 0000H, or 00H respectively
In single-chip mode 1 and ROM-less mode: FFFFH, or FFH respectively

2. In the single-chip mode 1 or in the ROM-less mode, this register can not be written. Reading is possible and returns FFFFH, or FFH respectively.

Remark: n = 0 to 15

20.3.16 Port CS

Port CS is a 4-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register CS (PCS).
- Input or output mode can be set in 1-bit units by using the port mode register CS (PMCS).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register CS (PMCCS).

Table 20-19: Alternate Function Pins and Port Types of Port CS

Port		Alternate Function	Remark	Port Type
Port CS	PCS0 to PCS5	$\overline{CS0}$	Chip select signal output	1
		$\overline{CS1}$		
		$\overline{CS3}$		
		$\overline{CS4}$		

Caution: In single-chip mode 1 and in ROM-less mode this port has external control bus function only. Reprogramming of port CS to port mode is not possible in these modes. Reading and writing of the port register PCS and port mode register PMCS is possible but has no effect. Reading of the port mode control register PMCCS is possible and the result is always 1BH. Writing of the port mode control register PMCCS is not possible.

(2) Control registers

(a) Port register CS (PCS)

The PCS register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins PCS0, PCS1, PCS3 and PCS4.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-70: Port Register CS (PCS)

After reset: Undefined R/W Address: FFFFF008H

	7	6	5	4	3	2	1	0
PCS	0	0	0	PCS4	PCS3	0	PCS1	PCS0

PCSn	Input/Output Data Control of Pin PCSn
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

Remark: n = 0, 1, 3, 4

(b) Port mode register CS (PMCS)

The PMCS register is an 8-bit register that specifies the input or output mode of port pins PCS0, PCS1, PCS3 and PCS4.
This register can be read or written in 8-bit units.
Reset input sets this register to FFH.

Figure 20-71: Port Mode Register CS (PMCS)

After reset: FFH R/W Address: FFFFF028H

	7	6	5	4	3	2	1	0
PMCS	1	1	1	PMCS4	PMCS3	1	PMCS1	PMCS0

PMCSn	Input/Output Mode Control of Pin PCSn (in Port Mode)
0	Output mode
1	Input mode

(c) Port mode control register CS (PMCCS)

The PMCCSL register is an 8-bit register that specifies the port mode or control mode (alternate function) of port pins PCS0, PCS1, PCS3 and PCS4.
This register can be read or written in 8-bit units.
Reset input sets this register to 00H in single-chip mode 0, and to 1BH in ROM-less mode and single-chip mode 1.

Figure 20-72: Port Mode Control Register CS (PMCCS)

After reset[Note 1]: 00H/3FH R/W[Note 2] Address: FFFFF048H

	7	6	5	4	3	2	1	0
PMCCS	0	0	0	PMCCS4	PMCCS3	0	PMCCS1	PMCCS0

PMCCSn	Port/Control Mode Specification of Pin PCSn
0	I/O port mode
1	Chip select signal output mode ($\overline{CSn}$)

Notes: 1. In single-chip mode 0: 00H
In single-chip mode 1 and ROM-less mode: 1BH

2. In the single-chip mode 1 or in the ROM-less mode, this register can not be written. Reading is possible and returns 1BH.

Remark: n = 0, 1, 3, 4

20.3.17 Port CT

Port CT is a 2-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register CT (PCT).
- Input or output mode can be set in 1-bit units by using the port mode register CT (PMCT).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register CT (PMCCT).

Table 20-20: Alternate Function Pins and Port Types of Port CT

Port		Alternate Function	Remark	Port Type
Port CT	PCT4	$\overline{RD}$	Read strobe signal output	1
	PCT5	$\overline{WR}$	Write strobe signal output	

Caution: In single-chip mode 1 and in ROM-less mode this port has external control bus function only. Reprogramming of port CT to port mode is not possible in these modes. Reading and writing of the port register PCT and port mode register PMCT is possible but has no effect. Reading of the port mode control register PMCCT is possible and the result is always 30H. Writing of the port mode control register PMCCT is not possible.

(2) Control registers

(a) Port register CT (PCT)

The PCT register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins PCT4 and PCT5.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-73: Port Register CT (PCT)

After reset: Undefined R/W Address: FFFFF00AH

	7	6	5	4	3	2	1	0
PCT	0	0	PCT5	PCT4	0	0	0	0

PCTn	Input/Output Data Control of Pin PCTn
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

Remark: n = 4, 5

(b) Port mode register CT (PMCT)

The PMCT register is an 8-bit register that specifies the input or output mode of port pins PCT4 and PCT5.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-74: Port Mode Register CT (PMCT)

After reset: FFH R/W Address: FFFFF02AH

	7	6	5	4	3	2	1	0
PMCT	1	1	PMCT5	PMCT4	1	1	1	1

PMCTn	Input/Output Mode Control of Pin PCTn (in Port Mode)
0	Output mode
1	Input mode

(c) Port mode control register CT (PMCCT)

The PMCCTL register is an 8-bit register that specifies the port mode or control mode (alternate function) of port pins PCT4 and PCT5.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H in single-chip mode 0, and to 30H in ROM-less mode and single-chip mode 1.

Figure 20-75: Port Mode Control Register CT (PMCCT)

After reset[Note 1]: 00H/3FH R/W[Note 2] Address: FFFFF04AH

	7	6	5	4	3	2	1	0
PMCCT	0	0	PMCCT5	PMCCT4	0	0	0	0

PMCCT5	Port/Control Mode Specification of Pin PCT5
0	I/O port mode
1	Write strobe signal output mode ($\overline{WR}$)

PMCCT4	Port/Control Mode Specification of Pin PCT4
0	I/O port mode
1	Read strobe signal output mode ($\overline{RD}$)

Notes: 1. In single-chip mode 0: 00H
In single-chip mode 1 and ROM-less mode: 30H

2. In the single-chip mode 1 or in the ROM-less mode, this register can not be written. Reading is possible and returns 30H.

Remark: n = 4, 5

20.3.18 Port CM

Port CM is a 4-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register CM (PCM).
- Input or output mode can be set in 1-bit units by using the port mode register CM (PMCM).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register CM (PMCCM).

Table 20-21: Alternate Function Pins and Port Types of Port CM

Port		Alternate Function	Remark	Port Type
Port CM	PCM0	$\overline{\text{WAIT}}$	Wait insertion signal input	2C
	PCM1	-		1
	PCM6	-		
	PCM7	-		

Caution: In single-chip mode 1 and in ROM-less mode this port has external control bus function only. Reprogramming of port CM to port mode is not possible in these modes. Reading and writing of the port register PCM and port mode register PMCM is possible but has no effect. Reading of the port mode control register PMCCM is possible and the result is always 01H. Writing of the port mode control register PMCCM is not possible.

(2) Control registers

(a) Port register CM (PCM)

The PCM register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins PCM0, PCM1, PCM6 and PCM7.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-76: Port Register CM (PCM)

After reset: Undefined R/W Address: FFFFF00CH

	7	6	5	4	3	2	1	0
PCM	PCM7	PCM6	0	0	0	0	PCM1	PCM0

PCMn	Input/Output Data Control of Pin PCMn
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

Remark: n = 0, 1, 6, 7

(b) Port mode register CM (PMCM)

The PMCM register is an 8-bit register that specifies the input or output mode of port pins PCM0, PCM1, PCM6 and PCM7.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-77: Port Mode Register CM (PMCM)

After reset: FFH R/W Address: FFFFF02CH

	7	6	5	4	3	2	1	0
PMCM	PMCM7	PMCM6	1	1	1	1	PMCM1	PMCM0

PMCMn	Input/Output Mode Control of Pin PCMn (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 0, 1, 6, 7

(c) Port mode control register CM (PMCCM)

The PMCCML register is an 8-bit register that specifies the port mode or control mode (alternate function) of port pins PCM0, PCM1, PCM6 and PCM7.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H in single-chip mode 0, and to 01H in ROM-less mode and single-chip mode 1.

Figure 20-78: Port Mode Control Register CM (PMCCM)

After reset[Note 1]: 00H/01H R/W[Note 2] Address: FFFFF04CH

	7	6	5	4	3	2	1	0
PMCCM	0	0	0	0	0	0	0	PMCCM0

PMCCM0	Port/Control Mode Specification of Pin PCM0
0	I/O port mode
1	Wait insertion signal input mode ($\overline{\text{WAIT}}$)

Notes: 1. In single-chip mode 0: 00H
In single-chip mode 1 and ROM-less mode: 01H

2. In the single-chip mode 1 or in the ROM-less mode, this register can not be written. Reading is possible and returns 01H.

20.3.19 Port CD

Port CD is a 4-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register CD (PCD).
- Input or output mode can be set in 1-bit units by using the port mode register CD (PMCD).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register CD (PMCCD).

Table 20-22: Alternate Function Pins and Port Types of Port CD

Port		Alternate Function	Remark	Port Type
Port CD	PCD2	$\overline{\text{BEN0}}$	Byte enable signal output	1
	PCD3	$\overline{\text{BEN1}}$		
	PCD4	$\overline{\text{BEN2}}$		
	PCD5	$\overline{\text{BEN3}}$		

Caution: In single-chip mode 1 and in ROM-less mode this port has external control bus function only. Reprogramming of port CD to port mode is not possible in these modes. Reading and writing of the port register PCD and port mode register PMCD is possible but has no effect. Reading of the port mode control register PMCCD is possible and the result is always 3CH. Writing of the port mode control register PMCCD is not possible.

(2) Control registers

(a) Port register CD (PCD)

The PCD register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins PCD2 to PCD5.
This register can be read or written in 8-bit or 1-bit units.
Reset input causes an undefined register content.

Figure 20-79: Port Register CD (PCD)

After reset: Undefined R/W Address: FFFFF00EH

	7	6	5	4	3	2	1	0
PCD	0	0	PCD5	PCD4	PCD3	PCD2	0	0

PCDn	Input/Output Data Control of Pin PCDn
0	Input mode: Low level is input Output mode: Low level is output
1	Input mode: High level is input Output mode: High level is output

Remark: n = 2 to 5

(b) Port mode register CD (PMCD)

The PMCD register is an 8-bit register that specifies the input or output mode of port pins PCD2 to PCD5.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to FFH.

Figure 20-80: Port Mode Register CD (PMCD)

After reset: FFH R/W Address: FFFFF02EH

	7	6	5	4	3	2	1	0
PMCD	1	1	PMCD5	PMCD4	PMCD3	PMCD2	1	1

PMCDn	Input/Output Mode Control of Pin PCDn (in Port Mode)
0	Output mode
1	Input mode

Remark: n = 2 to 5

(c) Port mode control register CD (PMCCD)

The PMCCDL register is an 8-bit register that specifies the port mode or control mode (alternate function) of port pins PCD2 to PCD5.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H in single-chip mode 0, and to 3CH in ROM-less mode and single-chip mode 1.

Figure 20-81: Port Mode Control Register CD (PMCCD)

After reset[Note 1]: 00H/3FH R/W[Note 2] Address: FFFFF04EH

	7	6	5	4	3	2	1	0
PMCCD	0	0	PMCCD5	PMCCD4	PMCCD3	PMCCD2	0	0

PMCCD5	Port/Control Mode Specification of Pin PCD5
0	I/O port mode
1	Bus enable signal output mode ($\overline{BEN3}$)

PMCCD4	Port/Control Mode Specification of Pin PCD4
0	I/O port mode
1	Bus enable signal output mode ($\overline{BEN2}$)

PMCCD3	Port/Control Mode Specification of Pin PCD3
0	I/O port mode
1	Bus enable signal output mode ($\overline{BEN1}$)

PMCCD2	Port/Control Mode Specification of Pin PCD2
0	I/O port mode
1	Bus enable signal output mode ($\overline{BEN0}$)

Notes: 1. In single-chip mode 0: 00H
In single-chip mode 1 and ROM-less mode: 3CH

2. In the single-chip mode 1 or in the ROM-less mode, this register can not be written. Reading is possible and returns 3CH.

20.4 Noise Elimination

A timing controller used to secure the noise elimination time is provided for the pins shown in Table 20-23 below. Input signals that change within the noise elimination time are not internally acknowledged.

Table 20-23: Noise Elimination (1/2)

Unit	Pin	Delay Type	Noise Elimination Time	Sampling Clock
Reset	$\overline{\text{RESET}}$	Analog Delay	Several 10 ns (typ.)	
On-chip debug	DRST			
Non-maskable interrupt	P00/NMI	Digital delay	4 to 5 clocks	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
• Maskable Interrupt • Forced output stop function (TMR) • A/D converter (ADC)	P01/INTP0/ESO0 P02/INTP1/ESO1	Analog Delay	60 ns to 200 ns	
	P03/INTP2/ADTRG0 P04/INTP3/ADTRG1	Digital delay	4 to 5 clocks	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
• Maskable Interrupt • Asynchronous serial Interface (UART C)	P30/RXDC0/INTP4 P32/RXDC1/INTP5			
• Maskable Interrupt • Clocked serial interface 3 (CSI3)	P83/SCS300/INTP6 P84/SCS301/INTP7 P85/SCS302/INTP8 P93/SCS300/INTP9 P94/SCS301/INTP10 P95/SCS302/INTP11			
Timer ENC (TMNEC10)	P100/TCLR0/TICC00/TOP81 P101/TCUD0/TICC01 P102/TIUD0/TO1			

Table 20-23: Noise Elimination (2/2)

Unit	Pin	Delay Type	Noise Elimination Time	Sampling Clock
Timer P (TMP)	P10/TIP00/TEVTP1/TOP00 P11/TIP01/TTRGP1/TOP01 P12/TIP10/TTRGP0/TOP10 P13/TIP11/TEVTP0/TOP11 P14/TIP20/TEVTP3/TOP20 P15/TIP21/TTRGP3/TOP21 P16/TIP30/TTRGP2/TOP30 P17/TIP31/TEVTP2/TOP31 P20/TIP40/TEVTP5/TOP40 P21/TIP41/TTRGP5/TOP41 P22/TIP50/TTRGP4/TOP50 P23/TIP51/TEVTP4/TOP51 P24/TIP60/TEVTP7/TOP60 P25/TIP61/TTRGP7/TOP61 P26/TIP70/TTRGP6/TOP70 P27/TIP71/TEVTP6/TOP71	Digital delay	4 to 5 clocks	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
Timer R (TMR)	P60/TOR10/TTRGR1 P61/TOR11/TIR10 P62/TOR12/TIR11 P63/TOR13/TIR12 P64/TOR14/TIR13 P67/TOR17/TEVTR1			
Timer T (TMT)	P70/TIT00/TEVTT1/TOT00 P71/TIT01/TTRGT1/TOT01 P72/TECRT0/INTP12 P73/TIT10/TTRGT0/TOT10 P74/TIT11/TEVTT0/TOT11 P75/TECRT1/AFO			

Caution: The noise elimination function is valid only in the control mode.

(1) Noise elimination control register (NRC)

The NRC register is an 8-bit register that specifies the sampling clock that is used to eliminate digital noise of input pins.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 00H.

Figure 20-82: Noise Elimination Control Register (NRC) (1/2)

After reset: 00H R/W Address: FFFFFtA0H

	7	6	5	4	3	2	1	0
NRC	NRC7	NRC6	NRC5	NRC4	NRC3	NRC2	NRC1	NRC0

NRC7	Noise elimination clock setting for pin group 7[Note]
0	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz)
1	f_{XX}/64 (1 μs @ f_{XX} = 64 MHz)

NRC6	Noise elimination clock setting for pin group 6[Note]
0	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz)
1	f_{XX}/64 (1 μs @ f_{XX} = 64 MHz)

NRC5	Noise elimination clock setting for pin group 5[Note]
0	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz)
1	f_{XX}/64 (1 μs @ f_{XX} = 64 MHz)

NRC4	Noise elimination clock setting for pin group 4[Note]
0	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz)
1	f_{XX}/64 (1 μs @ f_{XX} = 64 MHz)

NRC3	Noise elimination clock setting for pin group 3[Note]
0	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz)
1	f_{XX}/64 (1 μs @ f_{XX} = 64 MHz)

Note: Pin group 3: P10/TIP00/TEVTP1/TOP00, P11/TIP01/TTRGP1/TOP01, P12/TIP10/TTRGP0/TOP10, P13/TIP11/TEVTP0/TOP11
Pin group 4: P14/TIP20/TEVTP3/TOP20, P15/TIP21/TTRGP3/TOP21, P16/TIP30/TTRGP2/TOP30, P17/TIP31/TEVTP2/TOP31
Pin group 5: P20/TIP40/TEVTP5/TOP40, P21/TIP41/TTRGP5/TOP41, P22/TIP50/TTRGP4/TOP50, P23/TIP51/TEVTP4/TOP51
Pin group 6: P24/TIP60/TEVTP7/TOP60, P25/TIP61/TTRGP7/TOP61, P26/TIP70/TTRGP6/TOP70, P27/TIP71/TEVTP6/TOP71
Pin group 7: P60/TOR10/TTRGR1, P61/TOR11/TIR10, P62/TOR12/TIR11, P63/TOR13/TIR12, P64/TOR14/TIR13, P67/TOR17/TEVTR1

Figure 20-82: Noise Elimination Control Register (NRC) (2/2)

NRC2	Noise elimination clock setting for pin group 2[Note]
0	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz)
1	f_{XX}/64 (1 μs @ f_{XX} = 64 MHz)

NRC1	Noise elimination clock setting for pin group 1[Note]
0	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz)
1	f_{XX}/64 (1 μs @ f_{XX} = 64 MHz)

NRC0	Noise elimination clock setting for P00/NMI pin
0	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz)
1	f_{XX}/64 (1 μs @ f_{XX} = 64 MHz)

Note: Pin group 1: P03/INTP2/ADTRG0, P04/INTP3/ADTRG1, P30/RXDC0/INTP4, P32/RXDC1/INTP5, P83/SCS300/INTP6, P84/SCS301/INTP7, P85/SCS302/INTP8, P93/SCS300/INTP9, P94/SCS301/INTP10, P95/SCS302/INTP11

Pin group 2: P100/TCLR0/TICC00/TOP81, P101/TCUD0/TICC01, P102/TIUD0/TO1, P70/TIT00/TEVTT1/TOT00, P71/TIT01/TTRGT1/TOT01, P72/TECRT0/INTP12, P73/TIT10/TTRGT0/TOT10, P74/TIT11/TEVTT0/TOT11, P75/TECRT1/AFO

Cautions: 1. If the input pulse lasts for the duration of 4 to 5 clocks, it is undefined whether the pulse is detected as a valid edge or eliminated as noise. So that the pulse is actually detected as a valid edge, the same pulse level must be input for the duration of 5 clocks or more.

2. If noise is generated in synchronization with the sampling clock, eliminate the noise by attaching a filter to the input pin.

3. Noise is not eliminated if the corresponding pin is used as normal input port pin.

Chapter 21 Reset Function

21.1 Features

- Reset function by $\overline{\text{RESET}}$ input
- Forced reset function by DCU (refer to **Chapter 23 "On-Chip Debug Function (OCD)" on page 969**)
- Reset generator (RG) eliminates noise from the $\overline{\text{RESET}}$ pin.

