NEC

Preliminary User's Manual

V850/DB1™ AVALON

32-/16-bit Single-Chip Microcontroller

Hardware

µPD70F3080
µPD703081

Document No. U15011EE2V0UM00
Date Published October 2002

Printed in Germany

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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 once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build 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 bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

MS-DOS and MS-Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
PC/AT and PC DOS are trademarks of IBM Corp.

The related documents in this publication may include preliminary versions. However, preliminary versions are not marked as such.

The export of this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by th customer. The export or re-export of this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

The information in this document is current as of 29.10.2002. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC's data sheets or data books, etc., for the most up-to-date specifications of NEC semiconductor products. Not all products and/or types are available in every country. Please check with an NEC 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 prior written consent of NEC. NEC assumes no responsibility for any errors that may appear in this document. NEC 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 semiconductor 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 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 customer's equipment shall be done under the full responsibility of customer. NEC assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information. While NEC endeavours to enhance the quality, reliability and safety of NEC semiconductor 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 semiconductor products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features. NEC semiconductor products are classified into the following three quality grades: "Standard", "Special" and "Specific". The "Specific" quality grade applies only to semiconductor products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of a semiconductor product depend on its quality grade, as indicated below. Customers must check the quality grade of each semiconductor 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": Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

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

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

M5 2000.03

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 01
Fax: 0211-65 03 327

Sucursal en España
Madrid, Spain
Tel: 091- 504 27 87
Fax: 091- 504 28 60

Succursale Française
Vélizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

Filiale Italiana
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

Branch The Netherlands
Eindhoven, The Netherlands
Tel: 040-244 58 45
Fax: 040-244 45 80

Branch Sweden
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

United Kingdom Branch
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
Singapore
Tel: 65-6253-8311
Fax: 65-6250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

NEC do Brasil S.A.
Electron Devices Division
Guarulhos, Brasil
Tel: 55-11-6465-6810
Fax: 55-11-6465-6829

Preface

Readers This manual is intented for users who want to understand the functions of the V850/DB1 (nickname Avalon).

Purpose This manual presents the hardware manual of V850/DB1.

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

1.1 General

The V850/DB1 single chip microcontroller, is a member of NEC's V850 32-bit RISC family, which match the performance gains attainable with RISC-based controllers to the needs of embedded control applications. The V850 CPU offers easy pipeline handling and programming, resulting in compact code size comparable to 16-bit CISC CPUs.

The V850/DB1 offers an excellent combination of general purpose peripheral functions, like serial communication interfaces (UART, clocked SI), timers and measurement inputs (A/D converter), with dedicated CAN network support. To support more than one network, two DCAN interfaces are implemented on chip. The device offers power-saving modes to manage the power consumption effectively under varying conditions. Thus equipped, the V850/DB1 is ideally suited for automotive applications.

(1) V850 CPU

The V850 CPU supports the RISC instruction set, and through the use of basic instructions that can each be executed in 1-clock period and an optimized pipeline, achieves marked improvements in instruction execution speed. In addition a 32-bit hardware multiplier enables this CPU to support multiply instructions, saturated multiply instructions, bit operation instructions, etc.
Also, through 2-byte basic instructions and instructions compatible with high level languages, etc., object code efficiency in a C compiler is increased, and program size can be made more compact. Further, since the on-chip interrupt controller provides high speed interrupt response, including processing, this device is suited for high level real time control fields.

(2) On-chip flash memory

The µPD70F3080 has on-chip an high speed flash memory, which is able to fetch one instruction within one clock cycle. It is possible to program the user application direct in the target application, on which the V850/DB1 is mounted. In such case system development time can be reduced and system maintainability after shipping can be markedly improved.

(3) A full range of development environment products

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

1.2 Device Features

- CPU

  - Core: V850
  - Number of instructions: 74
  - Min. instruction execution time: 62.5 ns (@ f_{CPU} = 16 MHz)
  - General registers: 32 × 32 bits

- Instruction set:
  - V850 (compatible with most instructions of V850E)
  - Signed multiplication (16 bits × 16 bits → 32 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

- Internal memory

Device	Part Number	Internal ROM	Internal RAM	DCAN
V850/DB1	µPD70F3080GJ-UEU	128 K Flash	6 K	2*288 bytes (16 transmit, 2 receive message buffers)
	µPD703081GJ-UEU	128 K Mask	6 K	1*288 bytes (16 transmit, 2 receive message buffers)

- Flash secure selfprogramming support

- Clock Generator
  - Internal PLL: 4 fold PLL
  - Frequency range: 16 MHz
  - Crystal frequency: $f_{CRYSTAL}$ = 4 MHz

- Built-in power saving modes: HALT, WATCH, STOP

- Power supply voltage range: $4.0\ V \le V_{DD} \le 5.5\ V$

- Temperature range: Ta = - 40 to + 85°C

- I/O lines: 83

- I lines: 8

- O lines: 16

- A/D Converter: 10-bit resolution; 8 channels

- Serial Interfaces
  - 3-wire mode: 3 channels
  - UART mode: 2 channels

- DCAN Interface: 2 channel

- Timers
  - 16-bit multi purpose timer/event counter: 1 channel
  - 16-bit OS timer 2 channel
  - 8-bit cascadable timer: 2 channel
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- Interrupts ≤ 44 vectored interrupts
- Package 128 LQFP, 0.5 mm pin-pitch