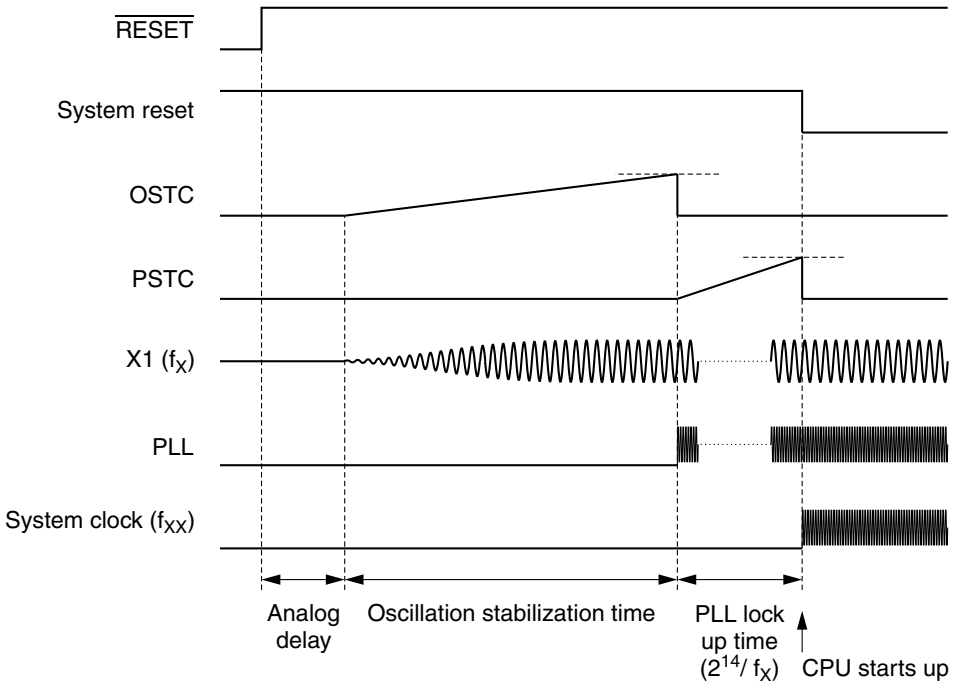
21.2 Configuration

During a system reset, most pins (all except the DCK, $\overline{\text{DRST}}$, DMS, DDI, DDO, $\overline{\text{RESET}}$, X2, V_{DD10} to V_{DD15}, V_{SS10} to V_{SS15}, V_{DD30} to V_{DD37}, V_{SS30} to V_{SS37}, CV_{DD}, CV_{SS}, AV_{DD}, AV_{REF0}, AV_{REF1}, AV_{SS0} and AV_{SS1} pins) enter the high-impedance state.
Therefore, if an external device always requires a defined input level (e.g. external memory) a pull-up (or pull-down) resistor must be connected to each concerned output pin to prevent signal lines from floating. If no resistor is connected, the external device may be destroyed when these pins enter the high-impedance state.

21.3 Operation

When a low-level signal is input to the $\overline{\text{RESET}}$ pin, a system reset is effected and each on-chip hardware is initialized.
When the $\overline{\text{RESET}}$ pin level changes from low to high, the oscillation stabilization time counter (OSTC) is started after analog delay by the reset generator. At that time the OSTC elapses, the PLL circuit will be enabled and the internal PLL stabilization time counter (PSTC) is started using the oscillator output clock (f_X). After 2^{14} oscillator clocks (f_X) the PLL output clock becomes the system clock (f_{XX}) and the internal system reset is released synchronously to the system clock.

Figure 21-1: Reset Timing



Remarks: 1. If no clock is supplied to V850E/PH2 (i.e. the oscillator does not work) the internal system reset will not be released independently from input level of the external $\overline{\text{RESET}}$ pin.

2. The on-chip debug function can force the activation of the system reset independently from input level of the $\overline{\text{RESET}}$ pin.

Chapter 22 Internal RAM Parity Check Function

The V850E/PH2 microcontroller is provided with a parity check function for the internal RAM (iRAM).

22.1 Features

- Maskable interrupt (INTPERR) on detection of parity error
- Indication of internal RAM address of detected parity error
- Indication of erroneous byte within 32-bit word

22.2 Operation

Each byte of data stored in the internal RAM is checked by its parity bit. A maskable interrupt (INTPERR) is generated, if a parity mismatch is detected on iRAM read operation. In this case the address of the erroneous data is latched in the RAMPADD register and the erroneous byte(s) are indicated in the RAMERR register.

Caution: Ensure that all internal RAM data is initialized on a word (32-bit) base by a write operation before first read access is made. It is important to initialize the whole memory word, even if only byte or half word is used. Otherwise a parity fail may be indicated mistakenly.

22.3 Control Registers

(1) Internal RAM parity error status register (RAMERR)

The RAMERR register is an 8-bit register that reflects the parity error flags of the four bytes of one word (32 bits) in the internal RAM. The corresponding error flag (bits RAE0 to RAE3) is set and a maskable interrupt (INTPERR) is generated, if a parity error is detected during read access.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

Figure 22-1: Internal RAM Parity Error Status Register (RAMERR)

After reset: 00H R/W Address: FFFFF4C0H

	7	6	5	4	3	2	1	0
RAMERR	0	0	0	0	RAE3	RAE2	RAE1	RAE0

RAEn	Internal RAM Parity Error Flag
0	No parity error detected in internal RAM.
1	Parity error detected in internal RAM for byte position n.

n	Bit Name	Function
0	RAE0	Parity error caused by bits 0 to 7
1	RAE1	Parity error caused by bits 8 to 15
2	RAE2	Parity error caused by bits 16 to 23
3	RAE3	Parity error caused by bits 24 to 31

Remark: The RAEn bit can be both read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it.

(2) Internal RAM parity error address register (RAMPADD)

The RAMPAD register is a 16-bit register that latches the internal RAM address causing the first parity error after hardware reset was released or RAMERR register was cleared.
This register can be read or written in 16-bit units.
Reset input clears this register to 8000H.

Caution: Do not read the RAMPADD register, when all internal RAM parity error flags RAEn (n = 0 to 3) are cleared. If a parity error is detected and the RAMPADD register is read before the respective RAEn flag is set, the read value might be invalid.

Figure 22-2: Internal RAM Parity Error Address Register (RAMPADD)

After reset: 8000H R/W Address: FFFFF4C2H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAMPADD	1	RAMPADD14	RAMPADD13	RAMPADD12	RAMPADD11	RAMPADD10	RAMPADD9	RAMPADD8	RAMPADD7	RAMPADD6	RAMPADD5	RAMPADD4	RAMPADD3	RAMPADD2	0	0

RAMPAD14 to RAMPADD2	Internal RAM Parity Error Address
	Internal RAM address of the 32-bit word causing the parity error.
Caution:	**Bit 15 of the RAMPADD register is always 1. This does not reflect the correct address bit 15 of the internal RAM, which starts at location FFF0000H.**
Remark:	Bits 0 and 1 of the RAMPADD register are always 0, because the parity check function is aligned on 32-bit words.

[MEMO]

Chapter 23 On-Chip Debug Function (OCD)

An on-chip debug unit is provided in the V850E/PH2 microcontroller and realizes stand-alone on-chip debugging of the V850E/PH2 microcontroller by connecting a N-Wire type emulator.

Caution: The debug function explained in this chapter is the function that can be realized by using the V850E/PH2 microcontroller, the NEC Electronics' IE-V850E1-CD-NW (N-Wire type emulator). When using a third-party N-Wire type emulator, refer to the manual for the debugger used.

23.1 Function Overview

23.1.1 On-chip debug unit type

The on-chip debug unit incorporated in the V850E/PH2 microcontroller is RCU1 (run control unit 1). The on-chip unit incorporated differs depending on the microcontroller, and also features different functions.

23.1.2 Debug function

For details of the debug function, refer to the corresponding debugger operation user's manual.

(1) Debug interface

This interface establishes communication with the host machine by using the $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO signals, via a N-Wire type emulator. The communication specifications of N-Wire are used for this interface. It does not support a boundary scan function.

(2) On-chip debug

On-chip debugging can be performed by providing wiring and connectors for debugging on the target system.
Connect a N-Wire type emulator to the emulator connector.

(3) Forced reset function

The V850E/PH2 can be forcibly reset.

(4) Break reset function

The CPU can be started in the debug mode immediately after resetting the CPU has been cleared.

(5) Forced break function

Execution of the user program can be forcibly stopped (however, the handler of the illegal instruction code exception (first address: 00000060H) cannot be used).

(6) Hardware break function

Two common instruction fetch/access breakpoints can be used. By using the instruction breakpoint, program execution can be suspended at an arbitrary address. By using the access breakpoint, program execution can be suspended by data-accessing an arbitrary address.

(7) Software break function

In addition to the hardware break function, a software break function is available. Up to eight software breakpoints can be set in the internal ROM area. The number of software breakpoints that can be set in the internal RAM area differs depending on the debugger used.

(8) Debug monitor function

During debugging, a memory space for debugging that differs from the user memory space is used (background monitor format). The user program can be executed starting from any address. While execution of the user program is stopped, the user resources (such as memory and I/O) can be read or written, and the user program can be downloaded.

(9) Mask function

$\overline{\text{RESET}}$, $\overline{\text{WAIT}}$, NMI and all maskable interrupt request signals can be masked.

(10) Timer function

The execution time of the user program can be measured.

23.2 Connection with N-Wire Type Emulator

To connect a N-Wire type emulator, it is necessary to mount an emulator connector and circuit for connection on the target system.
Select either the KEL connector, MICTOR connector (Part number: 2-767004-2, distributor: Tyco Electronics AMP K.K.), or 2.54 mm pitch 20-pin general-purpose connector as the emulator connector. Connectors other than the KEL connector may not be supported, depending on the emulator, so when using a connector, refer to the manual of the emulator used.

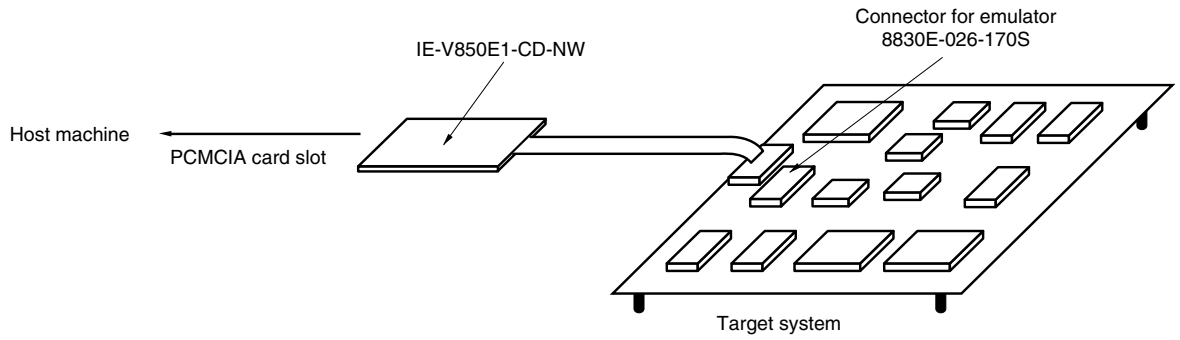
23.2.1 KEL connector

When the IE-V850E1-CD-NW is used, use of the following connector is recommended.

Part number

- 8830E-026-170S: Straight type
- 8830E-026-170L: Right-angle type

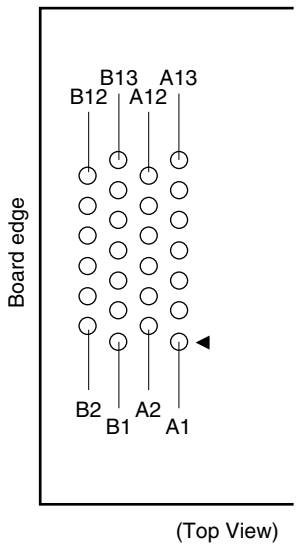
Figure 23-1: Connecting N-Wire Type Emulator (IE-V850E1-CD-NW (N-Wire Card))



(1) Pin configuration

Figure 23-2 shows the pin configuration of the emulator connector (target system side), and Table 23-1 shows the pin functions.

Figure 23-2: Pin Configuration of Emulator Connector (on Target System Side)



Caution: Design the board based on the dimensions of the connector when actually mounting the connector on the board.

(2) Pin functions

The following table shows the pin functions of the emulator connector (on the target system side).

Table 23-1: Pin Functions of Connector for IE-V850E1-CD-NW (on Target System Side)

Pin No.	Pin Name	I/O	Pin Function
A1	(Reserved 1)	-	(Connect to GND)
A2	(Reserved 2)	-	(Connect to GND)
A3	(Reserved 3)	-	(Connect to GND)
A4	(Reserved 4)	-	(Connect to GND)
A5	(Reserved 5)	-	(Connect to GND)
A6	(Reserved 6)	-	(Connect to GND)
A7	DDI	Input	Data input for N-Wire interface
A8	DCK	Input	Clock input for N-Wire interface
A9	DMS	Input	Transfer mode select input for N-Wire interface
A10	DDO	Output	Data output for N-Wire interface
A11	DRST	Input	On-chip debug unit reset input
A12	(Reserved 7)	-	(Leave open)
A13	FLMD0	Input	Control signal for flash memory downloading
B1	GND	-	-
B2	GND	-	-
B3	GND	-	-
B4	GND	-	-
B5	GND	-	-
B6	GND	-	-
B7	GND	-	-
B8	GND	-	-
B9	GND	-	-
B10	GND	-	-
B11	(Reserved 8)	-	(Connect to GND)
B12	(Reserved 9)	-	(Connect to GND)
B13	VDD	-	3.3 V input (for monitoring power application to target)

Cautions: 1. The processing of the pins not incorporated in the V850E/PH2 or unused pins depends on the emulator used.

2. The pattern on the target board must satisfy the following conditions.

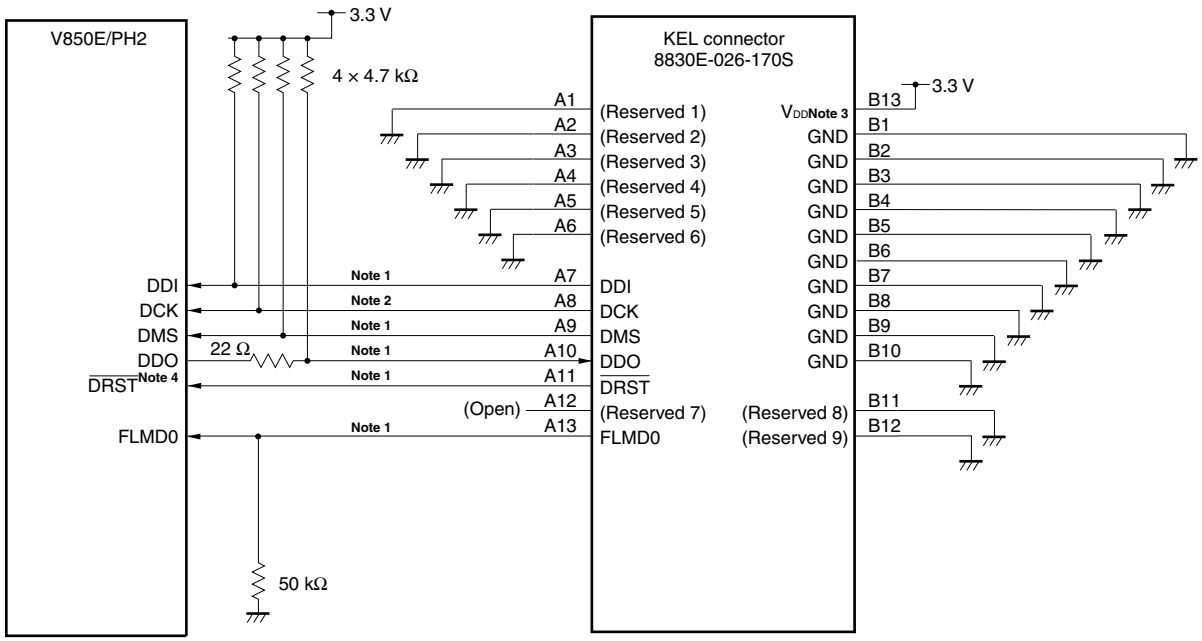
- **Keep the pattern length to within 100 mm.**
- **Shield the clock signal with GND.**

Remark: Input/output is as viewed from the device side.

(3) Recommended circuit example

The following figure shows an example of the recommended circuit of the emulator connector (on the target system side).

Figure 23-3: Example of Recommended Emulator Connection of V850E/PH2



Notes: 1. Keep the pattern length to within 100 mm.

2. Shield the DCK signal with GND.

3. For detecting power supply to the target board. Connect to the N-Wire interface voltage.

4. When $\overline{\text{DRST}}$ pin is high level: On-chip debug mode
When $\overline{\text{DRST}}$ pin is low level or open: Normal operation mode
$\overline{\text{DRST}}$ pin is connected to V_{SS3} via an internal pull-down resistor

Cautions: 1. The DDO signal is 3.3 V output, and the input level of the DDI, DCK, DMS, and $\overline{\text{DRST}}$ signals is TTL level.

2. A 3.3 V interface may not be supported, so a level shifter may be required by some N-Wire type emulators. Refer to the manual of the emulator used. Note that the IE-V850E1-CD-NW supports a 5 V interface.

23.3 Precautions

<1> The flash memory of the device used in debugging is rewritten during debugging, so the number of flash memory rewrites cannot be guaranteed. Therefore, do not use the device used in debugging for a mass production product.

<2> If a reset ($\overline{\text{RESET}}$ signal input from the target system or reset input by an internal reset source) occurs during RUN (program execution), the break function may malfunction.

<3> Even if reset is masked by using the mask function, the I/O buffers (port pins, etc.) are set to the reset state when the $\overline{\text{RESET}}$ signal is input.

<4> $\overline{\text{RESET}}$ signal input during a break is masked.

[MEMO]

Chapter 24 Flash Memory

The V850E/PH2 and has a 512 KB on-chip flash memory configured as 128 blocks of 4 KB block size.

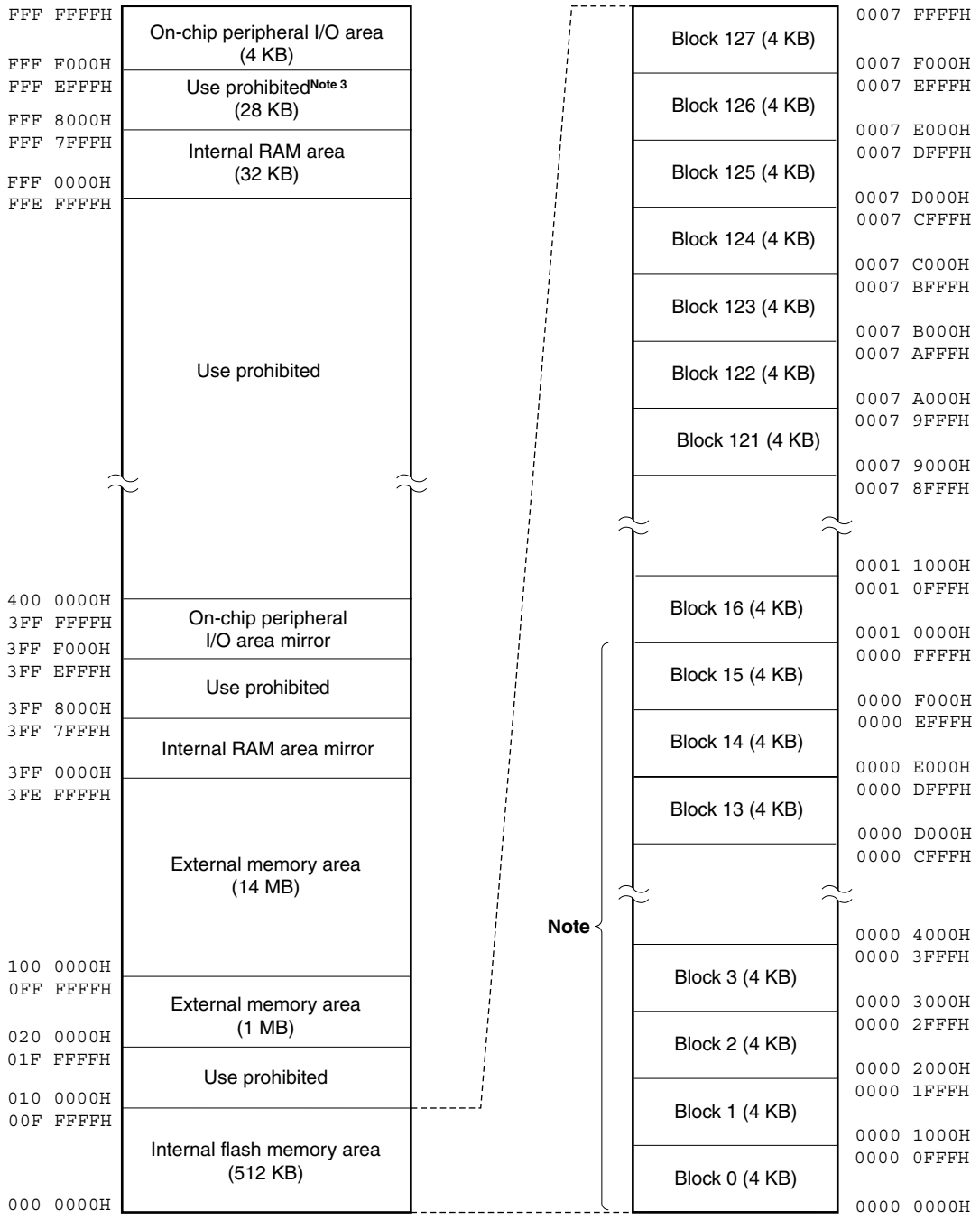
24.1 Features

- 4-byte/1-clock access (when instruction is fetched)
- Capacity: 512 KB
- Block size: 128 blocks of 4 KB
- Write voltage: Erase/write with single voltage
- Rewriting method
  - Rewriting by communication with dedicated flash programmer via serial interface (on-board/off-board programming)
  - Rewriting flash memory by user program (self-programming)
- 64 KB boot block cluster with write prohibit function supported (protection function)
- Interrupts can be acknowledged during self programming.