1.3 Application Fields

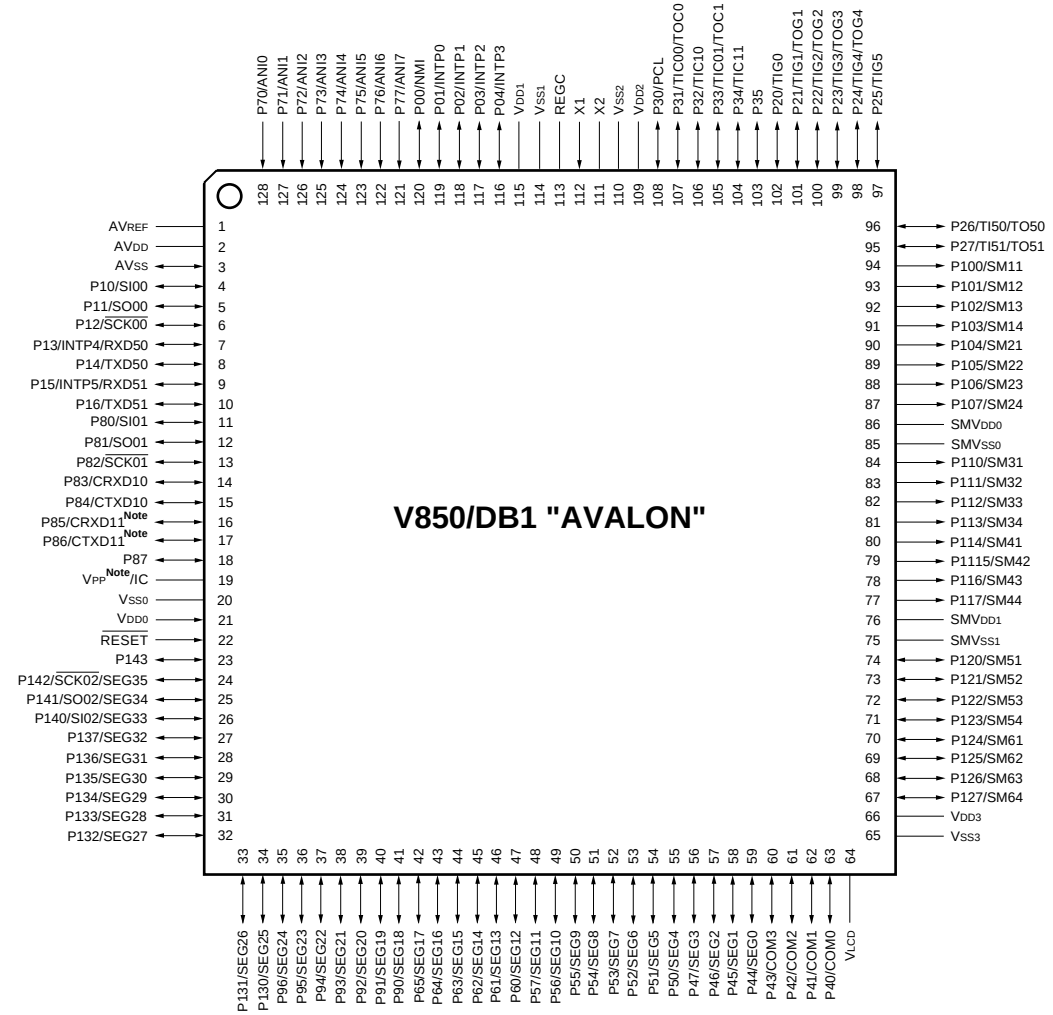
The V850/DB1 is a device designed for car manufacturers. It is ideally suited for automotive applications, like dashboard-, central body- or clime-control units. It is also an excellent choice for other applications where a combination of sophisticated peripheral functions with CAN network support is required.

1.4 Ordering Information

Device	Part Number	Package	Oper. Freq.	ROM	RAM	DCAN Option	Timer G	LCD	UART
V850/DB1	µPD70F3080GJ-UEU	LQFP128 (0.5 mm pitch) 20 x 20 mm	16 MHz	128 K Flash	6 K	2 Ch.	1 Ch.	4 x 36	2 Ch.
	µPD703081GJ-UEU			128 K Mask	6 K	1 Ch.	1 Ch.	4 x 36	2 Ch.

1.5 Pin Configuration (Top View)

Figure 1-1: Pin Configuration of the V850/DB1 Microcontroller



Note: µPD70F3080 (Flash version) only.

Pin Identification

ANI0 to ANI7	Analog Input	$\overline{\text{RESET}}$	Reset Input
AV_{DD}	Analog Power Supply	RXD50, RXD51	UART Receive Data
AV_{REF}	Reference Voltage ADC	$\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK02}}$	Synchronous Interface Clock
AV_{SS}	Ground	SEG0 to SEG35	LCD Segment Line
COM0 to COM3	LCD Common Line	SI00, SI01, SI02	Synchronous Interface Input
CRXD10, CRXD11**Note 1**	CAN Receive Data	SO00, SO01, SO02	Synchronous Interface Output
CTXD10, CTXD11**Note 1**	CAN Transmit Data	SM11 to SM14	Meter C/D Output (channel 1)
IC**Note 2**	Internal connected	SM21 to SM24	Meter C/D Output (channel 2)
INTP0 to INTP5	External Interrupt Input	SM31 to SM34	Meter C/D Output (channel 3)
NMI	Non-Maskable Interrupt Input	SM41 to SM44	Meter C/D Output (channel 4)
P00 to P04	Port 0	SM51 to SM54	Meter C/D Output (channel 5)
P10 to P16	Port 1	SM61 to SM64	Meter C/D Output (channel 6)
P20 to P27	Port 2	SMV_{DD0}, SMV_{DD1}	Power Supply for Port 10 to 12 (SM)
P30 to P35	Port 3	SMV_{SS0}, SMV_{SS1}	Ground for Port 10 to 12 (SM)
P40 to P47	Port 4	TI50, TI51	TM5 Count Input
P50 to P57	Port 5	TIC00, TIC01	TMC0 Capture Input
P60 to P65	Port 6	TIC10, TIC11	TMC1 Capture Input
P70 to P77	Port 7	TIG0 to TIG5	TMG Capture Input
P80 to P87	Port 8	TO50, TO51	TM5 Compare Output
P90 to P96	Port 9	TOC0, TOC1	TMC Compare Output
P100 to 107	Port 10	TOG1 to TOG4	TMG Compare Output
P110 to P117	Port 11	TXD50, TXD51	UART Transmit Data
P120 t P127	Port 12	V_{DD0} to V_{DD3}	Digital Power Supply
P130 to P137	Port 13	V_{LCD}	External LCD Voltage Input
P140 to P143	Port 14	V_{PP}**Note 1**	Programming Voltage
PCL	Processor Clock Output	V_{SS0} to V_{SS3}	Ground
REGC	Voltage Regulator Output	X1, X2	Main System Clock