24.2 Memory Configuration

The 512 KB internal flash memory area is divided into 128 blocks and can be programmed/erased in block units. All the blocks can also be erased at once.

Figure 24-1: Flash Memory Mapping



Note: Blocks 0 to 15 (64 KB): Boot block cluster

24.3 Functional Outline

The internal flash memory of the V850E/PH2 can be rewritten by using the rewrite function of the dedicated flash programmer, regardless of whether the V850E/PH2 has already been mounted on the target system or not (off-board/on-board programming).
In addition, different functions are implemented to protect unwanted Flash access via the programmer interface and to protect the boot area from any unwanted modification
The rewrite function using the user program (self programming) is ideal for an application where it is assumed that the program is changed or extended during or after production/shipment of the target system. Interrupt servicing is supported during self programming, so that the flash memory can be rewritten under various conditions, such as while communicating with an external device.

Table 24-1: Rewrite Method

Rewrite Method	Functional Outline	Operation Mode
Off-board programming	Flash memory can be rewritten before the device is mounted on the target system, by using a dedicated flash programmer and a dedicated program adapter board (e.g. FA series).	Flash memory programming mode
On-board programming	Flash memory can be rewritten after the device is mounted on the target system, by using a dedicated flash programmer.	
Self-programming	Flash memory can be rewritten by executing a user program that has been written to the flash memory in advance by means of off-board/on-board programming. (During self-programming, instructions cannot be fetched from or data access cannot be made to the internal flash memory area. Therefore, the rewrite program must be transferred to the internal RAM or external memory in advance). parts of the self-programming program	Normal operation mode

Remark: The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 24-2: Basic Functions

Function	Functional Outline	Support by	
		On-Board/Off-Board Programming	Self Programming
Block erasure	The contents of specified memory blocks are erased.	yes	yes
Chip erasure	The contents of the entire memory area are erased all at once.	yes	no
Write	Writing to specified addresses, and a verify check to see if write level is secured are performed.	yes	yes
Verify/check sum	Data read from the flash memory is compared with data transferred from the flash programmer.	yes	no (Can be read by user program)
Blank check	The erasure status of the entire memory is checked.	yes	yes
Read	The content of specified addresses can be read.	yes	yes
Protection setting	Setting of protection flags to prohibit programming interface commands (write, block erase, chip erase and read) and to prohibit boot block cluster modification via self-programming	yes	yes

The following Table 24-3 lists the protection functions. After shipment no protection feature is set on the device. Furthermore, after chip erase by a dedicated programmer (PG-FP4) the protection is reset. Each protection function can be used in combination with the others at the same time.

Table 24-3: Protection Functions

Function	Functional Outline	Operation	
		On-Board/Off-Board Programming	Self-Programming
Block erase command prohibit	Execution of a block erase command on all blocks is prohibited. Setting of prohibition can be initialized by execution of a chip erase command.	Block erase command: × Chip erase command: ⊕ Program command: ⊕	Can always be read or rewritten regardless of protection function setting
Chip erase command prohibit	Execution of block erase and chip erase commands on all the blocks is prohibited. Once prohibition is set, setting of prohibition cannot be initialized because the chip erase command cannot be executed.	Block erase command: × Chip erase command: × Program command: ⊕	
Program command prohibit	Write and block erase commands on all the blocks are prohibited. Setting of prohibition can be initialized by execution of the chip erase command.	Block erase command: × Chip erase command: ⊕ Program command: ×	
Read command prohibit	Read of memory content is prohibited.	Read command: ×	

Remark: ×: Operation prohibited
⊕: Operation executable

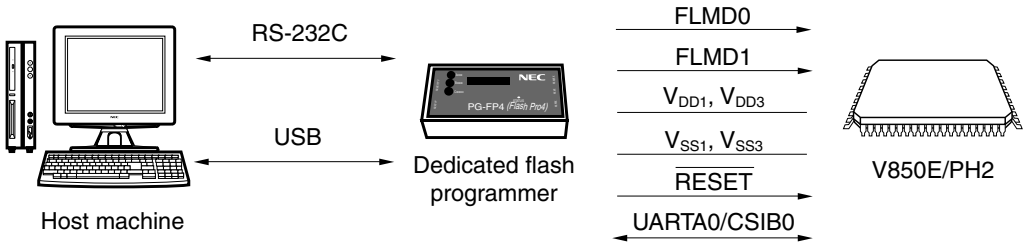
24.4 Rewriting by Dedicated Flash Programmer

The flash memory can be rewritten by using a dedicated flash programmer after the V850E/PH2 is mounted on the target system (on-board programming). The flash memory can also be rewritten before the device is mounted on the target system (off-board programming) by using a dedicated program adapter (FA series).

24.4.1 Programming environment

The following shows the environment required for writing programs to the flash memory of the V850E/PH2:

Figure 24-2: Environment Required for Writing Programs to Flash Memory



A host machine is required for controlling the dedicated flash programmer.
UARTC0 or CSIB0 is used for the interface between the dedicated flash programmer and the V850E/PH2 to perform writing, erasing, etc. A dedicated program adapter (FA series) is required for off-board writing.

Remark: The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

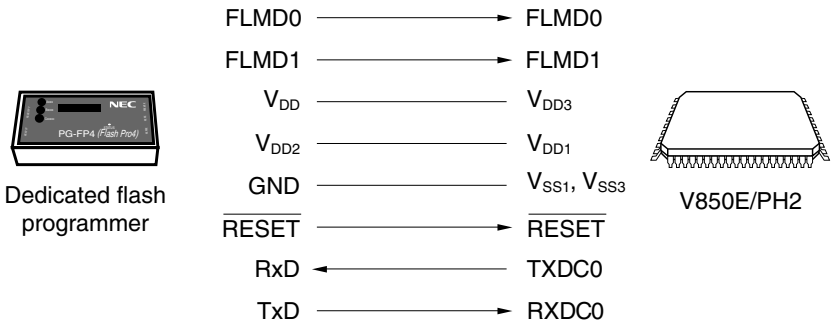
24.4.2 Communication mode

Communication between the dedicated flash programmer and the V850E/PH2 is performed by serial communication using the UARTC0 or CSIB0 interfaces of the V850E/PH2.

(1) UARTA0

Transfer rate: 9,600 to 153,600 bps

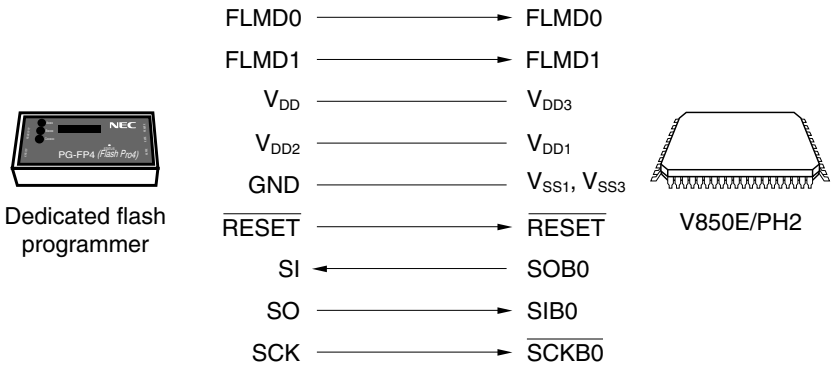
Figure 24-3: Communication with Dedicated Flash Programmer (UARTC0)



(2) CSIB0

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

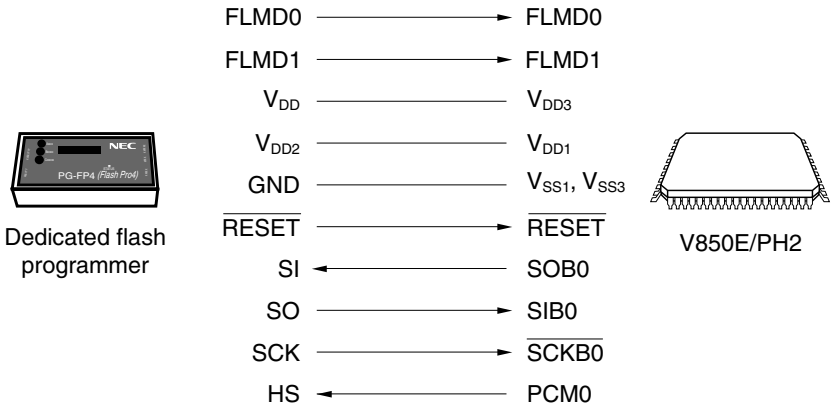
Figure 24-4: Communication with Dedicated Flash Programmer (CSIB0)



(3) CSIB0 + HS

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

Figure 24-5: Communication with Dedicated Flash Programmer (CSIB0 + HS)



The dedicated flash programmer outputs the transfer clock, and the V850E/PH2 operates as a slave.

When the PG-FP4 is used as the dedicated flash programmer, it generates the following signals to the V850E/PH2. For details, refer to the PG-FP4 User's Manual (U15260E).

Table 24-4: Signal Connections of Dedicated Flash Programmer (PG-FP4)

PG-FP4			V850E/PH2	Processing for Connection		
Signal Name	I/O	Pin Function	Pin Name	UARTC0	CSIB0	CSIB0 + HS
FLMD0	Output	Write enable/disable	FLMD0	⊕	⊕	⊕
FLMD1	Output	Write enable/disable	FLMD1	⊕ **Note 1**	⊕ **Note 1**	⊕ **Note 1**
VDD	–	V_{DD} voltage generation/ Voltage monitor	VDD3x	⊕ **Note 2**	⊕ **Note 2**	⊕ **Note 2**
VDD2	–	V_{DD2} voltage generation	VDD1x	× **Note 3**	× **Note 2**	× **Note 2**
GND	–	Ground	VSS1x, VSS3x	⊕	⊕	⊕
CLK	Output	Clock output	X1, X2	× **Note 4**	× **Note 3**	× **Note 3**
$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	⊕	⊕	⊕
SI/RXD	Input	Receive signal	SIB0/RXDC0	⊕	⊕	⊕
SO/TXD	Output	Transmit signal	SOB0/TXDC0	⊕	⊕	⊕
SCK	Output	Transfer clock	$\overline{\text{SCKB0}}$	×	⊕	⊕
HS	Input	Handshake signal	PCM0	×	×	⊕

Notes: 1. Connect to GND via pull-down resistor.

2. Connect these pins to supply power from the PG-FP4. When power is supplied externally to the target board, the voltage is monitored by PG-FP4.

3. Connect these pins to supply power from the PG-FP4, or supply power externally to the target board.

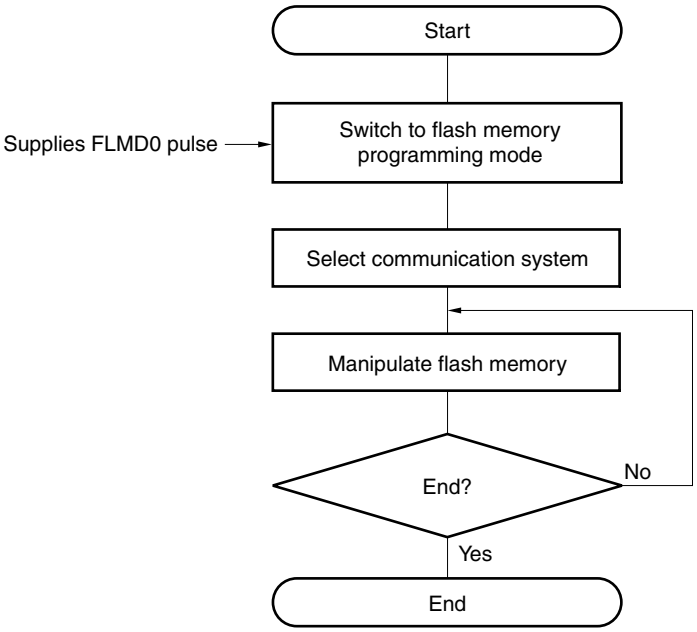
4. Clock supply is provided by an oscillator on the target board. Clock supply from PG-FP4 is not supported for V850E/PH2.

Remark: ⊕: Must be connected.
×: Do not need to be connected.

24.4.3 Flash memory control

The following shows the procedure for manipulating the flash memory.

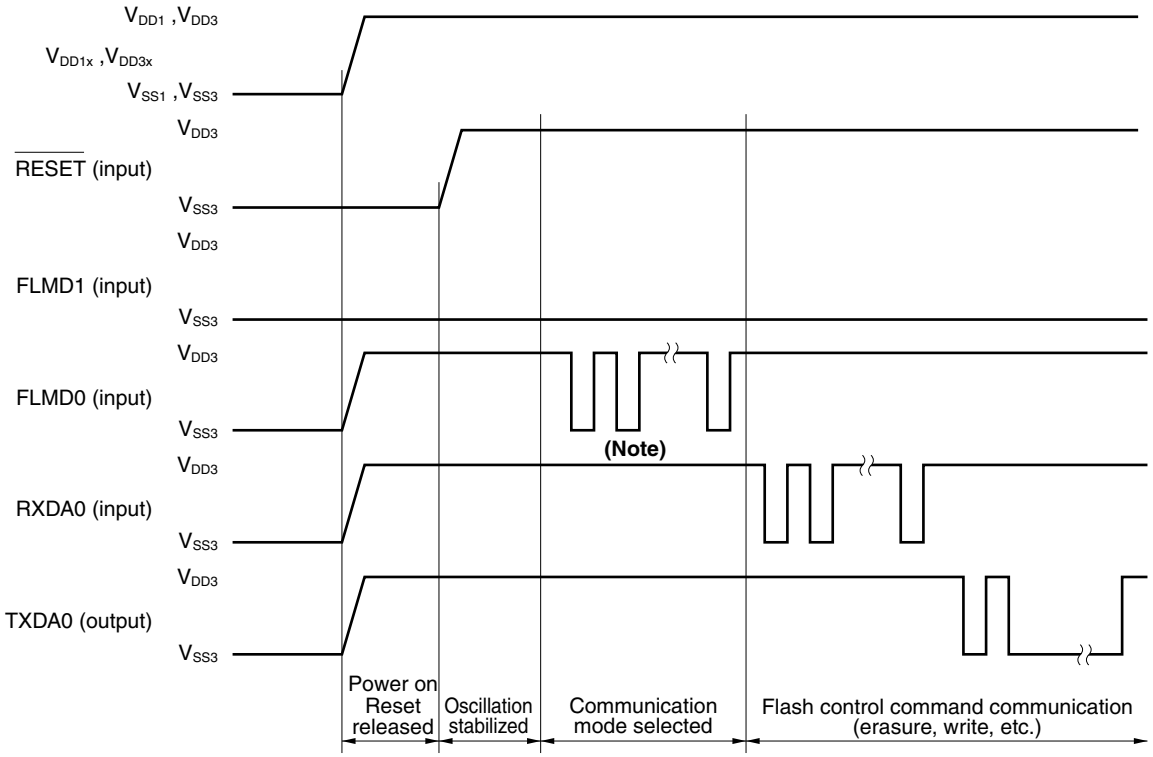
Figure 24-6: Procedure for Manipulating Flash Memory



24.4.4 Selection of communication mode

In the V850E/PH2, the communication mode is selected by inputting pulses (11 pulses max.) to the FLMD0 pin after switching to the flash memory programming mode. The FLMD0 pulse is generated by the dedicated flash programmer.
The following shows the relationship between the number of pulses and the communication mode.

Figure 24-7: Selection of Communication Mode



Note: The number of clocks is as follows depending on the communication mode.

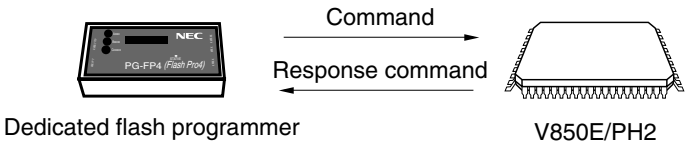
Number of FLMD0 Pulses	Communication Mode	Remarks
0	UARTC0	Communication rate: 9,600 bps (after reset), LSB first
8	CSIB0	V850E/PH2 performs slave operation, MSB first
11	CSIB0 + HS	
Others	RFU	Setting prohibited

Caution: When UARTC0 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash programmer after receiving the FLMD0 pulse.

24.4.5 Communication commands

The V850E/PH2 communicates with the dedicated flash programmer by means of commands. The signals sent from the dedicated flash programmer to the V850E/PH2 are called “commands”. The response signals sent from the V850E/PH2 to the dedicated flash programmer are called “response commands”.

Figure 24-8: Communication Commands



The following shows the commands for flash memory control in the V850E/PH2. All of these commands are issued from the dedicated flash programmer, and the V850E/PH2 performs the processing corresponding to the commands.

Table 24-5: Communication Commands

Classification	Command Name	Support			Function
		UARTC0	CSI0	CSI0 + HS	
Blank check	Block blank check command	⊕	⊕	⊕	Checks if the contents of the memory in the specified block have been correctly erased.
Erase	Chip erase command	⊕	⊕	⊕	Erases the contents of the entire memory.
	Block erase command	⊕	⊕	⊕	Erases the contents of the memory of the specified block.
Write	Write command	⊕	⊕	⊕	Writes the specified address range, and executes a contents verify check.
Verify	Verify command	⊕	⊕	⊕	Compares the contents of memory in the specified address range with data transferred from the flash programmer.
	Checksum command	⊕	⊕	⊕	Reads the checksum in the specified address range.
Read	Read command	⊕	⊕	⊕	Reads the specified address range.
System setting, control	Silicon signature command	⊕	⊕	⊕	Reads silicon signature information.
	protection setting command	⊕	⊕	⊕	Disables the chip erase command, enables the block erase command, and disables the write command.