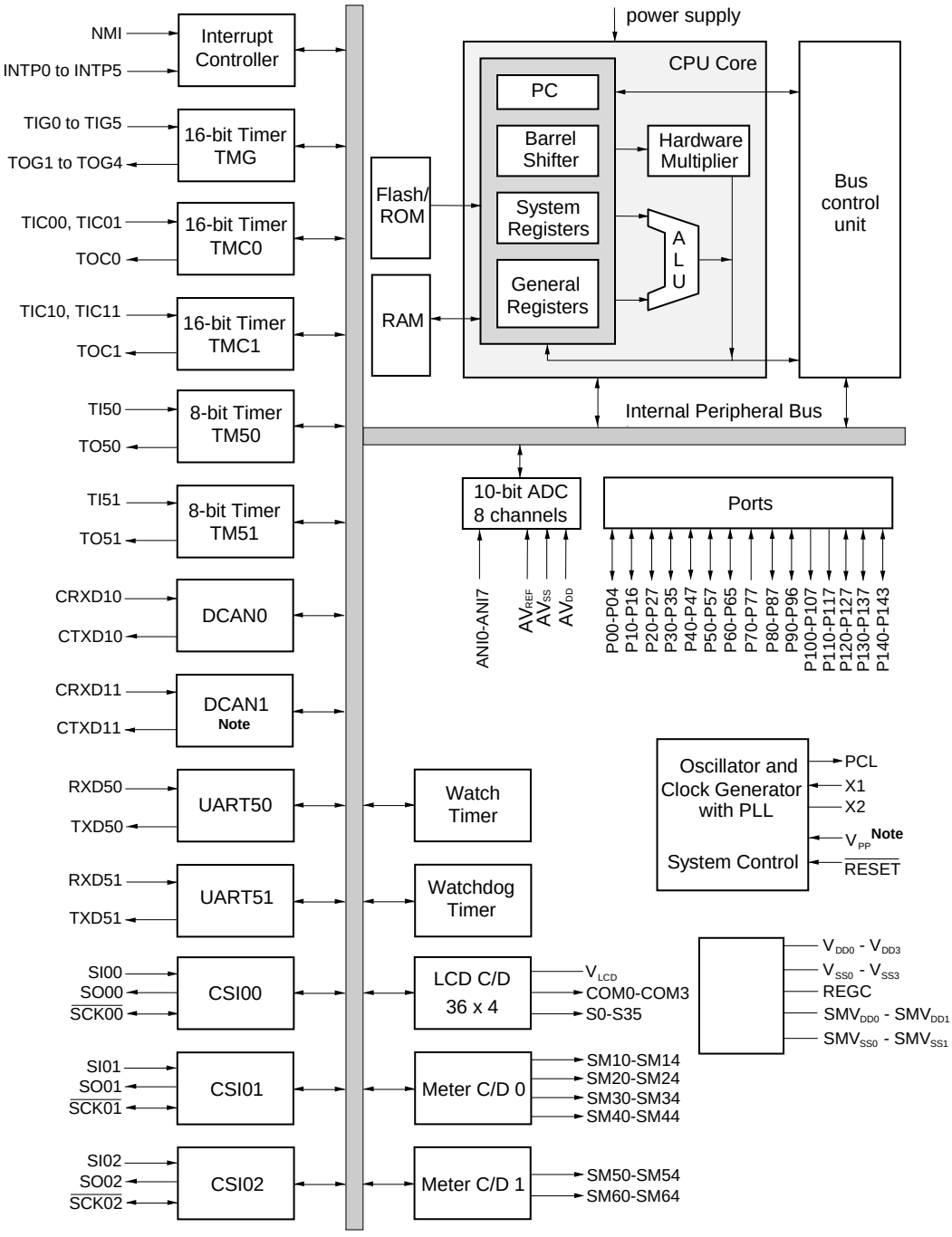
Notes: 1. µPD70F3080 (Flash version) only.

2. µPD703081 (Mask Version) only

1.6 Configuration of Function Block

1.6.1 Block Diagram of V850/DB1

Figure 1-2: Block Diagram of the V850/DB1 Microcontroller



Note: µPD70F3080 (Flash version) only.

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 the multiplier (16 bits × 16 bits → 32 bits) and the barrel shifter (32 bits), help accelerate processing of complex instructions.

(2) ROM

The µPD70F3080 has on-chip flash ROM (128 Kbytes).
The µPD703081 has on-chip mask ROM (128 Kbytes).
During instruction fetch, the memory can be accessed from the CPU in 1-clock cycles.
If the single chip mode or flash memory programming mode is set, memory mapping is done from address 00000000H.

(3) RAM

RAM are mapped from address FFFF D800H.
During instruction fetch, data can be accessed from the CPU in 1-clock cycles.

(4) Interrupt controller (INTC)

This controller handles hardware interrupt requests (NMI, INTP0 to INTP5) from on-chip peripheral I/O and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple-interrupt servicing control can be performed for interrupt sources.

(5) Clock generator (CG)

This clock generator supplies frequencies which are 4 times the input clock (f_{XX}) by the internal PLL and the input clock as an internal system clock (f_X) for the LCD, Watchdog Timer and Watch Timer. As the input clock, an external oscillator is connected to pins X1 and X2.

(6) Real-time pulse unit (RPU)

This unit has a16-bit multi purpose timer/event counter and 2 channels of 16-bit interval timer built in, and 2 channel of 8-bit cascadable timer/event counter. It is possible to measure pulse widths or frequency and to output a programmable pulse.

Timer G provides 6 inputs via the TIGn and 4 outputs via the TOGm pins (n = 0 to 5, m = 1 to 4).
Timer Cn provides 2 inputs via the TICnn and 1 output via the TOCn pin (n = 0 to 1).
Timer 5n provides 1 input via the TI5n and 1 output via the TO5n pin (n = 0 to 1).

(7) Watch Timer (WT)

The Watch Timer provides a continuously interrupt for clock applications. This macro provides two different clocks, that can be used simultaneously. Furthermore a Watch Timer Clock Generator is added to set a special timer period without software compensation.

(8) Watchdog Timer (WDT)

The Watchdog Timer detects a program runaway. When a runaway is detected, a hardware RESET an NMI or a maskable Interrupt can be generated.

(9) Serial interface (SIO)

A 2-channel asynchronous serial interface (UART5n), 3-channel clocked serial interface (CSI0n) and 2-channel DCAN are provided as serial interface.

UART5n transfers data by using the TXD5n and RXD5n pins. (n = 0, 1)
CSI0n transfers data by using the SO0n, SI0n, and $\overline{\text{SCK0n}}$ pins. (n = 0 to 2)
DCAN performs data transfer using CTXDn and CRXDn pins. (n = 0 to 2)

(10) A/D converter (ADC)

One high-resolution 10-bit A/D converter, it includes 8 analog input pins. Conversion uses the successive approximation method.
This A/D converter supported a Power-fail detection function. This function is to detect a voltage drop in the battery of an automobile. If the chosen condition for comparison is satisfied, an interrupt can be generated.

(11) Meter Controller/Driver (MTRC)

The meter controller/driver is a function to drive up to 6 stepping motors for external meter control or cross coil.

(12) LCD Controller/Driver (LCD)

This Controller supported an automatic output of segment signals and common signals for an LCD display. It generates the necessary AC voltage. For a good display contrast different modes are available.