Remark: ×: Operation not supported
⊕: Operation supported

24.4.6 Pin connection

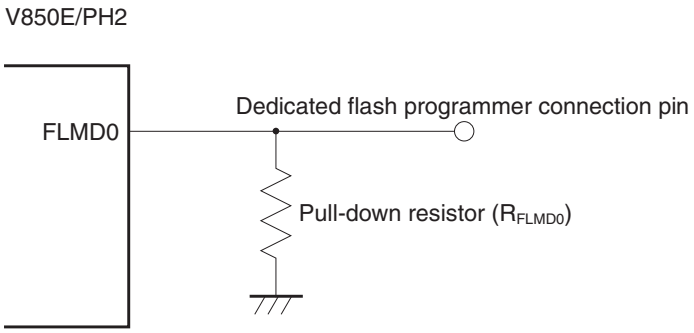
When performing on-board writing, mount a connector on the target system to connect to the dedicated flash programmer. Also, incorporate a function on-board to switch from the normal operation mode to the flash memory programming mode.
In the flash memory programming mode, all the pins not used for flash memory programming become the same status as that immediately after reset. Therefore, pin handling is required when the external device does not acknowledge the status immediately after a reset.

(1) FLMD0 pin

In the normal operation mode, input a voltage of V_{SS3} level to the FLMD0 pin. In the flash memory programming mode, supply a write voltage of V_{DD3} level to the FLMD0 pin.

Because the FLMD0 pin serves as a write protection pin in the self programming mode, a voltage of V_{DD3} level must be supplied to the FLMD0 pin via port control, etc., before writing to the flash memory. For details, refer to the self-programming application note (U16929E).

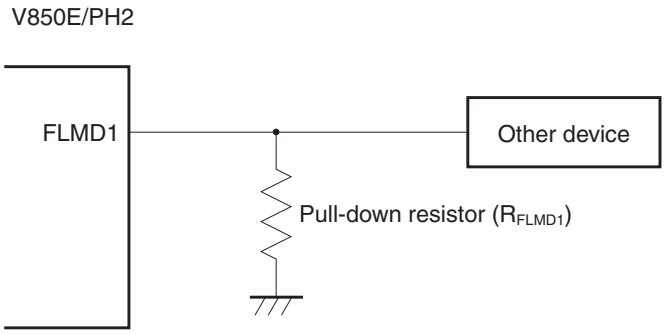
Figure 24-9: FLMD0 Pin Connection Example



(2) FLMD1 pin

When 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. When V_{DD3} is supplied to the FLMD0 pin, the flash memory programming mode is entered, so 0 V must be input to the FLMD1 pin. The following shows an example of the connection of the FLMD1 pin.

Figure 24-10: FLMD1 Pin Connection Example



Caution: If the V_{DD3} signal is input to the FLMD1 pin from another device during on-board writing and immediately after reset, isolate this signal.

Table 24-6: Relationship Between FLMD0 and FLMD1 Pins and Operation Mode when Reset is Released

FLMD0	FLMD1	Operation Mode
0	Don't care	Normal operation mode
V_{DD3}	0	Flash memory programming mode
V_{DD3}	V_{DD3}	Setting prohibited

(3) Serial interface pin

The following shows the pins used by each serial interface.

Table 24-7: Pins Used by Serial Interfaces

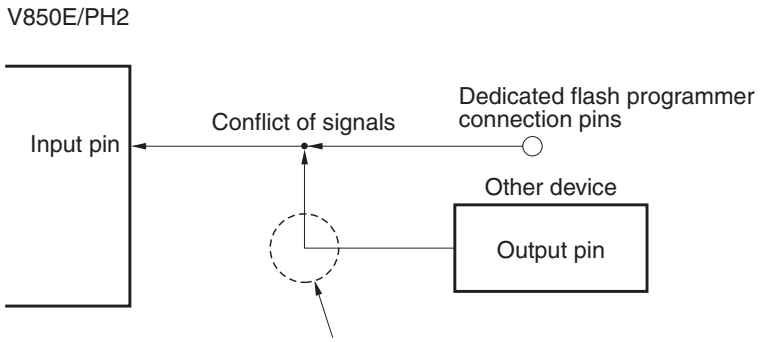
Serial Interface	Used Pins
UARTC0	TXDC0, RXDC0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, PCM0

When connecting a dedicated flash programmer to a serial interface pin that is connected to another device on-board, care should be taken to avoid conflict of signals and malfunction of the other device.

(a) Conflict of signals

When the dedicated flash programmer (output) is connected to a serial interface pin (input) that is connected to another device (output), a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the output high-impedance status.

Figure 24-11: Conflict of Signals (Serial Interface Input Pin)

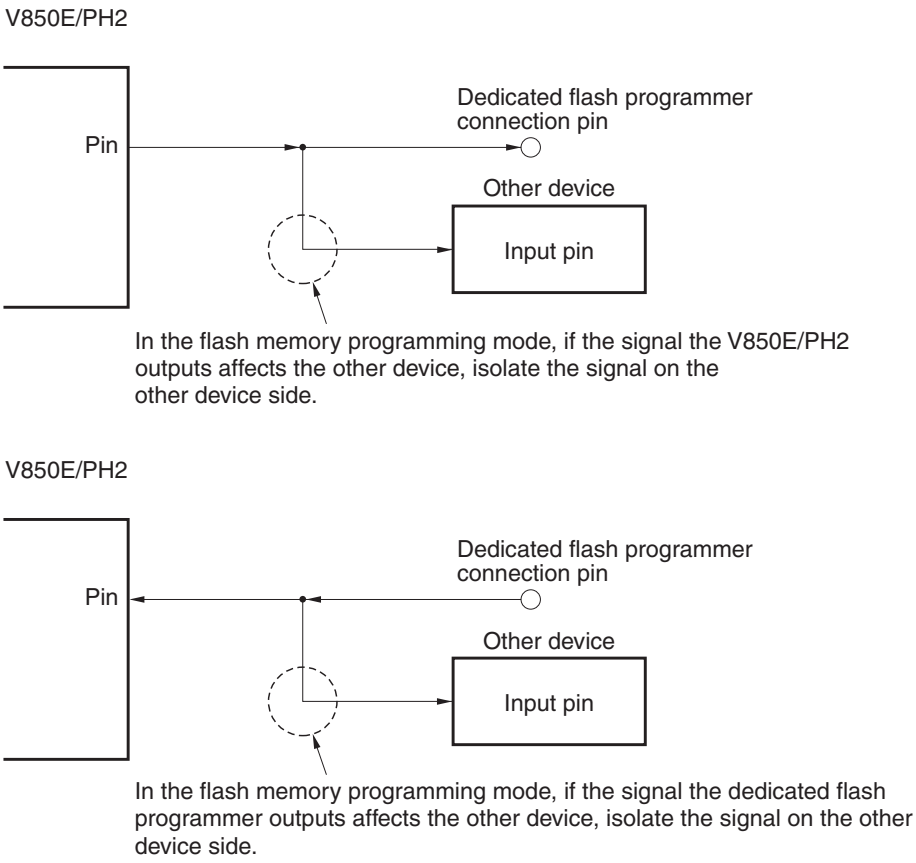


In the flash memory programming mode, the signal that the dedicated flash programmer sends out conflicts with signals another device outputs. Therefore, isolate the signals on the other device side.

(b) Malfunction of other device

When the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) that is connected to another device (input), the signal is output to the other device, causing the device to malfunction. To avoid this, isolate the connection to the other device.

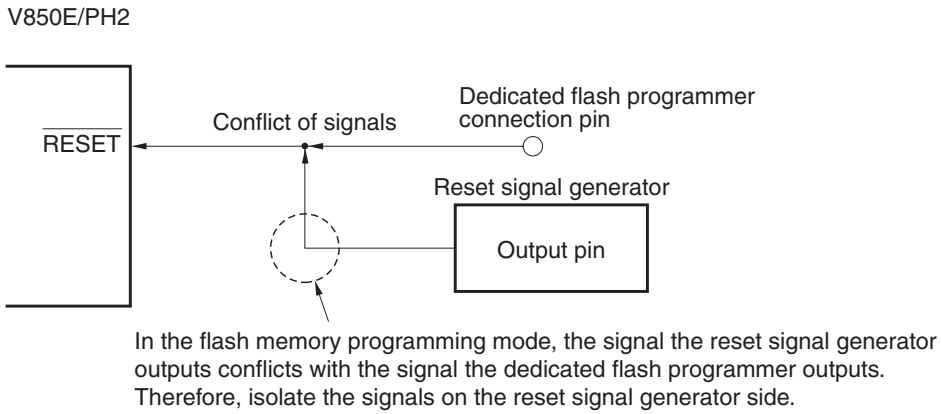
Figure 24-12: Malfunction of Other Device



(4) $\overline{\text{RESET}}$ pin

When the reset signals of the dedicated flash programmer are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on-board, a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the reset signal generator.
When a reset signal is input from the user system in the flash memory programming mode, the programming operation will not be performed correctly. Therefore, do not input signals other than the reset signals from the dedicated flash programmer.

Figure 24-13: Conflict of Signals ($\overline{\text{RESET}}$ Pin)



(5) Port pins (including NMI)

When the system shifts to the flash memory programming mode, all the pins that are not used for flash memory programming are in the same status as that immediately after reset. If the external device connected to each port does not recognize the status of the port immediately after reset, pins require appropriate processing, such as connecting to V_{DD3} via a resistor or connecting to V_{SS3} via a resistor.

(6) Other signal pins

Connect X1 and X2 in the same status as that in the normal operation mode.
During flash memory programming, input a low level to the $\overline{\text{DRST}}$ pin or leave it open. Do not input a high level.

(7) Power supply

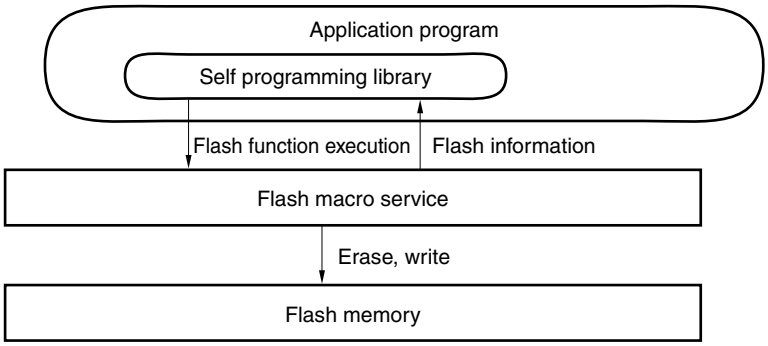
Supply the same power (V_{DD1}, V_{SS1}, V_{DD3}, V_{SS3}, CV_{DD}, CV_{SS}, AV_{DD}, AV_{SS}, AV_{REF0}, AV_{REF1}) as in normal operation mode.

24.5 Rewriting by Self Programming

24.5.1 Overview

The V850E/PH2 supports a flash macro service that allows the user program to rewrite the internal flash memory by itself. By using this interface and a self-programming library that is used to rewrite the flash memory with a user application program, the flash memory can be rewritten by a user application transferred in advance to the internal RAM or external memory. Consequently, the user program can be upgraded and constant data can be rewritten in the field.

Figure 24-14: Concept of Self Programming



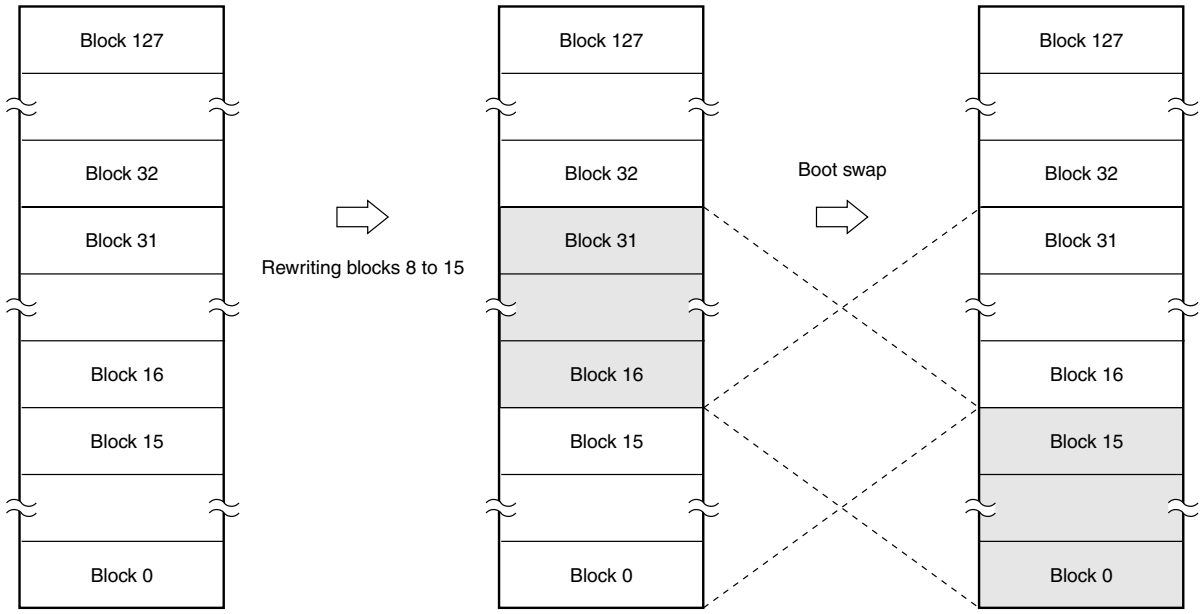
For further details, refer to the self-programming application note (U16929E).

24.5.2 Features

(1) Secure self-programming (boot swap function)

The V850E/PH2 supports a boot swap function that can exchange the physical memory of blocks 0 to 15 with the physical memory of blocks 16 to 31. By writing the start program to be rewritten to blocks 16 to 31 in advance and then swapping the physical memory, the entire area can be safely rewritten even if a power failure occurs during rewriting because the correct user program always exists in blocks 0 to 15.

Figure 24-15: Rewriting Entire Memory Area (Boot Swap)



(2) Interrupt support

Instructions cannot be fetched from the flash memory during self programming. Conventionally, therefore, a user handler written to the flash memory could not be used even if an interrupt occurred. With the V850E/PH2, a user handler can be registered to an entry RAM area by using a library function, so that interrupt servicing can be performed by internal RAM or external memory execution.

[MEMO]

Chapter 25 Electrical Specifications

25.1 Absolute Maximum Ratings

Table 25-1: Absolute Maximum Ratings

(T_A = 25°C, V_{SS1} [Note 1] = 0 V)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD1}	**Note 2**	-0.5 to +2.0	V
	V_{DD3}	**Note 4**	-0.5 to +4.6	V
	CV_{DD}		-0.5 to +2.0	V
	AV_{DD}	V_{DD3} - 0.5 V < AV_{DD} < V_{DD3} + 0.5 V	-0.5 to +4.6	V
	V_{SS1}	**Note 1**	-0.5 to +0.5	V
	V_{SS3}	**Note 3**	-0.5 to +0.5	V
	CV_{SS}		-0.5 to +0.5	V
	AV_{SS}	**Note 5**	-0.5 to +0.5	V
Input voltage	V_{I1}	All pins except X1 pin, ANI00 to ANI19 pins V_{I1} < V_{DD3} + 0.3 V	-0.5 to +4.6	V
	V_{I2}	X1 pin V_{I2} < CV_{DD} + 0.5 V	-0.5 to +2.0	V
Analog input voltage	V_{IAN}	ANI00 to ANI19 pins AV_{DD} = 3.0 V to 3.6 V	-0.3 to AV_{DD} + 0.3	V
Analog reference input voltage	AV_{REF0}, AV_{REF1}		-0.3 to AV_{DD} + 0.3	V
Output current, low	I_{OL}	Per pin	4.0	mA
		Total of all pins	100	mA
Output current, high	I_{OH}	Per pin	-4.0	mA
		Total of all pins	-100	mA
Operating temperature	T_A	µPD70F3187	-40 to +85	°C
		µPD70F3187(A1)	-40 to +110	°C
		µPD70F3187(A2)	-40 to +125	°C
Storage temperature	T_{stg}		-65 to +150	°C

Notes: 1. V_{SS1} applied to V_{SS10} to V_{SS15} pins.

2. V_{DD1} applied to V_{DD10} to V_{DD15} pins.

3. V_{SS3} applied to V_{SS30} to V_{SS37} pins.

4. V_{DD3} applied to V_{DD30} to V_{DD37} pins.

5. AV_{SS} applied to AV_{SS0} to AV_{SS1} pins.

Cautions: 1. Do not directly connect output (or I/O) pins of IC products to each other, or to V_{DD}, V_{SS}, and GND.

2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded. The ratings and conditions shown below for DC characteristics and AC characteristics are within the range for normal operation and quality assurance.

25.2 General Characteristics

Unless specified otherwise, the following conditions are assumed for all characteristics in this chapter.

$V_{DD3x} = AV_{DD}$ = 3.0 V to 3.6 V
$V_{DD1x} = CV_{DD}$ = 1.35 V to 1.65 V
$V_{SS1x} = CV_{SS} = V_{SS3x} = AV_{SSx}$ = 0 V
µPD70F3187: T_A = -40 °C to +85 °C
µPD70F3187(A1): T_A = -40 °C to +110 °C
µPD70F3187(A2): T_A = -40 °C to +125 °C

25.2.1 Capacitance

Table 25-2: Capacitance

(T_A = 25°C, $V_{DD1x} = CV_{DD} = V_{DD3x} = AV_{DD} = V_{SS1x} = CV_{SS} = V_{SS3x} = AV_{SSx}$ = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f_C = 1 MHz Un-measured pins returned to 0 V.			15	pF
Output capacitance	C_O				15	pF
I/O capacitance	C_{IO}				15	pF

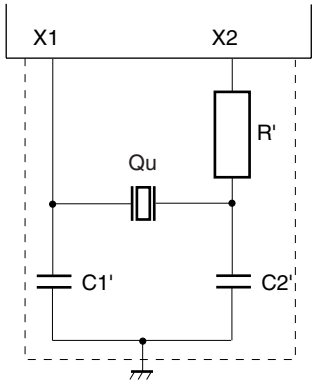
25.2.2 Operating conditions

Table 25-3: Operating Conditions

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Internal system clock frequency	f_{XX}	f_{OSC} = 16 MHz		64		MHz

25.2.3 Oscillator characteristics

Figure 25-1: Oscillator Recommendations



Remark: Values of capacitors C1', C2' and R' depend on used crystal or resonator and must be specified in cooperation with the manufacturer.

Cautions:

1. **External clock input is prohibited.**
2. **Wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.**
   - **Keep the wiring length as short as possible.**
   - **Do not cross the wiring with the other signal lines.**
   - **Do not route the wiring near a signal line through which a high fluctuating current flows.**
   - **Always make the ground point of the oscillator capacitor the same potential as CV_{SS}.**
   - **Do not ground the capacitor to a ground pattern through which a high current flows.**
   - **Do not fetch signals from the oscillator.**

Table 25-4: Oscillator Characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_{OSC}			16		MHz
Oscillation stabilization time	t_{OST}	f_{OSC} = 16 MHz			4.096	ms

25.3 DC Characteristics

Unless specified otherwise, the following conditions are assumed for all characteristics in this chapter.