(13) Ports

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

Port	Port Function	Control Function
Port 0	5-bit input/output	External interrupt input
Port 1	7-bit input/output	Serial interface (CSI00, UART50, UART51[Note]) input/output External interrupt input
Port 2	8-bit input/output	16-bit Timer G inputs/outputs 8-bit Timer 50/51 inputs/outputs
Port 3	6-bit input/output	Processor Clock output 16-bit Timer C0/C1 inputs/outputs
Port 4	8-bit input/output	LCD Segment output
Port 5	8-bit input/output	LCD Segment output
Port 6	6-bit input/output	LCD Segment output
Port 7	8-bit input	Analog/Digital Converter inputs
Port 8	8-bit input/output	Serial interface (CSI02, DCAN0, DCAN1[Note]) input/output
Port 9	7-bit input/output	LCD Segment output
Port 10	8-bit output	Meter-PWM output
Port 11	8-bit output	Meter-PWM output
Port 12	8-bit input/output	Meter-PWM output
Port 13	8-bit input/output	LCD Segment output
Port 14	4-bit input/output	Serial interface input/output (CSI02) LCD Segment output

Note: µPD70F3080 (Flash version) only.

[MEMO]

Chapter 2 Pin Functions

2.1 List of Pin Functions

The names and functions of this product's pins are listed below. These pins can be divided into port pins and non-port pins according to their functions.
Moreover, besides its function as a port, the most of them has functions as the input/output pins of on-chip peripheral I/O.
The selection of the port function or alternate peripheral function is mentioned in Table 2-1 in the Mode Logic column.

(1) Port pins

Table 2-1: Port pins (1/4)

Pin Name	I/O	Function	Driver Type	Pull Up	Mode Logic **Note**	Alternate
P00	I/O	Port 0: 5-bit input/output port	8-A	×	SIM	NMI
P01						INTP0
P02						INTP1
P03						INTP2
P04						INTP3
P10	I/O	Port 1: 7-bit input/output port	8-A	×	SIM	SI00
P11			5-A		OR	SO00
P12			8-A		AND	$\overline{\text{SCK00}}$
P13					SIM	INTP4/RXD50
P14			5-A		AND	TXD50
P15			8-A		SIM	INTP5/RXD51
P16			5-A		AND	TXD51
P20	I/O	Port 2: 8-bit input/output port	8-A	×	SIM	TIG0
P21					OR	TIG1/TOG1
P22					OR	TIG2/TOG2
P23					OR	TIG3/TOG3
P24					OR	TIG4/TOG4
P25					SIM	TIG5
P26					OR	TI50/TO50
P27					OR	TI51/TO51
P30	I/O	Port 3: 6-bit input/output port	5-A	×	OR	PCL
P31			8-A		OR	TIC00/TOC0
P32					SIM	TIC10
P33					OR	TIC01/TOC1
P34					SIM	TIC11
P35			5-A		-	-

Table 2-1: Port pins (2/4)

Pin Name	I/O	Function	Driver Type	Pull Up	Mode Logic **Note**	Alternate
P40	I/O	Port 4: 8-bit input/output port	18-C	–	Selector	COM0
P41						COM1
P42						COM2
P43						COM3
P44			17-G			SEG0
P45						SEG1
P46						SEG2
P47						SEG3
P50	I/O	Port 5: 8-bit input/output port	17-G	–	Selector	SEG4
P51						SEG5
P52						SEG6
P53						SEG7
P54						SEG8
P55						SEG9
P56						SEG10
P57						SEG11
P60	I/O	Port 6: 6-bit input/output port	17-G	–	Selector	SEG12
P61						SEG13
P62						SEG14
P63						SEG15
P64						SEG16
P65						SEG17
P70	I	Port 7: 8-bit input port	9	–	-	ANI0
P71						ANI1
P72						ANI2
P73						ANI3
P74						ANI4
P75						ANI5
P76						ANI6
P77						ANI7
P80	I/O	Port 8: 8-bit input/output port	8-A	×	SIM	SI01
P81			5-A		OR	SO01
P82			8-A		AND	$\overline{\text{SCK01}}$
P83					SIM	CRXD10
P84			5-A		AND	CTXD10
P85			8-A		SIM	CRXD11
P86			5-A		AND	CTXD11
P87					-	-

Table 2-1: Port pins (3/4)

Pin Name	I/O	Function	Driver Type	Pull Up	Mode Logic **Note**	Alternate
P90	I/O	Port 9: 7-bit input/output port	17-G	–	Selector	SEG18
P91						SEG19
P92						SEG20
P93						SEG21
P94						SEG22
P95						SEG23
P96						SEG24
P100	O	Port 10: 8-bit output port	4	–	Selector	SM11
P101						SM12
P102						SM13
P103						SM14
P104						SM21
P105						SM22
P106						SM23
P107						SM24
P110	O	Port 11: 8-bit output port	4	–	Selector	SM31
P111						SM32
P112						SM33
P113						SM34
P114						SM41
P115						SM42
P116						SM43
P117						SM44
P120	I/O	Port 12: 8-bit input/output port	5	–	Selector	SM51
P121						SM52
P122						SM53
P123						SM54
P124						SM61
P125						SM62
P126						SM63
P127						SM64
P130	I/O	Port 13: 8-bit input/output port	17-G	–	Selector	SEG25
P131						SEG26
P132						SEG27
P133						SEG28
P134						SEG29
P135						SEG30
P136						SEG31
P137						SEG32

Table 2-1: Port pins (4/4)

Pin Name	I/O	Function	Driver Type	Pull Up	Mode Logic **Note**	Alternate
P140	I/O	Port 14: 4-bit input/output port	17-G	–	SIM	SI02/SEG33
P141			17-G		OR/Selector	SO02/SEG34
P142			17-G		AND/Selector	$\overline{\text{SCK02}}$/SEG35
P143			5		-	-

Note: The port pins of the V850/DB1 has different properties for the output/input mode configuration. The following list shows the meaning of the abbreviation:

- SIM: the port function and alternate function are **simultaneously** available, this is only possible for alternate input functions
- AND: the port function and alternate function are **AND** connected (i.e. if only the alternate function should output then the port has to be set to “1“)
- OR: the port function and alternate function are **OR** connected, (i.e. if only the alternate function should output then the port has to be set to “0“)
- Selector: the port function and alternate function are connected via a hardware **selector**, that is software controlled.