$V_{DD3x} = AV_{DD}$ = 3.0 V to 3.6 V
$V_{DD1x} = CV_{DD}$ = 1.35 V to 1.65 V
$V_{SS1x} = CV_{SS} = V_{SS3x} = AV_{SSx}$ = 0 V
µPD70F3187: T_A = -40 °C to +85 °C
µPD70F3187(A1): T_A = -40 °C to +110 °C
µPD70F3187(A2): T_A = -40 °C to +125 °C

Table 25-5: DC Characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	PAL0 to PAL15, PAH0 to PAH5, PDL0 to PDL15, PDH0 to PDH15, PCS0, PCS1, PCS3, PCS4, PCD2 to PCD5, PCT4, PCT5, PCM0, PCM1, PCM6, PCM7, DCK, DMS, DDI, DDO	0.7 V_{DD3}		V_{DD3} + 0.3	V
	V_{IH3}	P00 to P04, P10 to P17, P20 to P27, P30 to P37, P40 to P45, P50 to P57, P60 to P67, P70 to P75, P80 to P86, P90 to P96, P100 to P102, $\overline{\text{RESET}}$, MODE0 to MODE2, $\overline{\text{DRST}}$				
Input voltage, low	V_{IL1}	PAL0 to PAL15, PAH0 to PAH5, PDL0 to PDL15, PDH0 to PDH15, PCS0, PCS1, PCS3, PCS4, PCD2 to PCD5, PCT4, PCT5, PCM0, PCM1, PCM6, PCM7, DCK, DMS, DDI, DDO	-0.5		0.3 V_{DD3}	V
	V_{IL3}	P00 to P04, P10 to P17, P20 to P27, P30 to P37, P40 to P45, P50 to P57, P60 to P67, P70 to P75, P80 to P86, P90 to P96, P100 to P102, $\overline{\text{RESET}}$, MODE0 to MODE2, $\overline{\text{DRST}}$				
Output voltage, high	V_{OH1}	I_{OH} = -2.5 mA	V_{DD3} - 1.0			V
		I_{OH} = -100 µA	V_{DD3} - 0.4			V
Output voltage, low	V_{OL1}	I_{OL} = 2.5 mA			0.8	V
		I_{OL} = 100 µA			0.4	V
Input leakage current, high	I_{LIH}	$V_{IH} = V_{DD3}$, **Note 1**			10	µA
		$V_{IH} = AV_{DD}$, **Note 2**			3	µA
Input leakage current, low	I_{LIL}	V_{IL} = 0 V, **Note 1**			-10	µA
		V_{IL} = 0 V, **Note 2**			-3	µA
Power supply current	I_{DD1}	$V_{DD1} + CV_{DD}$			200	mA
	I_{DD3}	V_{DD3}, **Note 3**			50	mA

Notes: 1. Pins other than analog input pins ANI00 to ANI19

2. Analog input pins ANI00 to ANI19

3. No external loads considered (C_L = 0pF). External loads cause additional pin currents. Pin current for each pin can be calculated according to following formula:
$I[\mu A] = 3.63 \times C_L [pF] \times F [MHz]$
where C_L is external load capacitance and F is the average pin toggle frequency.
Load dependent pin currents must be summed up and added to I_{DD3}.

25.4 AC Characteristics

Unless specified otherwise, the following conditions are assumed for all characteristics in this chapter.

$V_{DD3x} = AV_{DD}$ = 3.0 V to 3.6 V
$V_{DD1x} = CV_{DD}$ = 1.35 V to 1.65 V
$V_{SS1x} = CV_{SS} = V_{SS3x} = AV_{SSx}$ = 0 V
µPD70F3187: T_A = -40 °C to +85 °C
µPD70F3187(A1): T_A = -40 °C to +110 °C
µPD70F3187(A2): T_A = -40 °C to +125 °C
Output pin load capacitance: C_L = 35 pF

Figure 25-2: AC Test Input/Output Waveform

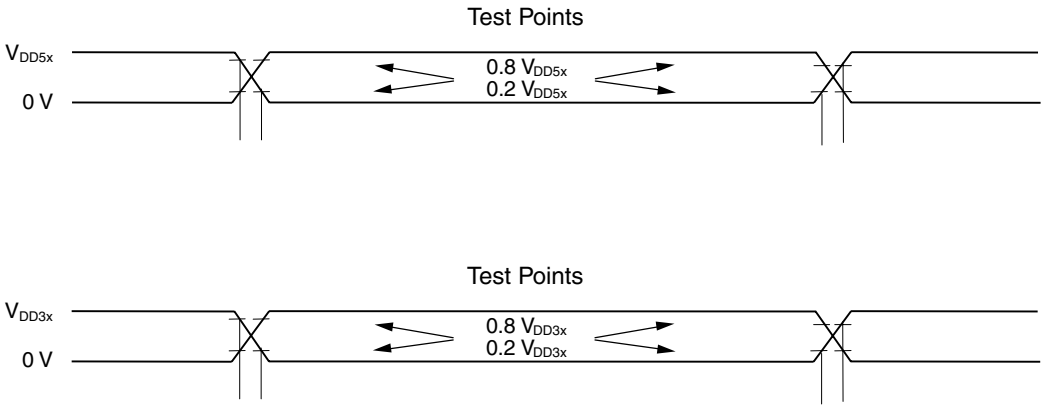
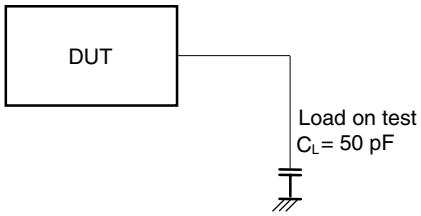


Figure 25-3: AC Test Load Condition



25.4.1 External asynchronous memory access read timing

Table 25-6: External Asynchronous Memory Access Read Timing

Parameter		Symbol	MIN.	MAX.	Unit
Data input set up time (vs. address)	<10>	t_{SAID}		$(2 + w_{AS} + w_D + w)$ T - 30	ns
Data input set up time (vs. $\overline{RD}\downarrow$)	<11>	t_{SRDID}		$(1.5 + w_D + w)$ T - 30	ns
$\overline{RD}$ Low level width	<12>	t_{WRDL}	$(1.5 + w_D + w)$ T - 15		ns
$\overline{RD}$ High level width	<13>	t_{WRDH}	$(0.5 + w_{AS} + i)$ T - 15		ns
Address, $\overline{CSn} \rightarrow \overline{RD}\downarrow$ delay time	<14>	t_{DARD}	$(0.5 + w_{AS})$ T - 20		ns
$\overline{RD}\uparrow \rightarrow$ address delay time	<15>	t_{DRDA}	iT - 2		ns
Data input hold time (vs. $\overline{RD}\uparrow$)	<16>	t_{HRDID}	0		ns
$\overline{RD}\uparrow \rightarrow$ data output delay time	<17>	t_{DRDOD}	(1 + i) T - 15		ns
$\overline{WAIT}$ set up time (vs. address)	< 31 >	t_{SAW}		$(1 + w_{AS})$ T- 30	ns
$\overline{WAIT}$ high level width	<32>	t_{WWH}	T - 2		ns

Remarks: **1.** T: $2/f_{XX}$

2. i: Number of idle states specified by BCC register

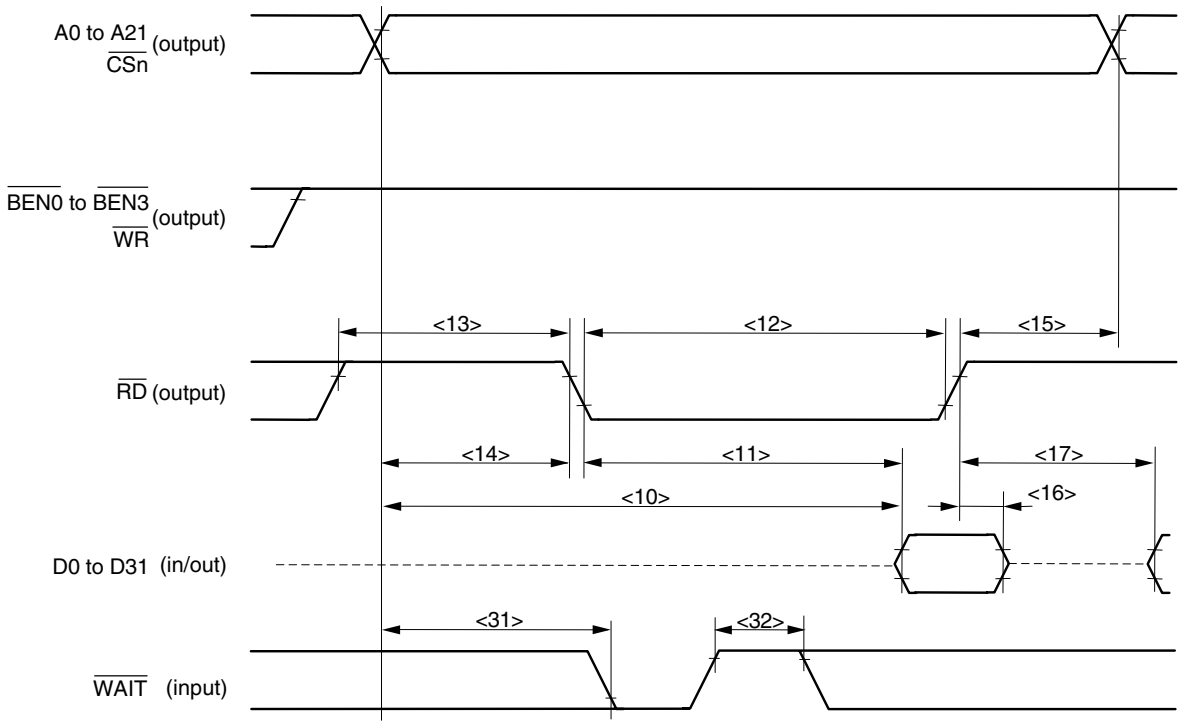
3. w_{AS}: Number of waits specified by ASC register

4. w_D: Number of waits specified by DWC1, DWC2 register; $w_D \geq 1$

5. w: Number of waits due to external wait signal ($\overline{WAIT}$)

6. n = 0, 1, 3, 4

Figure 25-4: External Asynchronous Memory Access Read Timing



25.4.2 External asynchronous memory access write timing

Table 25-7: External Asynchronous Memory Access Write Timing

Parameter		Symbol	MIN.	MAX.	Unit
Address, $\overline{CSn} \rightarrow \overline{WR} \downarrow$ delay time	<20>	T_{DAWR}	$(1 + w_{AS})T - 20$		ns
Address set up (vs. $\overline{WR}\uparrow$)	<21>	T_{SAWR}	$(1.5 + w_{AS} + w_D + w)\ T - 10$		ns
$\overline{WR}\uparrow \rightarrow$address delay time	<22>	T_{DWRA}	$(0.5 + i)\ T - 5$		ns
$\overline{WR}$ High level width	<23>	T_{WWRH}	$(1.5 + i + w_{AS})\ T - 15$		ns
$\overline{WR}$ Low level width	<24>	T_{WWRL}	$(0.5 + w + w_D)\ T - 12$		ns
Data output set up time (vs. $\overline{WR}\uparrow$)	<25>	T_{SODWR}	$(0.5 + w_{AS} + w_D + w)\ T - 15$		ns
Data output hold time (vs. $\overline{WR}\uparrow$)	<26>	T_{HWROD}	$(0.5 + i)\ T - 15$		ns
$\overline{WAIT}$ set up time (vs. address)	<31>	T_{SAW}		$(1 + w_{AS})T - 30$	ns
$\overline{WAIT}$ high level width	<32>	T_{WWH}	T - 2		ns

Remarks: **1.** T: $2/f_{XX}$

2. i: Number of idle states specified by BCC register

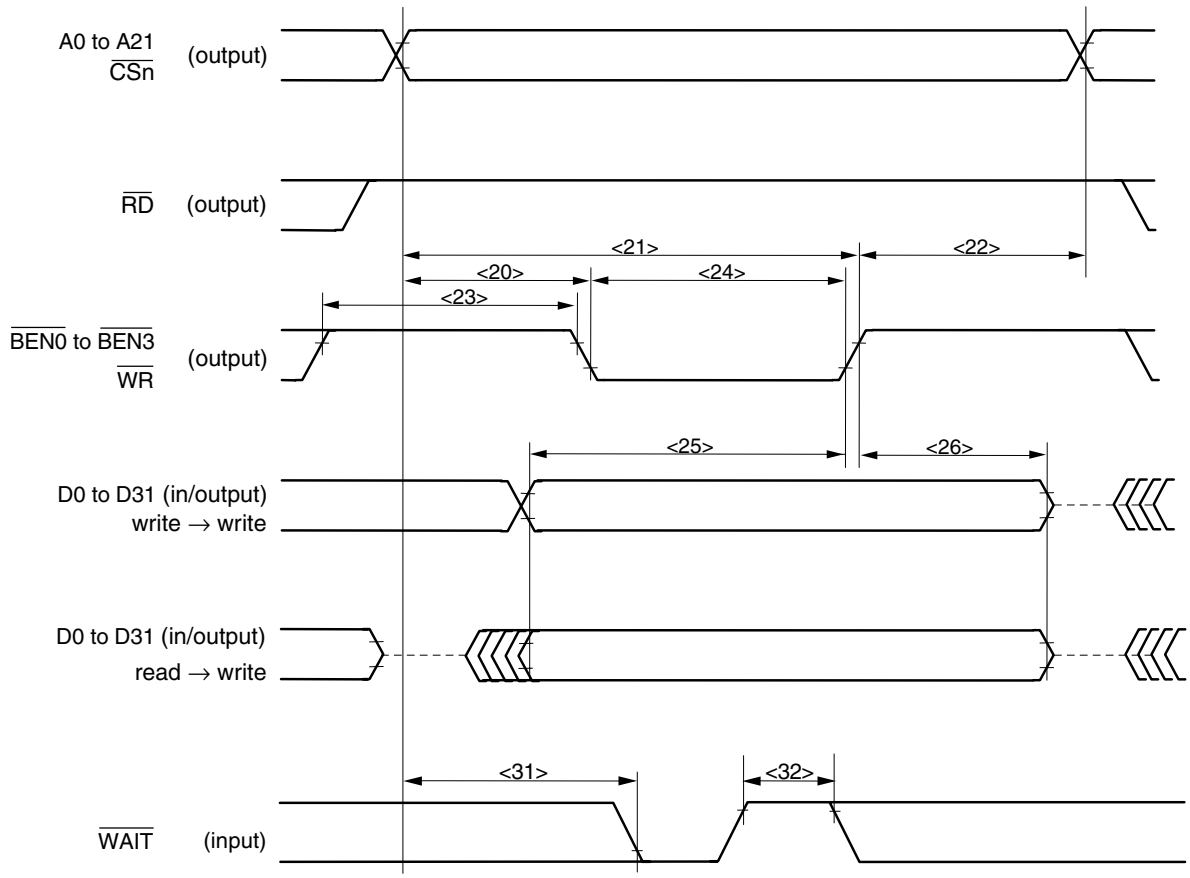
3. w_{AS}: Number of waits specified by ASC register

4. w_D: Number of waits specified by DWC1, DWC2 register; $w_D \geq 1$

5. w: Number of waits due to external wait signal ($\overline{WAIT}$)

6. n = 0, 1, 3, 4

Figure 25-5: External Asynchronous Memory Access Write Timing

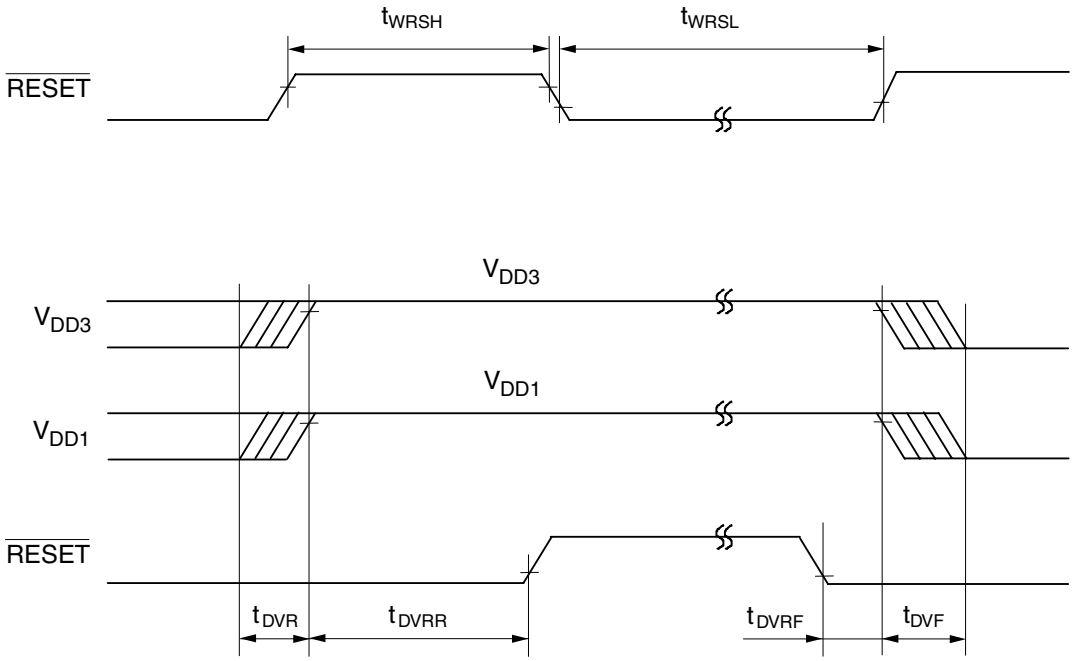


25.4.3 Reset Timing (Power Up/Down Sequence)

Table 25-8: Reset Timing

Parameter	Symbol	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ high-level width	t_{WRSH}	500		ns
$\overline{\text{RESET}}$ low-level width	t_{WRSL}	500		ns
$V_{DD3x} \leftrightarrow V_{DD1x}$ power up delay	t_{DVR}	0		ns
$V_{DD3x} \leftrightarrow V_{DD1x}$ power down delay	t_{DVF}	0		ns
$\overline{\text{RESET}}$ hold time	t_{DVRR}	1		µs
$\overline{\text{RESET}}$ setup time	t_{DVRF}	0		ns

Figure 25-6: Reset Timing



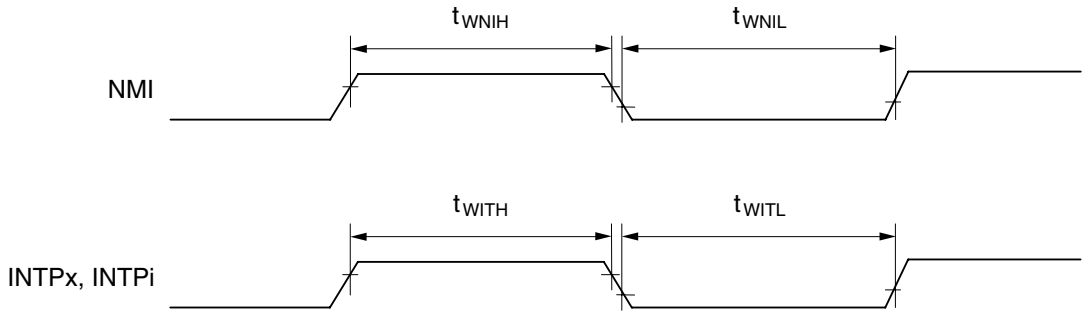
Caution: Ensure that a valid $\overline{\text{RESET}}$ signal (low active) is applied to the $\overline{\text{RESET}}$ pin at any time if the voltage power of V_{DD1x} is below its operating condition range.

25.4.4 Interrupt timing

Table 25-9: Interrupt Timing

Parameter	Symbol	Condition	MIN.	MAX.	Unit
NMI high-level width	t_{WNIH}	NRC0 bit = 0	96 T + 10		ns
		NRC0 bit = 1	384 T + 10		ns
NMI low-level width	t_{WNIL}	NRC0 bit = 0	96 T + 10		ns
		NRC0 bit = 1	384 T + 10		ns
INTPx high-level width	t_{WITH}	NRC1 bit = 0	96 T + 10		ns
		NRC1 bit = 1	384 T + 10		ns
INTPx low-level width	t_{WITL}	NRC1 bit = 0	96 T + 10		ns
		NRC1 bit = 1	384 T + 10		ns
INTP0, INTP1 high-level width	t_{WTIH}		500		ns
INTP0, INTP1 low-level width	t_{WTIL}		500		ns

Figure 25-7: Interrupt Timing



25.5 Peripheral Characteristics

Unless specified otherwise, the following conditions are assumed for all characteristics in this chapter.