(2) Non-port pins

Table 2-2: Non-port pins (1/2)

Pin Name	I/O	Function	Alternate Function
Only Port Pins	I/O	Input/output port	P35, P87, P143
ANI0 to ANI7	Input	A/D converter input pin	P70 to P77
AV_{DD}	-	Power supply pin for A/D converter	-
AV_{REF}	-	Reference-Voltage supply pin for A/D converter	-
AV_{SS}	-	Ground potential for A/D converter	-
COM0 to COM3	Output	LCD common signal output pin	P40 to P43
CRXD10	Input	CAN channel 0 serial data input pin	P83
CRXD11**Note 1**	Input	CAN channel 1 serial data input pin	P85
CTXD10	Output	CAN channel 0 serial data output pin	P84
CTXD11**Note 1**	Output	CAN channel 1 serial data output pin	P86
INTP0 to INTP3	Input	Maskable interrupt input pin	P01 to P04
INTP4	Input	Maskable interrupt input pin	P13/RXD50
INTP5	Input	Maskable interrupt input pin	P15/RXD51
NMI	Input	Non-maskable interrupt input pin	P00
PCL	Output	Processor Clock output pin	P30
REGC	-	Pin for external 3.3 V Back-Up-Capacitor	-
$\overline{RESET}$	Input	external System reset input	-
RXD50	Input	UART5 channel 0 Serial data input pin	P13/INTP4
RXD51**Note 2**	Input	UART5 channel 1 Serial data input pin	P15/INTP5
$\overline{SCK00}$	I/O	CSI0 channel 0 Serial clock input pin	P12
$\overline{SCK01}$	I/O	CSI0 channel 1 Serial clock input pin	P82
$\overline{SCK02}$	I/O	CSI0 channel 2 Serial clock input pin	P142/SEG35
SEG0 to SEG3	Output	LCD segment signal output pin	P44 to P47
SEG4 to SEG11	Output	LCD segment signal output pin	P50 to P57
SEG12 to SEG17	Output	LCD segment signal output pin	P60 to P65
SEG18 to SEG24	Output	LCD segment signal output pin	P90 to P96
SEG25 to SEG32	Output	LCD segment signal output pin	P130 to P137
SEG33	Output	LCD segment signal output pin	P140/SI02
SEG34	Output	LCD segment signal output pin	P141/SO02
SEG35	Output	LCD segment signal output pin	P142/$\overline{SCK02}$
SI00	Input	CSI0 channel 0 Serial data input pin	P10
SI01	Input	CSI0 channel 1 Serial data input pin	P80
SI02	Input	CSI0 channel 2 Serial data input pin	P140/SEG33
SM11 to SM24	Output	Meter drive output pin	P100 to P107
SM31 to SM44	Output	Meter drive output pin	P110 to P117
SM51 to SM64	Output	Meter drive output pin	P120 to P127
SMV_{DD0}	-	Power supply pin for Ports P100 - P107, P110 - P113 (Meter Controller/Driver for SM11 - SM34)	-
SMV_{DD1}	-	Power supply pin for Ports P114 - P117, P120 - P127 (Meter Controller/Driver for SM41 - SM64)	-

Table 2-2: Non-port pins (2/2)

Pin Name	I/O	Function	Alternate Function
SMV_{SS0}	-	Ground potential for Meter Controller/Driver for SM11 - SM34 P100 - P107, P110 - P113	-
SMV_{SS1}	-	Ground potential for Meter Controller/Driver for SM41 - SM64 P114 - P117, P120 - P127	-
SO00	Output	CSI0 channel 0 Serial data output pin	P11
SO01	Output	CSI0 channel 1 Serial data output pin	P81
SO02	Output	CSI0 channel 2 Serial data output pin	P141/SEG34
TI50	Input	Event counter input channel 0 pin	P26/TO50
TI51	Input	Event counter input channel 1 pin	P27/TO51
TIC00	Input	Timer C0 Capture trigger input pin	P31/TOC0
TIC10	Input	Timer C0 Capture trigger input pin	P32
TIC01	Input	Timer C1 Capture trigger input pin	P33/TOC1
TIC11	Input	Timer C1 Capture trigger input pin	P34
TIG0	Input	Timer G Capture trigger input pin	P20
TIG1 to TIG4	Output	Timer G PWM capture input pin	P21 to P24/ TOG1 to TOG4
TIG5	Input	Timer G Capture trigger input pin	P25
TO50	Output	Timer 50 PWM output channel 0 pin	P26/TI50
TO51	Output	Timer 51 PWM output channel 1 pin	P27/TI51
TOC0	Output	Timer C0 output pin	P31/TIC00
TOC1	Output	Timer C1 output pin	P33/TIC01
TOG1 to TOG4	Output	Timer G PWM output pin	P21 to P24/ TIG1 to TIG4
TXD50	Output	UART5 channel 0 Serial data output pin	P14
TXD51	Output	UART5 channel 1 Serial data output pin	P16
V_{DD0}	-	Power supply pin for internal peripherals	-
V_{DD1}	-	Power supply pin for CPU (Voltage Regulator)	-
V_{DD2}	-	Power supply pin for internal peripherals	-
V_{DD3}	-	Power supply pin for internal peripherals	-
V_{LCD}	-	Power supply pin for external LCD-Voltage supply	-
V_{PP}**Note**	-	High Voltage apply pin for program write/verify	-
V_{SS0}	-	Ground potential pin for internal peripherals	-
V_{SS1}	-	Ground potential pin for CPU	-
V_{SS2}	-	Ground potential pin for internal peripherals	-
V_{SS3}	-	Ground potential pin for internal peripherals	-
X1, X2	-	Resonator connection for main-clock	-

Note: Only for µPD70F3080 (Flash product)

Cautions: 1. On REGC-pin and each pin of V_{DDn}, a capacitor has to be attached as tight as possible to the pin. The capacitors used should have only very low serial impedance.

(3) Pin related to V850/DB1 status

Table 2-3: Pin related to V850/DB1 status

Pin name	HALT mode	WATCH mode	STOP mode	In Reset	After Reset
P00 to 04	Status Hold	Status Hold	Status Hold	Hi-Z	Hi-Z
P10 to 16					
P20 to 27					
P30 to 35					
P40 to 47					
P50-57					
P60 to 65					
P70 to 77					
P80 to 86[Note]					
P90 to 96					
P100 to 107					
P110 to 117					
P120 to 127					
P130 to 137					
P140 to 143					

Note: Output values must be set to recessive level by software before activating standby mode. Otherwise CAN bus might be continuously blocked by dominant level.

2.2 Description of Pin Functions

(1) P00 to P14 (Port 0) … Input/output

Port 0 is an 5-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, P00 operates as NMI and P01 to P04 operates as external interrupt request input pins.