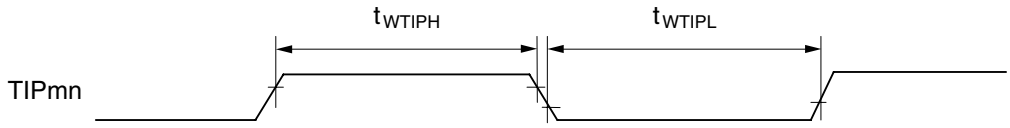
$V_{DD3x} = AV_{DD}$ = 3.0 V to 3.6 V
$V_{DD1x} = CV_{DD}$ = 1.35 V to 1.65 V
$V_{SS1x} = CV_{SS} = V_{SS3x} = AV_{SSx}$ = 0 V
µPD70F3187: T_A = -40 °C to +85 °C
µPD70F3187(A1): T_A = -40 °C to +110 °C
µPD70F3187(A2): T_A = -40 °C to +125 °C

25.5.1 Timer characteristics

Table 25-10: Timer P Characteristics

Parameter	Symbol	Condition	MIN.	MAX.	Unit
TIPmn input high-level width	t_{WTIPH}	NRCx bit = 0	96 T + 10		ns
		NRCx bit = 1	384 T + 10		ns
TIPmn input low-level width	t_{WTIPL}	NRCx bit = 0	96 T + 10		ns
		NRCx bit = 1	384 T + 10		ns

Figure 25-8: Timer P Characteristics

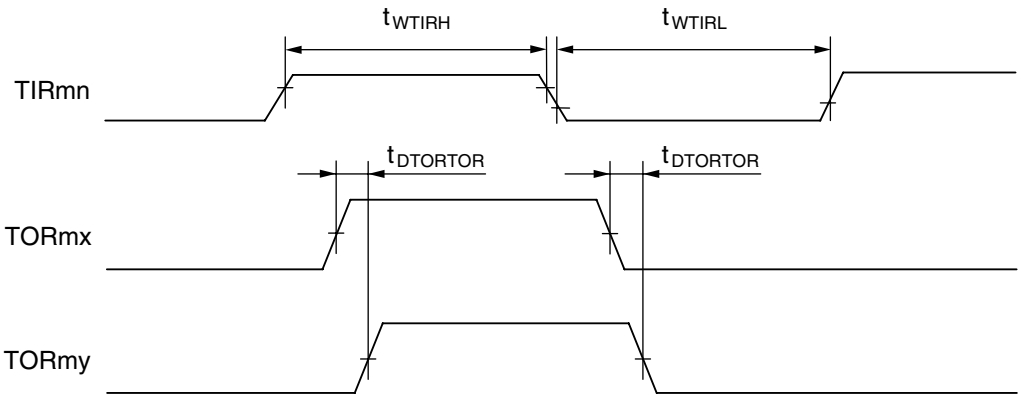


Remark: m = 0 to 7
n = 0 to 1
x = 3 to 6 (depending on the pin group the TIPmn belongs to, refer to 20.4 ”Noise Elimination” on page 959).

Table 25-11: Timer R Characteristics

Parameter	Symbol	Condition	MIN.	MAX.	Unit
TIR1n input high-level width	t_{WTIRH}	NRC7 bit = 0	96 T + 10		ns
		NRC7 bit = 1	384 T + 10		ns
TIR1n input low-level width	t_{WTIRL}	NRC7 bit = 0	96 T + 10		ns
		NRC7 bit = 1	384 T + 10		ns
TORmx to TORmy output delay	$t_{DTORTOR}$			15	ns

Figure 25-9: Timer R Characteristics

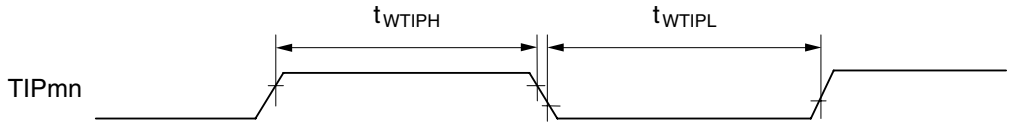


Remark: m = 0, 1
n = 0 to 3
x = 0 to 7, y = 0 to 7, x ≠ y

Table 25-12: Timer T Characteristics

Parameter	Symbol	Condition	MIN.	MAX.	Unit
TITmn input high-level width	t_{WTITH}	NRC2 bit = 0	96 T + 10		ns
		NRC2 bit = 1	384 T + 10		ns
TITmn input low-level width	t_{WTITL}	NRC2 bit = 0	96 T + 10		ns
		NRC2 bit = 1	384 T + 10		ns

Figure 25-10: Timer T Characteristics



Remark: m = 0, 1
n = 0, 1

25.5.2 Serial interface characteristics

(1) Clocked serial interface B (CSIB) characteristics

Table 25-13: CSIB Characteristics (Master Mode)

CBnSCK2 to CBnSCK0 ≠ 111B

Parameter	Symbol	MIN.	MAX.	Unit
$\overline{SCKBn}$ output clock cycle time	t_{CYSKM}	125		ns
$\overline{SCKBn}$ output high level width	t_{WSKHM}	0.5 t_{CYSKM} - 10		ns
$\overline{SCKBn}$ output low level width	t_{WSKLM}	0.5 t_{CYSKM} - 10		ns
SIBn input setup time (vs. $\overline{SCKBn}\uparrow$)	t_{SSISKM}	20		ns
SIBn input hold time (vs. $\overline{SCKBn}\uparrow$)	t_{HSKSIM}	10		ns
SOBn output delay (vs. $\overline{SCKBn}\downarrow$)	t_{DSKSOM}		10	ns
SOBn output hold time (vs. $\overline{SCKBn}\uparrow$)	t_{HSKSOM}	0.5 t_{CYSKM} - 10		ns

Table 25-14: CSIB Characteristics (Slave Mode)

CBnSCK2 to CBnSCK0 = 111B

Parameter	Symbol	MIN.	MAX.	Unit
$\overline{SCKBn}$ input clock cycle time	t_{CYSKS}	125		ns
$\overline{SCKBn}$ input high level width	t_{WSKHS}	0.5 t_{CYSKS} - 10		ns
$\overline{SCKBn}$ input low level width	t_{WSKLS}	0.5 t_{CYSKS} - 10		ns
SIBn input setup time (vs. $\overline{SCKBn}\uparrow$)	t_{SSISKS}	5		ns
SIBn input hold time (vs. $\overline{SCKBn}\uparrow$)	t_{HSKSIS}	10		ns
SOBn output delay (vs. $\overline{SCKBn}\downarrow$)	t_{DSKSOS}		25	ns
SOBn output hold time (vs. $\overline{SCKBn}\uparrow$)	t_{HSKSOS}	0.5 t_{CYSKS} - 10		ns

Remark: n = 0, 1

Figure 25-11: CSIB Timing in Master Mode (CKP, DAP bits = 00B or 11B)

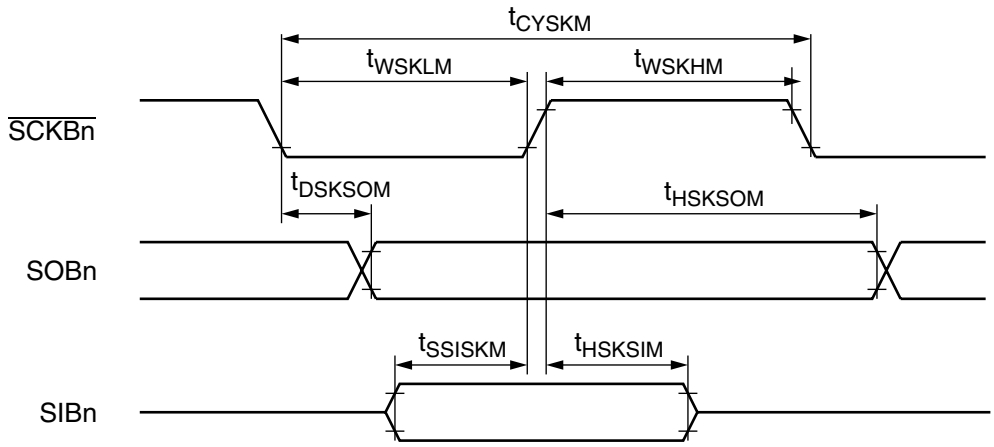


Figure 25-12: CSIB Timing in Master Mode (CKP, DAP bits = 01B or 10B)

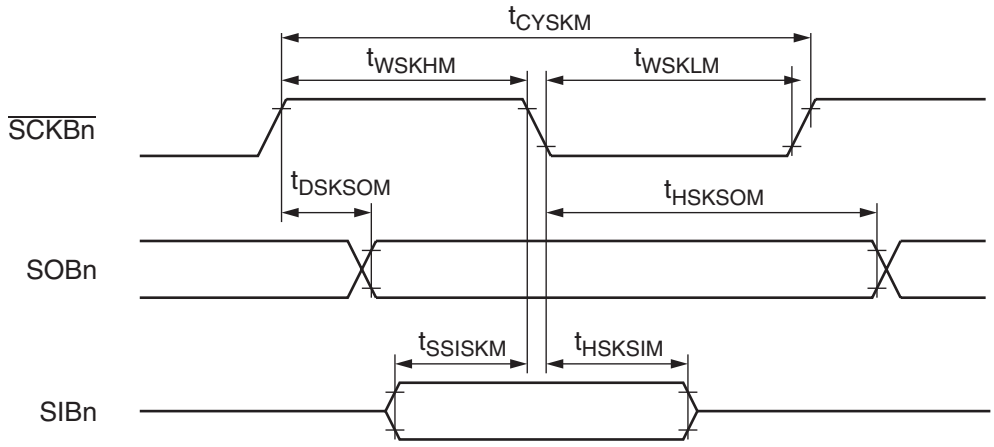


Figure 25-13: CSIB Timing in Slave Mode (CKP, DAP bits = 00B or 11B)

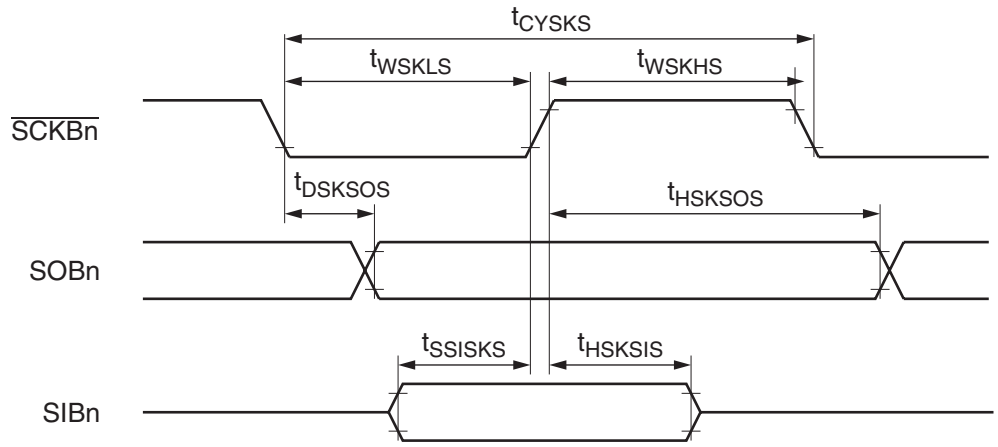
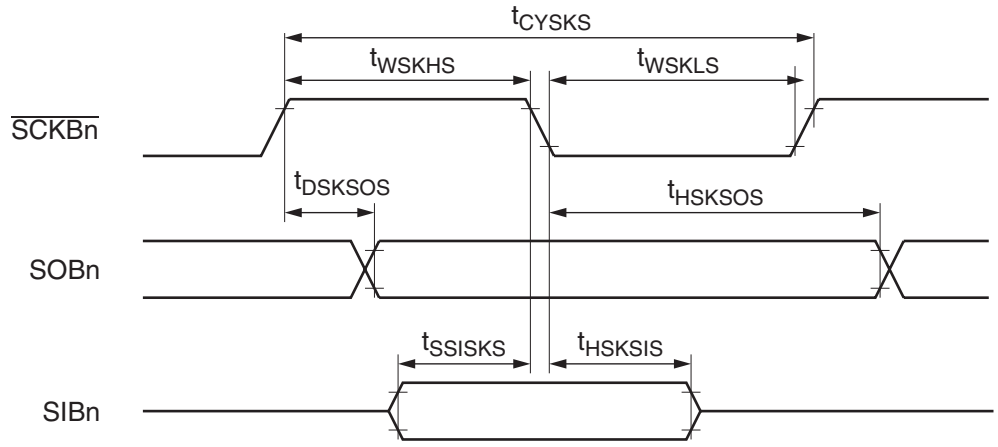


Figure 25-14: CSIB Timing in Slave Mode (CKP, DAP bits = 01B or 10B)



(2) Clocked serial interface 3 (CSI3) timing

Table 25-15: CSI3 Characteristics (Master Mode)

CKS3n2 to CKS3n0 ≠ 111B

Parameter	Symbol	MIN.	MAX.	Unit
CSI3 operation clock cycle time	t_{CYK}	15.625		ns
$\overline{SCK3n}$ clock cycle time	t_{CYSKM}	125		ns
$\overline{SCK3n}$ high level width	t_{WSKHM}	0.5 t_{CYSKM} - 10		ns
$\overline{SCK3n}$ low level width	t_{WSKLM}	0.5 t_{CYSKM} - 10		ns
SI3n setup time (vs. $\overline{SCK3n}\uparrow$)	t_{SSISKM}	20		ns
SI3n hold time (vs. $\overline{SCK3n}\uparrow$)	t_{HSKSIM}	10		ns
SO3n output delay (vs. $\overline{SCK3n}\downarrow$)	t_{DSKSOM}		10	ns
SO3n output hold time (vs. $\overline{SCK3n}\uparrow$)	t_{HSKSOM}	0.5 t_{CYSKM} - 10		ns
SCS3nm inactive width	t_{WSKCSB}	0.5 t_{CYSKM} - 10		ns
SCS3nm setup time (vs. $\overline{SCK3n}\downarrow$)	$t_{SCSZCK0}$	t_{CYK} - 10		ns
	$t_{SCSZCK1}$	$t_{CYSKM} + t_{CYK - 10}$		ns
	$t_{SCSZCK2}$	$t_{CYSKM} - t_{CYK - 10}$		ns
SCS3nm hold time (vs. $\overline{SCK3n}\uparrow$)	$t_{HSKCSZ0}$	t_{CYK} - 10		ns
	$t_{HSKCSZ1}$	0.5 t_{CYSKM} - 10		ns

Table 25-16: CSI3 Characteristics (Slave Mode)

CKS3n2 to CKS3n0 = 111B

Parameter	Symbol	MIN.	MAX.	Unit
CSI3 operation clock cycle time	t_{CYK}	15.625		ns
$\overline{SCK3n}$ clock cycle time	t_{CYSKS}	125		ns
$\overline{SCK3n}$ high level width	t_{WSKHS}	0.5 t_{CYSKS} - 10		ns
$\overline{SCK3n}$ low level width	t_{WSKLS}	0.5 t_{CYSKS} - 10		ns
SI3n setup time (vs. $\overline{SCK3n}\uparrow$)	t_{SSISKS}	5		ns
SI3n hold time (vs. $\overline{SCK3n}\uparrow$)	t_{HSKSIS}	1.5 t_{CYK} + 10		ns
SO3n output delay (vs. $\overline{SCK3n}\downarrow$)	t_{DSKSOS}		25	ns
SO3n output hold time (vs. $\overline{SCK3n}\uparrow$)	t_{HSKSOS}	0.5 t_{CYSKS} - 10		ns

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

Figure 25-15: CSI3 Timing in Master Mode (CKP, DAP bits = 00B or 11B)

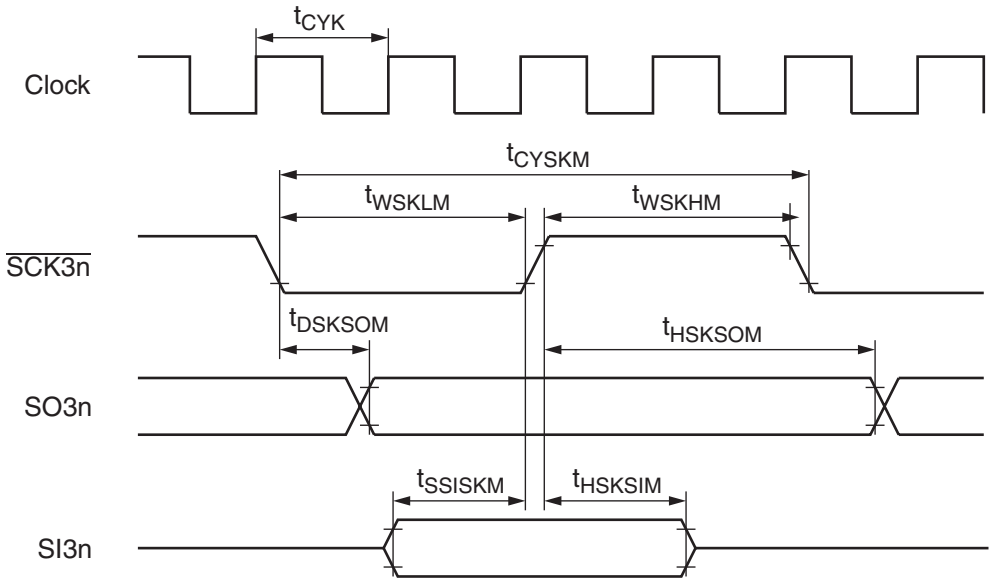


Figure 25-16: CSI3 Timing in Master Mode (CKP, DAP bits = 01B or 10B)

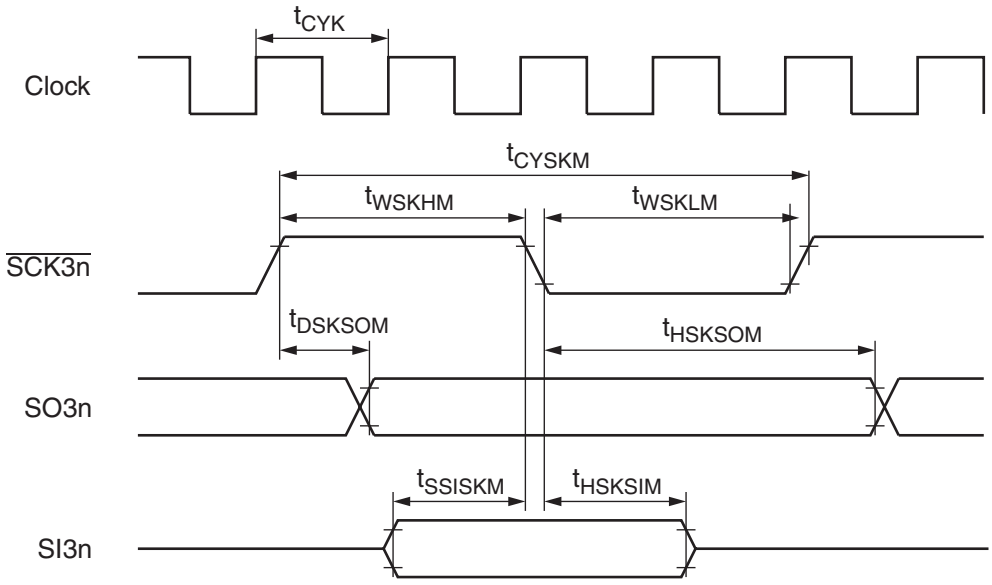


Figure 25-17: CSI3 Timing in Slave Mode (CKP, DAP bits = 00B or 11B)

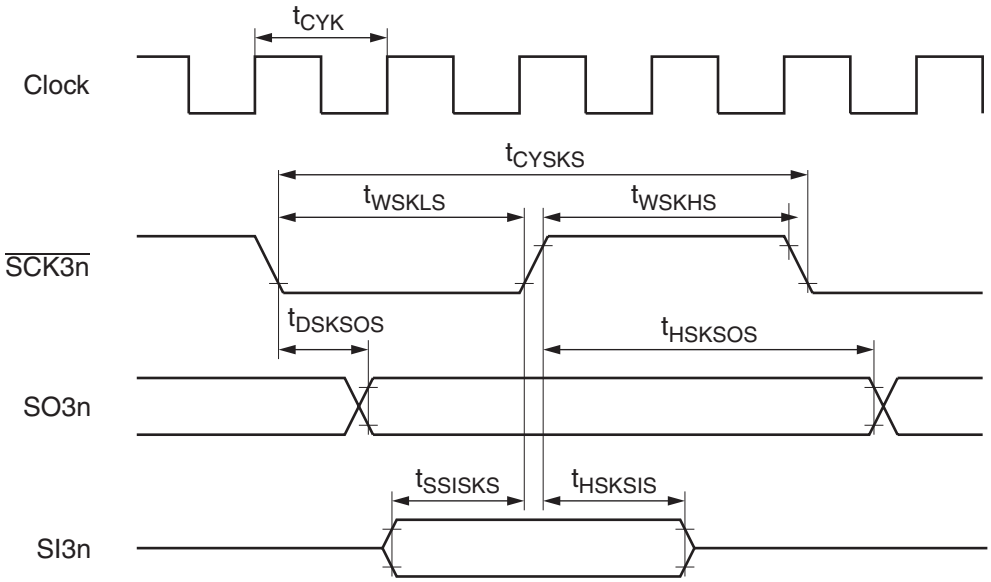


Figure 25-18: CSI3 Timing in Slave Mode (CKP, DAP bits = 01B or 10B)

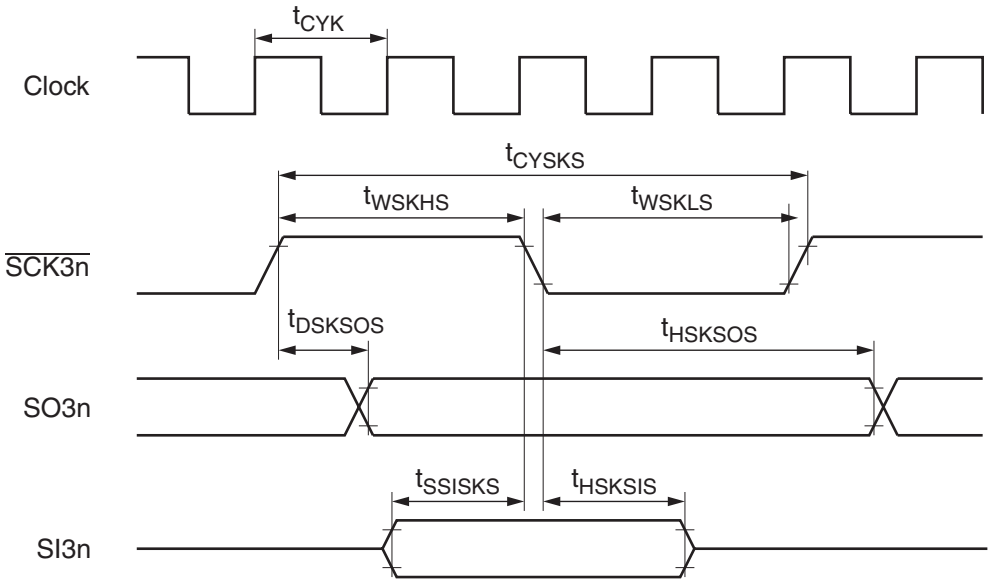


Figure 25-19: CSI3 Chip Select Timing (Master Mode only) (CSIT = 0, CSWE = 0, CSMD = 0)

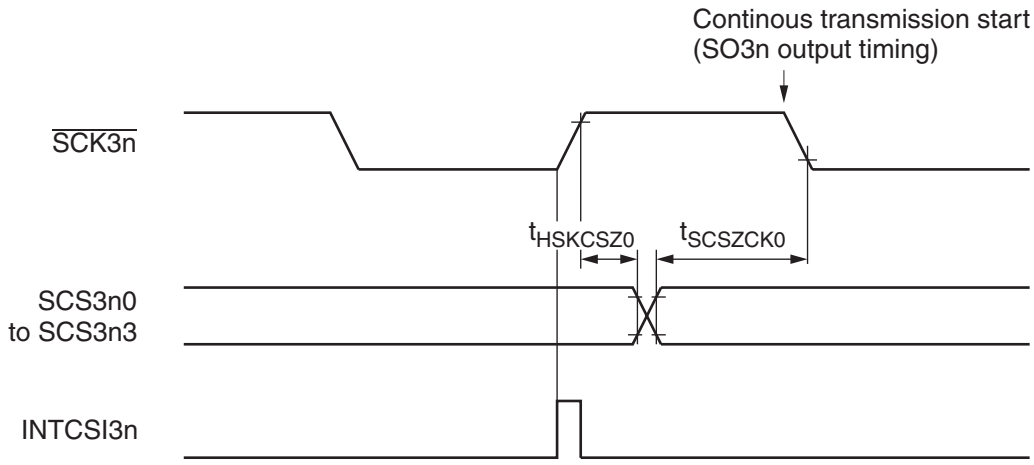


Figure 25-20: CSI3 Chip Select Timing (Master Mode only) (CSIT = 0, CSWE = 1, CSMD = 0)

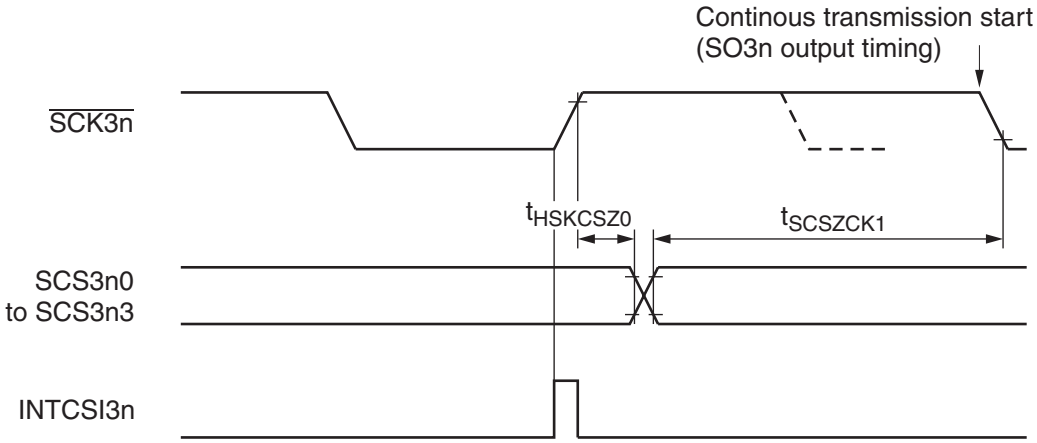


Figure 25-21: CSI3 Chip Select Timing (Master Mode only) (CSIT = 0, CSWE = 1, CSMD = 1)

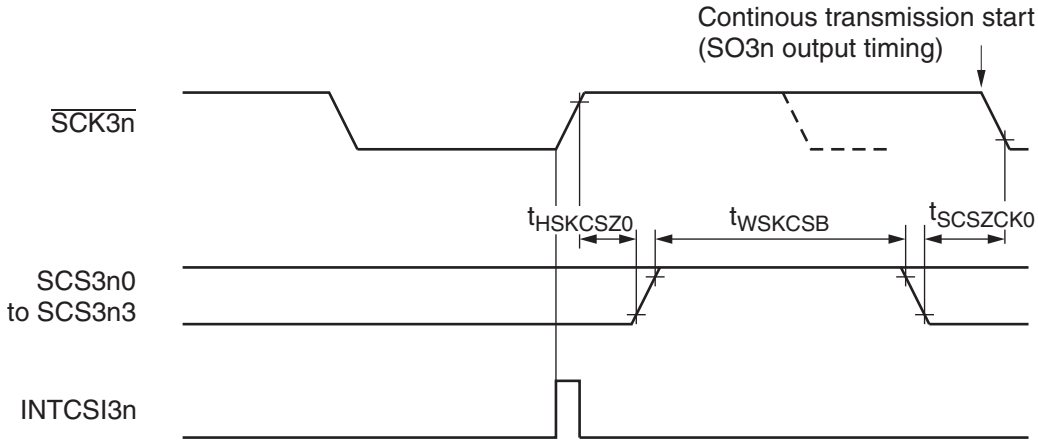


Figure 25-22: CSI3 Chip Select Timing (Master Mode only) (CSIT = 1, CSWE = 0, CSMD = 0)

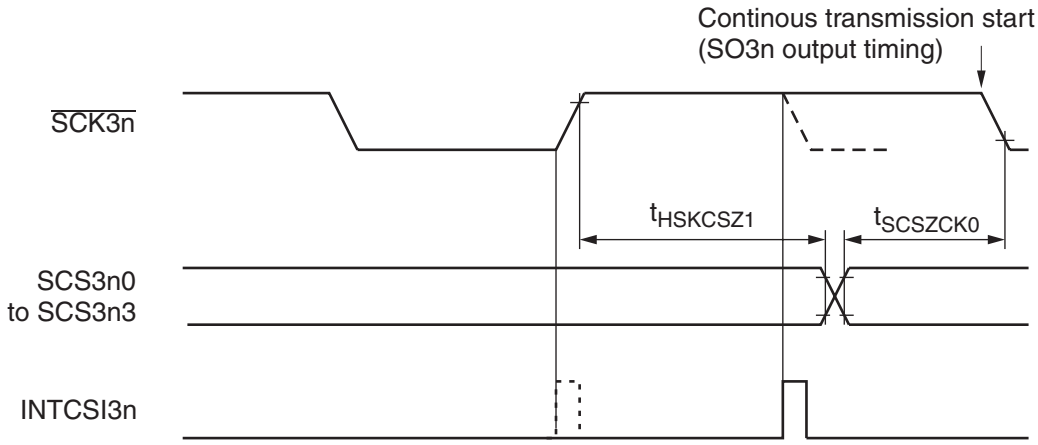


Figure 25-23: CSI3 Chip Select Timing (Master Mode only) (CSIT = 1, CSWE = 1, CSMD = 0)

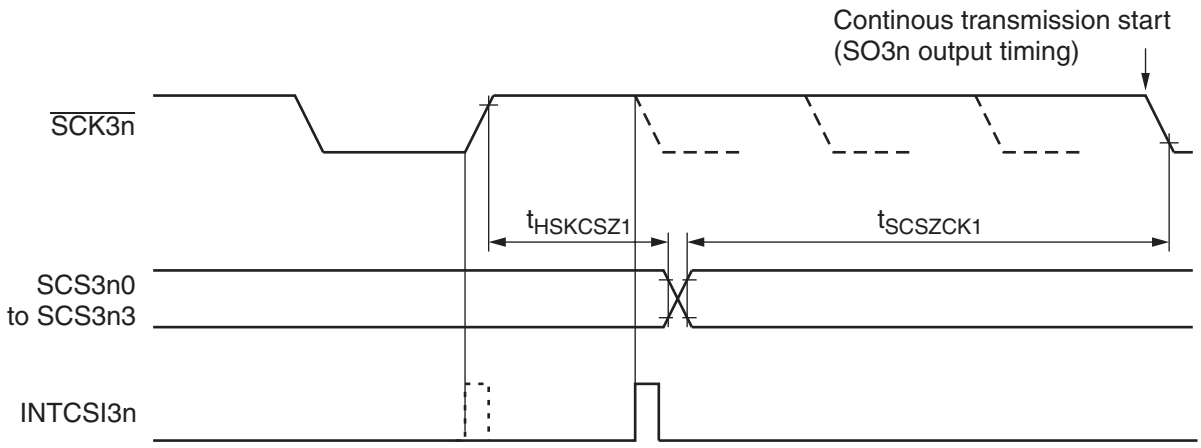
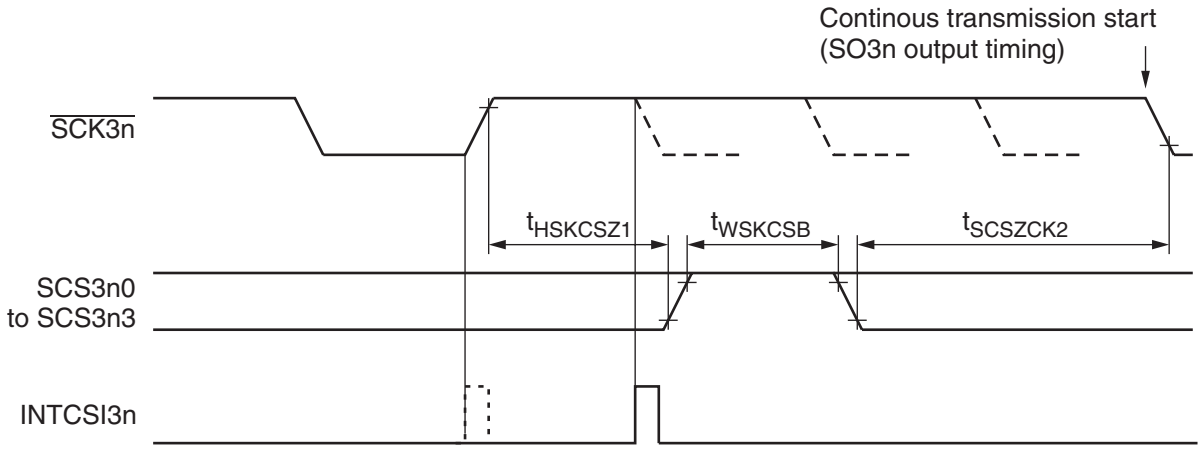


Figure 25-24: CSI3 Chip Select Timing (Master Mode only) (CSIT = 1, CSWE = 1, CSMD = 1)



25.5.3 A/D Converter Characteristics

Table 25-17: A/D Converter Characteristics

Parameter	Symbol	MIN.	TYP.	MAX.	Unit
Resolution	-		10		Bit
Overall error	-			±4	LSB
Conversion time	T_{CONV}	2		8	µs
Sampling time	T_{SAM}	0.375		1.5	µs
Analog input voltage	V_{IAN}	AV_{SS}		AV_{DD}	V
Analog supply current	I_{AVDD}			2.4	mA
Reference voltage	AV_{REF}	AV_{DD}		AV_{DD}	V

25.6 Flash Programming Characteristics

Unless specified otherwise, the following conditions are assumed for all characteristics in this chapter.

$V_{DD3x} = AV_{DD} = 3.0$ V to 3.6 V
$V_{DD1x} = CV_{DD} = 1.35$ V to 1.65 V
$V_{SS1x} = CV_{SS} = V_{SS3x} = AV_{SSx} = 0$ V

Table 25-18: Flash Memory Basic Characteristics

Parameter	Condition	Symbol	MIN.	TYP.	MAX.	Unit
Number of rewrites		C_{WRT}			100	times/ block
Ambient programming temperature		T_{APRG}	-40		+85	°C
Data retention time	6000 h key-on time		15			years

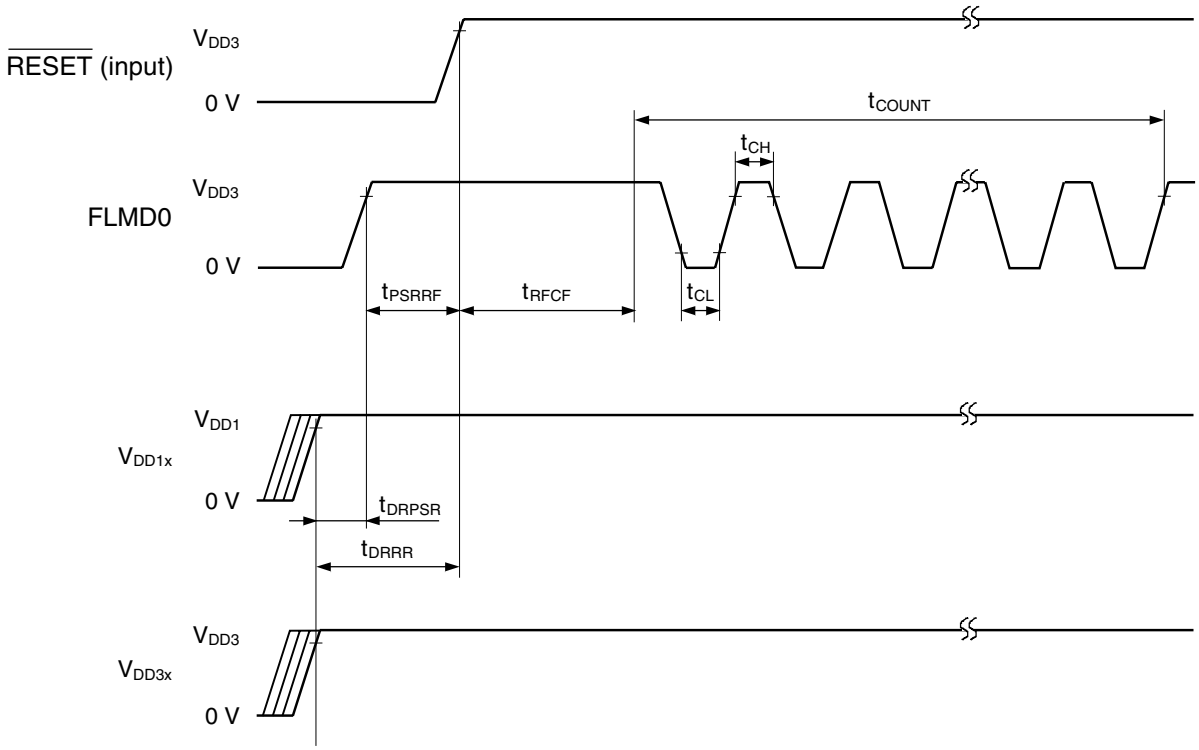
Table 25-19: Flash Memory Programming Characteristics

Parameter	Symbol	MIN.	TYP.	MAX.	Unit
Write time			30	300	µs/ word
Erase time			0.2	2	s

Table 25-20: Serial Write Operation Characteristics

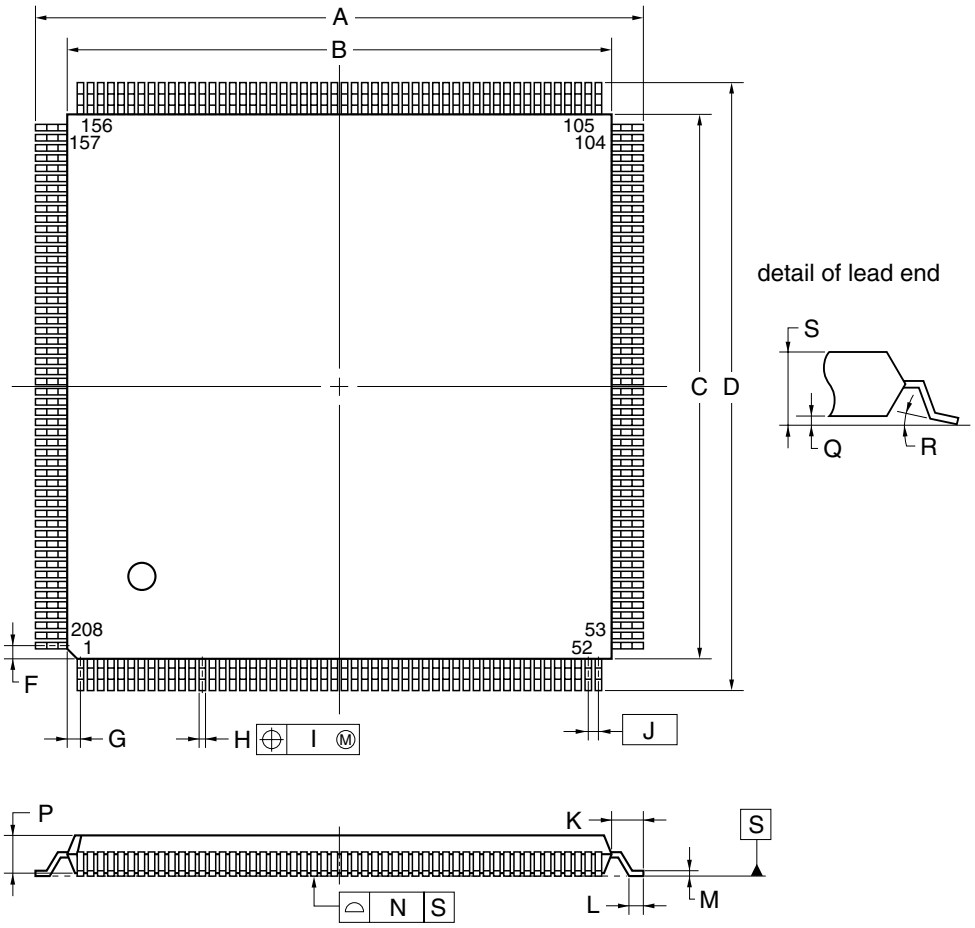
Parameter	Symbol	MIN.	TYP.	MAX.	Unit
$V_{DD}\uparrow$ setup time to FLMD0$\uparrow$	t_{DRPSR}	0			ns
$V_{DD}\uparrow$ setup time to $\overline{\text{RESET}}\uparrow$	t_{DRRR}	2			ms
FLMD0 setup time to $\overline{\text{RESET}}\uparrow$	t_{PSRRF}	2			ms
FLMD0 count start time from $\overline{\text{RESET}}\uparrow$	t_{RFCF}	10			ms
FLMD0 count time	t_{COUNT}			10	ms
FLMD0 counter high-level width/low-level width	t_{CH}/t_{CL}	10			µs