(a) Port mode

P00 to P04 can be set to input or output in 1-bit units using the port 0 mode register (PM0).

(b) Input/Output Control

The inputs/outputs of the port functions and the alternative functions are shared simultaneously at the same time.

(c) Pull Up Resistor Control

The outputs of the buffer include a pull up resistor which can be activated via a control bit of the Pull Up Resistor Option register (PU0).

(d) NMI (NON-Maskable Interrupt Request)... Input

This is the non-maskable interrupt request input pin.

(e) INPT00 to INTP50 (Interrupt request from peripherals) … Input

These are maskable external interrupt request input pins.

(2) P10 to P16 (Port 1) … Input/output

Port 1 is an 7-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, P10 to P16 operate as serial interface (CSI00, CSI01, UART50) input/output and furthermore P13, P15 operates as maskable external interrupt request input pins.

(a) Port mode

P10 to P16 can be set to input or output in 1-bit units using the port 1 mode register (PM1).

(b) Input/Output Control

The inputs/outputs of pin P10, P13 and P15 port functions and the alternative functions are shared simultaneously at the same time.
The inputs/outputs of pin P11 port function and the alternative functions are shared with an OR-function at the same time.
The inputs/outputs of pin P12, P14 and P16 port function and the alternative functions are shared with an AND-function at the same time.

(c) Pull Up Resistor Control

The outputs of the buffer include a pull up resistor which can be activated via a control bit of the Pull Up Resistor Option register (PU1).

(d) SO00 (Serial output) … Output

These pin output CSI00 serial transmit data.

(e) SI00 (Serial input) … Input

These pin input CSI00 serial receive data.

(f) $\overline{\text{SCK00}}$ (Serial clock) … Input/output

These is CSI00 serial clock input/output pin.

(g) TXD50, TXD51 (Transmit data) … Output

These pins output serial transmit data of UART50 and UART51.

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

Port 2 is a 8-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, in control mode, P20 to P25 operate as Real-time pulse unit (RPU) (Timer G) input/output.

(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) Input/Output Control

The inputs/outputs of pin P20 and P25 port functions and the alternative functions are shared simultaneously at the same time.
The inputs/outputs of pin P21 to P24 and P26, P27 port function and the alternative functions are shared with an OR-function at the same time.

(c) Pull Up Resistor Control

The outputs of the buffer include a pull up resistor which can be activated via a control bit of the Pull Up Resistor Option register (PU2).

(d) TOG1 to TOG4 (Timer G output) … Output

These pins output a Timer G pulse signal.

(e) TIG0 to TIG5 (Timer G input) … Input

These pins input a Timer G external capture input pin.

(f) TO50, TO51 (Timer 5 output) … Output

These pins output a Timer 5 pulse signal.

(g) TI50, TI51 (Timer 5 input) … Input

These pins input a Timer 5 external count input pin.

(4) P30 to P36 (Port 3) … Input/Output

Port 3 is a 8-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, in control mode, P30 operates as PCL and P31 to P34 operate as Real-time pulse unit (RPU) (Timer C) input/output.

(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) Input/Output Control

The inputs/outputs of pin P32, P34 and P35 port functions and the alternative functions are shared simultaneously at the same time.
The inputs/outputs of pin P30, P31 and P33 port function and the alternative functions are shared with an OR-function at the same time.

(c) Pull Up Resistor Control

The outputs of the buffer include a pull up resistor which can be activated via a control bit of the Pull Up Resistor Option register (PU3).

(d) PCL (Processor Clock Output) … Output

This is an internal system clock output pin.

(e) TOC0 to TOC1 (Timer C output) … Output

These pins output a Timer C0, Timer C1 pulse signal.

(f) TIC00 to TIC11 (Timer C input) … Input

These pins input a Timer C0, Timer C1 external capture input pin.

(5) P40 to P47 (Port 4) … Input/output

Port 4 is a 8-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, in control mode, P40 to P47 operate as segment signal output of LCD controller/driver.
An operation mode of port or control mode can be selected for each bit and specified by the port 4 mode control register (PMC4).

(a) Port mode

P40 to P47 can be set to input or output in 1-bit units using the port 4 mode register (PM4).

(b) Control mode

P40 to P47 can be set to port or control mode in 1-bit units using PMC4.

(c) COM0 - COM3 (Common signal)... Output

These are the common signal output of LCD-controller/driver

(d) SEG0 - SEG3 Segment Signal output of LCD controller/driver … Output

These pins functions as segment signal output of LCD controller/driver.

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

Port 5 is a 8-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, in control mode, P50 to P57 operate as segment signal output of LCD controller/driver.
An operation mode of port or control mode can be selected for each bit and specified by the port 5 mode control register (PMC5).

(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 PMC5.

(c) SEG4 - SEG11 Segment Signal output of LCD controller/driver … Output

These pins functions as segment signal output of LCD controller/driver.

(7) P60 to P65 (Port 6) … Input/output

Port 6 is a 6-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, in control mode, P60 to P65 operate as segment signal output of LCD controller/driver.
An operation mode of port or control mode can be selected for each bit and specified by the port 6 mode control register (PMC6).

(a) Port mode

P60 to P65 can be set to input or output in 1-bit units using the port 6 mode register (PM6).

(b) Control mode

P60 to P65 can be set to port or control mode in 1-bit units using PMC6.

(c) SEG12 - SEG17 Segment Signal output of LCD controller/driver … Output

These pins functions as segment signal output of LCD controller/driver.

(8) P70 to P75 (Port 7) … Input

Port 7 is a 8-bit input port in which input or output can be set in 1-bit units.
Besides functioning as an input port, P70 to P77 operate as analog input pins to the A/D converter.
An operation mode of port or control mode can be selected for each bit and specified by the port 7 mode control register (PMC7).

(a) Port mode

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

(b) Input/Output Control

The inputs of pin P70 to P77 port functions and the alternative functions are shared simultaneously at the same time.

(c) ANI0 to ANI7 (Analog input) … Input

These are analog input pins to the A/D converter.

(9) P80 to P87 (Port 8) … Input/output

Port 1 is an 8-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, in control mode, P80 to P87 operate as serial interface (CSI01, DCAN0, DCAN1) input/output.

(a) Port mode

P80 to P87 can be set to input or output in 1-bit units using the port 8 mode register (PM8).

(b) Input/Output Control

The inputs/outputs of pins P80, P83, P85 and P87 port functions and the alternative functions are shared simultaneously at the same time.
The input/output of pin P81 port function and the alternative function is shared with an OR-function at the same time.
The inputs/outputs of pins P82, P84 and P86 port function and the alternative functions are shared with an AND-function at the same time.

(c) Pull Up Resistor Control

The outputs of the buffer include a pull up resistor which can be activated via a control bit of the Pull Up Resistor Option register (PU8).

(d) SO01 (Serial output) … Output

This pin outputs CSI01 serial transmit data.

(e) SI01 (Serial input) … Input

This pin input CSI01 serial receive data.

(f) $\overline{\text{SCK01}}$ (Serial clock) … Input/output

These are CSI01 serial clock input/output pins.

(g) CTXD10, CTXD11[Note] (Transmit data for controller area network) … Output

This pin outputs DCAN0, DCAN1 serial transmit data.

(h) CRXD10, CRXD11[Note] (Receive data for controller area network) … Input

This pin inputs DCAN0, DCAN1 serial receive data.

Note: Only in µPD70F3080 (128 K Flash product)

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

Port 9 is a 7-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, in control mode, P90 to P96 operate as segment signal output of LCD controller/driver.
An operation mode of port or control mode can be selected for each bit and specified by the port 9 mode control register (PMC9).

(a) Port mode

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

(b) Control mode

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

(c) SEG18 - SEG24 Segment Signal output of LCD controller/driver … Output
These pins functions as segment signal output of LCD controller/driver.

(11) P100 to P107 (Port 10) … Input/output

Port 10 is a 8-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, in control mode, P100 to P107 operate as pulse output of Meter Controller/Driver.
An operation mode of port or control mode can be selected for each bit and specified by the port 10 mode control register (PMC10).

(a) Port mode

P100 to P107 can be set to input or output in 1-bit units using the port 10 mode register (PM10).

(b) Control mode

P100 to P107 can be set to port or control mode in 1-bit units using PMC10.

(c) SM51 - SM64 PWM output of Meter controller/driver … Output

These pins functions as PWM output of Meter controller/driver.

(12) P110 to P117 (Port 11) … Input/output

Port 11 is a 8-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, in control mode, P110 to P117 operate as pulse output of Meter Controller/Driver.
An operation mode of port or control mode can be selected for each bit and specified by the port 11 mode control register (PMC11).

(a) Port mode

P110 to P117 can be set to input or output in 1-bit units using the port 11 mode register (PM11).

(b) Control mode

P110 to P117 can be set to port or control mode in 1-bit units using PMC11.

(c) SM31 - SM44 PWM output of Meter controller/driver … Output

These pins functions as PWM output of Meter controller/driver.

(13) P120 to P127 (Port 12) … Input/output

Port 12 is a 8-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, in control mode, P120 to P127 operate as pulse output of Meter Controller/Driver.
An operation mode of port or control mode can be selected for each bit and specified by the port 12 mode control register (PMC12).

(a) Port mode

P120 to P127 can be set to input or output in 1-bit units using the port 12 mode register (PM12).

(b) Control mode

P120 to P127 can be set to port or control mode in 1-bit units using PMC12.

(c) SM31 - SM44 PWM output of Meter controller/driver … Output

These pins functions as PWM output of Meter controller/driver.

(14) P130 to P136 (Port 13) … Input/output

Port 13 is a 7-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, in control mode, P130 to P136 operate as segment signal output of LCD controller/driver.
An operation mode of port or control mode can be selected for each bit and specified by the port 13 mode control register (PMC13).

(a) Port mode

P130 to P136 can be set to input or output in 1-bit units using the port 13 mode register (PM13).

(b) Control mode

P130 to P136 can be set to port or control mode in 1-bit units using PMC13.

(c) SEG25 - SEG32 Segment Signal output of LCD controller/driver … Output

These pins functions as segment signal output of LCD controller/driver.

(15) P140 to P143 (Port 14) … Input/output

Port 14 is a 4-bit input/output port in which input or output can be set in 1-bit units.
Besides functioning as an input/output port, in control mode, P140 to P143 operate as segment signal output of LCD controller/driver.
An operation mode of port or control mode can be selected for each bit and specified by the port 14 mode control register (PMC14).

(a) Port mode

P140 to P143 can be set to input or output in 1-bit units using the port 14 mode register (PM14).

(b) Control mode

P140 to P143 can be set to port or control mode in 1-bit units using PMC14.

(c) SO02 (Serial output) … Output

This pin outputs CSI02 serial transmit data.

(d) SI02 (Serial input) … Input

This pin inputs CSI02 serial receive data.

(e) $\overline{\text{SCK02}}$ (Serial clock) … Input/output

These are CSI02 serial clock input/output pins.

(f) SEG33 - SEG35 Segment Signal output of LCD controller/driver … Output

These pins function as segment signal output of LCD controller/driver.

(16) V_{PP}/IC

(a) V_{PP} (Flash Memory Programming Voltage) only for µPD70F3080 (128 K Flash product)
High voltage apply pin for FLASH programming mode setting. Connect to V_{SS} in normal operating mode. To use this pin for on board flash programming connect this pin with a pull down resistor to V_{SS}.

(b) IC (Internal Connection) for µPD703081 (Mask product)
For the Mask products this pin is internal connected.

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

$\overline{\text{RESET}}$ input is asynchronous input. When a signal having a certain low level width is input in asynchronous with the operation clock, a system reset that takes precedence over all operations occurs.
Besides a normal initialize or start, this signal is also used to release a standby mode (HALT, WATCH, software STOP).

(18) NMI (Non-Maskable Interrupt Request)... Input

This is the non-maskable interrupt request input pin.

(19) X1, X2 (Crystal) … Output

These pins connect a resonator or crystal for system clock generation.

(20) REGC (Capacitor) … Output

These pin connects an external Capacitor for the internal voltage system. This capacitor is used as a back up capacitor for the internal voltage generator.

(21) V_{DD0} to V_{DD3} (Power supply) … Input

This are the positive power supply pins for the CPU and the peripherals.

(22) V_{SS0} to V_{SS3} (Ground) … Input

This are the ground pins for the CPU and the peripherals.

(23) SMV_{DD0}, SMV_{DD1} (Meter-PWM Power Supply) … Input

These are the positive power supply pins for the Port buffer of P100 to P127 (Stepper Motor Driver).

(24) SMV_{SS0}, SMV_{SS1} (Meter-PWM Power Supply) … Input

These are the negative power supply pins for the Port buffer of P100 to P127 (Stepper Motor Driver).

(25) AV_{DD} (Analog power supply) … Input

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

(26) AV_{SS} (Analog ground) … Input

This is the ground pin for the A/D converter.