Figure 25-25: Serial Write Operation Characteristics



Chapter 26 Package Drawings

Figure 26-1: 208-Pin Plastic QFP (Fine Pitch) (28 x 28)



NOTE

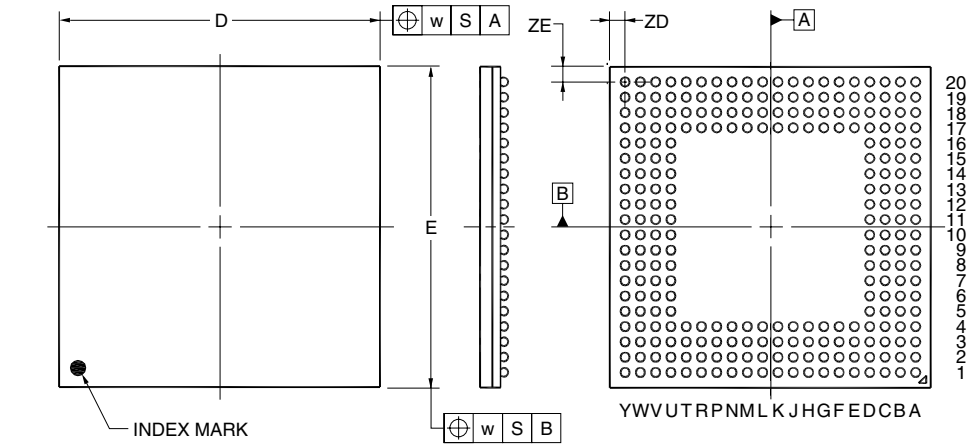
Each lead centerline is located within 0.10 mm of its true position (T.P.) at maximum material condition.

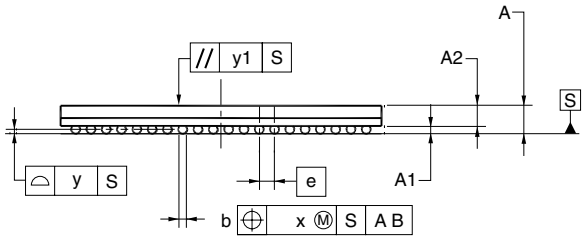
ITEM	MILLIMETERS
A	30.6±0.2
B	28.0±0.2
C	28.0±0.2
D	30.6±0.2
F	1.25
G	1.25
H	$0.22^{+0.05}_{-0.04}$
I	0.10
J	0.5 (T.P.)
K	1.3±0.2
L	0.5±0.2
M	$0.17^{+0.03}_{-0.07}$
N	0.10
P	3.2±0.1
Q	0.4±0.1
R	5°±5°
S	3.8 MAX.

P208GD-50-LML, MML, SML, WML-7

Figure 26-2: 256-Pin Plastic BGA (Fine Pitch) (21 x 21)

256-PIN PLASTIC BGA (21x21)





(UNIT:mm)

ITEM	DIMENSIONS
D	21.00±0.10
E	21.00±0.10
w	0.30
e	1.00
A	1.83±0.17
A1	0.50±0.10
A2	1.33
b	0.60±0.10
x	0.15
y	0.15
y1	0.35
ZD	1.00
ZE	1.00

P256F1-100-JN4

Chapter 27 Recommended Soldering Conditions

Solder this product under the following recommended conditions.
For details of the recommended soldering conditions, refer to information document Semiconductor Device:
Mounting Technology Manual (C10535E).

For soldering methods and conditions other than those recommended please consult NEC.

Table 27-1: Soldering Conditions

Soldering Method	Soldering Condition	Symbol of Recommended Soldering Condition
Infrared reflow	Package peak temperature: tbd °C, Time: tbd seconds max. (tbd °C min.), Number of times: tbd max., Number of days: tbd **Note**	tbd
VPS	Package peak temperature: tbd °C, Time: tbd seconds max. (tbd°C min.), Number of times: tbd max., Number of days: tbd**Note**	tbd
Partial heating	Pin temperature: tbd °C max., Time: tbd seconds max. (per side of device)	-

Note: After that, prebaking is necessary at tbd °C for tbd hours.
The number of days refers to storage at 25°C, 65% RH MAX after the dry pack has been opened.

Caution: Do not use two or more soldering methods in combination (except partial heating method).

[MEMO]

Appendix A Index

U

[MEMO]

Appendix B Revision History

The following shows the revision history up to present. Application portions signifies the chapter of each edition.

(1/4)

Edition No.	Major items revised	Revised Sections
EE2	Due to the new guideline of wording and terminology, the nickname “PHOENIX-F” removed throughout the UM	whole User’s Manual
	“Burst flash memory” removed throughout the UM	whole User’s Manual
	BCLK (Burst clock output), STST and STNXT ((Burst control output) removed	whole User’s Manual
	1.4 “Ordering Information”, Note deleted	1.4, p.35
	Figure 1-1 “Pin Configuration 208-pin Plastic LQFP” changed	1.5, p. 36
	Table 1-1 “256-pin Plastic BGA” changed	1.5, p.38
	Section “Pin identification” changed	1.5, p.40
	Figure 1-3 “Internal Block Diagram” changed	1.6.1, p.41
	Table 2-1 “Port Pins” changed	2.1, p.48
	Table 2-2 “Non-Port Pins” changed	2.1, p.50, p.52
	Table 3-5 “Peripheral I/O Registers” changed	3.4.6, p.101, 104, 107
	Figure 3-20 “Programmable Peripheral Area Control Register BPC” bits 13 and 12 changed	3.4.7 (1), p.112
	3.4.8 (1), Setting data to specific registers changed, Caution 2 added	3.4.8 (1), p.127
	3.4.10, DMA wait control register 0 (DMAWC0) set value changed	3.4.10, p.130
	Figure 4-11 “Bus Clock Dividing Control Register (DVC)” address value changed	4.7 (2), p.166
	Table 3-5 “Peripheral I/O Registers” changed	3.4.6, p.101, 104, 107
	Chapter 5.2 “Burst Mode Flash” removed	5.2, p.180
	6.1 “Features of the DMA” added	6.1, p.181
	Figure 6-1 “DMA Transfer Memory Start Address Registers 0 to 7 (MAR0 to MAR7)”, Caution 2 added	6.2 (1), p.182
	Figure 6-2 “DMA Transfer SFR Start Address Registers 2, 3 (SAR2, SAR3)”, Caution added	6.2 (2), p.183
	Figure 6-3 “DMA Transfer Count Registers 0 to 7 (DTCR0 to DTCR7)”, Caution 3 deleted	6.2 (3), p.184
	Figure 6-8 “Initialization of DMA Transfer for A/D Conversion Result” changed	6.4.1, p.189
	Figure 6-9 “Operation of DMA Channel 0/1” added	6.4.1, p.190
	Figure 6-10 “DMA Channel 0 and 1 Trigger Signal Timing”, added	6.4.1, p.191
	6.4.2 “DMA transfer of PWM timer reload (TMR0, TMR1)”, DMA trigger count changed	6.4.2, p.192
	Figure 6-11 “Initialization of DMA Transfer for TMRn Compare Registers”, changed	6.4.2, p.193
	Figure 6-12 “Operation of DMA Channel 2/3”, added	6.4.2, p.194
	Figure 6-13 “DMA Channel 2 and 3 Trigger Signal Timing”, added	6.4.2, p.195
	Table 6-2 “DMA Configuration of Serial Data Reception”, DMA trigger factor changed	6.4.3 (1), p.196

(2/4)

Edition No.	Major items revised	Revised Sections
EE2	Figure 6-14 “Initialization of DMA Transfer for Serial Data Reception” added	6.4.3 (1), p.197
	Figure 6-15 “Operation of DMA Channel 4/5”, added	6.4.3 (1), p.198
	Figure 6-16 “DMA Channel 4 and 5 Trigger Signal Timing” added	6.4.3 (1), p.199
	Table 6-3 “DMA Configuration of Serial Data Transmission”, DMA trigger factor changed	6.4.3 (2), p. 200
	Figure 6-17 “Initialization of DMA Transfer for Serial Data Transmission” added	6.4.3 (2), p. 201
	Figure 6-18 “DMA Channel 6 and 7 Trigger Signal Timing”, added	6.4.3 (2), p.202
	Figure 6-19 “Operation of DMA Channel 6/7” added	6.4.3 (2), p.203
	Figure 6-20 “CPU and DMA Controller Processing of DMA Transfer Termination (Example)” added	6.4.4, p.204
	Table 6-4 “Relations Between DMA Trigger Factors and DMA Completion Interrupts”, DMA trigger factor changed	6.5, p. 205
	Figure 6-21 “Correlation between Serial I/O Interface Interrupts and DMA Completion Interrupts” added	6.5, p.206
	Figure 7-10 “Interrupt Control Register (PICn)”, value after reset and Man bit description changed	7.3.4, p.224
	Figure 7-13 “Interrupt Service Priority Register (ISPR)”, R/W changed to R	7.3.6, p.230
	Figure 7-15 “Interrupt Mode Register 0 (INTM0)”, description of bits ESN1 and ESN0 added	7.3.8 (1), p.233
	Figure 9-10 “TMPn Option Register 0 (TPnOPT0)”, description of TPnOVF flag changed	9.4 (6), p.258
	Figure 9-27 “Basic Operation Timing in PWM Mode”, remark 2 changed	9.5.6, p.283, 284
	10.3 (7) “TMRn I/O control register 4 (TRnIOC4)”, description changed	10.3 (7), p.318
	Figure 10-19 “TMRn Option Register 0 (TRnOPT0)” changed	10.3 (8), p.320
	10.3 (11) “TMRn option register 3 (TRnOPT3)” Caution changed	10.3(11), p.325
	Figure 10-35 “A/D Conversion Trigger Output Controller”, changed	10.8, p.362
	Figure 10-56 “High-Accuracy T-PWM Mode Block Diagram”, changed	10.10.9 (1), p.405
	10.10.9 (2) (d) “Interrupts” changed, Notes added	10.10.9 (2) (d), p.406
	10.10.9 (7) “Caution on timer output in high-accuracy T-PWM mode”, changed	10.10.9 (7), p.414-416
	Table 10-1 “Positive Phase Operation Condition List” changed	10.10.9 (8), p.417
	Table 10-2 “Negative Phase Operation Condition List” changed	10.10.9 (8), p.417
	10.10.9 (8) “Timer output change after compare register updating”, added	10.10.9 (8), p.417-429
	10.10.9 (9) “Dead time control in high-accuracy T-PWM mode”, changed	10.10.9 (9), p.430
	10.10.9 (10) “Cautions on dead time control in high-accuracy T-PWM mode”, added	10.10.9 (10), p.431
	10.10.9 (11) “Caution on rewriting cycles in high-accuracy T-PWM mode”, changed	10.10.9 (11), p.432
	10.10.9 (12) “Error interrupt (INTTRnER) in high-accuracy T-PWM mode”, changed	10.10.9 (12), p.433
	Figure 10-72 “Block Diagram in PWM Mode With Dead Time”, changed	10.10.10 (1), p.434
	10.10.10 (3) (e) “Reload thinning out function setting”, changed	10.10.10 (3) (e), p.437
	Table 11-3 “Capture/Compare Functions in Each Mode” changed	11.3 (1), p.447
	Table 11-4 “Capture/Compare Functions in Each Mode” changed	11.3 (2), p.449

(3/4)

Edition No.	Major items revised	Revised Sections
EE2	Figure 11-6 “TMTn Control Register 0 (TTnCTL0)”, changed	11.4 (1), p.452
	Figure 11-7 “TMTn Control Register 1 (TTnCTL1)”, changed	11.4 (2), p.453
	Figure 11-8 “TMTn Control Register 2 (TTnCTL2)”, changed	11.4 (3), p.455-456
	Figure 11-9 “TMTn I/O Control Register 0 (TTnIOC0)”, changed	11.4 (4), p.457
	Figure 11-11 “TMTn I/O Control Register 2 (TTnIOC2)”, changed	11.4 (6), p.459
	Figure 11-12 “TMTn I/O Control Register 3 (TTnIOC3)”, changed	11.4 (7), p.460, 461
	Figure 11-13 “TMTn Option Register 0 (TTnOPT0)”, changed	11.4 (8), p.462
	Figure 11-14 “TMTn Option Register 1 (TTnOPT1)” changed	11.4 (9), p.464
	Table 11-6 “Counter Clear Operation” changed	11.5.1 (2), p.467
	Table 11-7 “Capture/Compare Rewrite Methods in Each Mode” changed	11.5.2 82), p.472
	11.6.9 “Encoder count function” changed	11.6.9, p.499-500
	11.6.9 (6) (a) removed	11.6.9 (6), p.512
	Figure 11-37 (a) removed, ((c) changed, (i) and (j) removed	11.6.9 (6), p. 512-518
	Figure 11-38 “Basic Timing in Offset Trigger Generation Mode” changed	11.6.10, p.517
	Figure 14-1 “Block Diagram of A/D Converter (ADCn)”, changed	14.2, p.559
	Figure 14-5 “A/D Converter n Trigger Source Select Register (ADTRSELn)”, changed	14.3 (4), p.564
	14.4.2 (1) (b) “Timer trigger mode”, bit names changed	14.4.2 (1) (b), p.570
	14.6 “Operation in Timer Trigger Mode”, timer event signals’s names changed	14.6, p.580
	14.6.1 (1) “1-buffer mode operation (timer trigger select: 1 buffer)”, timer event signals’s names changed	14.6.1 (1), p.580
	Table 14-6 “Correspondence Between Analog Input Pins and ADCRnm Register (1-Buffer Mode (Timer Trigger Select: 1 Buffer)”, changed	14.6.1, p.581
	Figure 14-15 “Example of 1-Buffer Mode Operation (Timer Trigger Select: 1 Buffer) (ANIn1)”, changed	14.6.1, p.581
	14.6.1 (2) “4-buffer mode operation (timer trigger select: 4 buffers)”, timer event signals’s names changed	14.6.1 (2), p.582
	Table 14-7 “Correspondence Between Analog Input Pins and ADCRnm Register (4-Buffer Mode (Timer Trigger Select: 4 Buffers)”, changed	14.6.1, p.582
	Figure 14-16 “Example of 4-Buffer Mode Operation (Timer Trigger Select: 4 Buffers) (ANIn3)”, changed	14.6.1, p.583
	Table 14-8 “Correspondence Between Analog Input Pins and ADCRnm Register (Scan Mode (Timer Trigger Scan))”, changed	14.6.2, p.584
	Figure 14-17 “Example of Scan Mode Operation (Timer Trigger Scan) (ANIn0 to ANIn4)”, changed	14.6.2, p.585
	Figure 15-4 “UARTCn Control Register 2 (UCnCTL2)”, changed	15.3 (3), p.599
	Figure 16-28 “Prescaler Compare Registers 0 and 1 (PRSCM0, PRSCM1)”, changed	16.7.2 (2), p.659
	Figure 17-20 “Delay Control of Transmission/Reception Completion Interrupt (INTC3n)”, changed	17.5.14, p.692
	Figure 17-21 “Transfer Wait Function” (3/3), CSITn bit value changed from 0 to 1	17.5.15, p.695
	Figure 18-24 “CAN Global Clock Selection Register (CnGMCS)”, Bit 7 changed, Remark 1 changed	18.6.2, p.761
	Table 20-1 “Port Type and Function Overview”, Port CD changed	20.2.1, p.865

(4/4)

Edition No.	Major items revised	Revised Sections
EE2	Table 20-2 “Peripheral Registers of I/O Ports”, value after reset changed	20.2.3, p.890
	Figure 20-32 “Port Mode Control Register 3 (PMC3)”, bit PMC32 description changed	20.3.4 (2) (c), p.905
	Figure 20-34 “Port Mode Register 4 (PM4)”, bits 7 and 6 values changed from “0” to “1”	20.3.5 (2) (b), p.907
	Figure 20-47 “Port Mode Register 7 (PM7)”, bits 7 and 6 values changed from “0” to “1”	20.3.8 (2) (b), p.921
	Figure 20-50 “Port Mode Register 8 (PM8)”, bit 7 value changed from “0” to “1”	20.3.9 (2) (b), p.925
	Figure 20-53 “Port Mode Register 9 (PM9)”, bit 7 value changed from “0” to “1”	20.3.10 (2) (b), p.929
	Figure 20-56 “Port Mode Register 10 (PM10)”, bits 7 to 3 values changed from “0” to “1”, register address changed	20.3.11 (2) (b), p.933
	Figure 20-57 “Port Mode Control Register 10 (PMC10)”, register address changed	20.3.11 (2) (c), p.934
	Figure 23-3 “Example of Recommended Emulator Connection of V850E/PH2”, Cautions, voltage changed from 5 V to 3.3 V	23.2.1 (3), p.974
	Table 25-1 “Absolute Maximum Ratings”, V_{SS1} added, operating temperature conditions changed	25.1, p.997
	25.2.3 “Oscillator characteristics”, Remark, R1 changed to R	25.2.3, p.999
	Table 25-5 “DC Characteristics”, changed, Notes 1 and 2 added	25.3, p.1000
	Table 25-6 “External Asynchronous Memory Access Read Timing”, values changed, Remark 6 added	25.4.1, p.1002
	Figure 25-4 “External Asynchronous Memory Access Read Timing”, changed	25.4.1, p.1003
	Table 25-7 “External Asynchronous Memory Access Write Timing”, parameters and values changed, Remark 6 added	25.4.2, p.1004
	Figure 25-5 “External Asynchronous Memory Access Write Timing”, changed	25.4.2, p.1005
	Table 25-8 “Reset Timing”, $\overline{\text{RESET}}$ high level width deleted	25.4.3, p.1006
	Figure 25-6 “Reset Timing”, changed	25.4.3, p.1006
	Table 25-15 “CSI3 Characteristics (Master Mode)”, minimum value changed	25.5.2 (2), p.1014
	Table 25-17 “A/D Converter Characteristics” changed	25.5.3, p.1020
	Section 25.6 “Flash Programming Characteristics” added	25.6, p.1021
	Figure 26-2 “256-Pin Plastic BGA (Fine Pitch) (21 x 21)”, added	26, p.1024