(27) AV_{REF} (Analog reference voltage) … Input

These are the reference voltage supply pins for the A/D converter.

2.3 Types of Pin I/O Circuit and Connection of Unused Pin

Table 2-4: Types of Pin I/O Circuit and Connection of Unused Pin (1/4)

Pin Name			I/O Circuit Type	I/O Buffer Power Supply	Recommended Connection for Unused Pins
P00	INTP0		8-A	V_{DD2}	For input: Individually connect to V_{DD2} or V_{SS2} via a resistor For output: Leave open
P01	INTP1				
P02	INTP2				
P03	INTP3				
P04	INTP4				
P10	SI00		8-A	V_{DD0}, V_{DD3}	For input: Individually connect to V_{DD0}, V_{DD3} or V_{SS0}, V_{SS3} via a resistor For output: Leave open
P11	SO00		5-A		
P12	$\overline{SCK00}$		8-A		
P13	RXD50	INTP5			
P14	TXD50		5-A		
P15	RXD51	INTP6	8-A		
P16	TXD51		5-A		
P20	TIG0		8-A	V_{DD2}	For input: Individually connect to V_{DD2} or V_{SS2} via a resistor For output: Leave open
P21	TIOG1				
P22	TIOG2				
P23	TIOG3				
P24	TIOG4				
P25	TIG5				
P26	TIO50				
P27	TIO51				
P30	PCL		5-A	V_{DD2}	For input: Individually connect to V_{DD2} or V_{SS2} via a resistor For output: Leave open
P31	TIC00	TOC0	8-A		
P32	TIC10				
P33	TIC01	TOC1			
P34	TIC11				
P35			5-A		
P40	COM0		18-C	V_{DD0}, V_{DD3}	For input: Individually connect to V_{DD0}, V_{DD3} or V_{SS0}, V_{SS3} via a resistor For output: Leave open
P41	COM1				
P42	COM2				
P43	COM3				
P44	SEG0		17-G		
P45	SEG1				
P46	SEG2				
P47	SEG3				

Table 2-4: Types of Pin I/O Circuit and Connection of Unused Pin (2/4)

Pin Name			I/O Circuit Type	I/O Buffer Power Supply	Recommended Connection for Unused Pins
P50	SEG4		17-G	V_{DD0}, V_{DD3}	For input: Individually connect to V_{DD0}, V_{DD3} or V_{SS0}, V_{SS3} via a resistor For output: Leave open
P51	SEG5				
P52	SEG6				
P53	SEG7				
P54	SEG8				
P55	SEG9				
P56	SEG10				
P57	SEG11				
P60	SEG12		17-G	V_{DD0}, V_{DD3}	For input: Individually connect to V_{DD0}, V_{DD3} or V_{SS0}, V_{SS3} via a resistor For output: Leave open
P61	SEG13				
P62	SEG14				
P63	SEG15				
P64	SEG16				
P65	SEG17				
P70	ANI0		9	AV_{DD}	Individually connect to AV_{DD} or AV_{SS} via a resistor
P71	ANI1				
P72	ANI2				
P73	ANI3				
P74	ANI4				
P75	ANI5				
P76	ANI6				
P77	ANI7				
P80	SI01		8-A	V_{DD0}, V_{DD3}	For input: Individually connect to V_{DD0}, V_{DD3} or V_{SS0}, V_{SS3} via a resistor For output: Leave open
P81	SO01		5-A		
P82	$\overline{\text{SCK01}}$		8-A		
P83	CRXD10				
P84	CTXD10		5-A		
P85	CRXD11		8-A		
P86	CTXD11		5-A		
P87					
P90	SEG18		17-G	V_{DD0}, V_{DD3}	For input: Individually connect to V_{DD0}, V_{DD3} or V_{SS0}, V_{SS3} via a resistor For output: Leave open
P91	SEG19				
P92	SEG20				
P93	SEG21				
P94	SEG22				
P95	SEG23				
P96	SEG24				

Table 2-4: Types of Pin I/O Circuit and Connection of Unused Pin (3/4)

Pin Name			I/O Circuit Type	I/O Buffer Power Supply	Recommended Connection for Unused Pins
P100	SM11		4	SMV_{DD0}	Leave open
P101	SM12				
P102	SM13				
P103	SM14				
P104	SM21				
P105	SM22				
P106	SM23				
P107	SM24				
P110	SM31		4	SMV_{DD0}	Leave open
P111	SM32				
P112	SM33				
P113	SM34				
P114	SM41			SMV_{DD1}	
P115	SM42				
P116	SM43				
P117	SM44				
P120	SM51		5	SMV_{DD1}	For input: Individually connect to SMV_{DD1} or SMV_{SS1} via a resistor For output: Leave open
P121	SM52				
P122	SM53				
P123	SM54				
P124	SM61				
P125	SM62				
P126	SM63				
P127	SM64				
P130	SEG25		17-G	V_{DD0}, V_{DD3}	For input: Individually connect to V_{DD0}, V_{DD3} or V_{SS0}, V_{SS3} via a resistor For output: Leave open
P131	SEG26				
P132	SEG27				
P133	SEG28				
P134	SEG29				
P135	SEG30				
P136	SEG31				
P137	SEG32				
P140	SEG33	SI02	17-G	V_{DD0}, V_{DD3}	For input: Individually connect to V_{DD0}, V_{DD3} or V_{SS0}, V_{SS3} via a resistor For output: Leave open
P141	SEG34	SO02	17-G		
P142	SEG35	$\overline{SCK02}$	17-G		
P143			5		
X1, X2			-	-	has to be used
$\overline{RESET}$					
V_{LCD}			-	-	Leave open

Table 2-4: Types of Pin I/O Circuit and Connection of Unused Pin (4/4)

Pin Name			I/O Circuit Type	I/O Buffer Power Supply	Recommended Connection for Unused Pins
SMV_{DD0} [SM11-SM34]			-	-	has to be used
SMV_{DD1} [SM41-SM64]					
SMV_{SS0} [SM11-SM34]					
SMV_{SS1} [SM41-SM64]					
V_{DD0}					
V_{DD1}					
V_{DD2}					
V_{DD3}					
V_{SS0}					
V_{SS1}					
V_{SS2}					
V_{SS3}					
AV_{DD}			-	-	Connect to V_{DD}
AV_{REF}					
AV_{SS}			-	-	Connect to V_{SS}
V_{PP}/IC					
REGC			-	-	has to be used

Remark: Please aware, that the port pins are in input mode after RESET release.

Figure 2-1: Pin I/O Circuits (1/2)

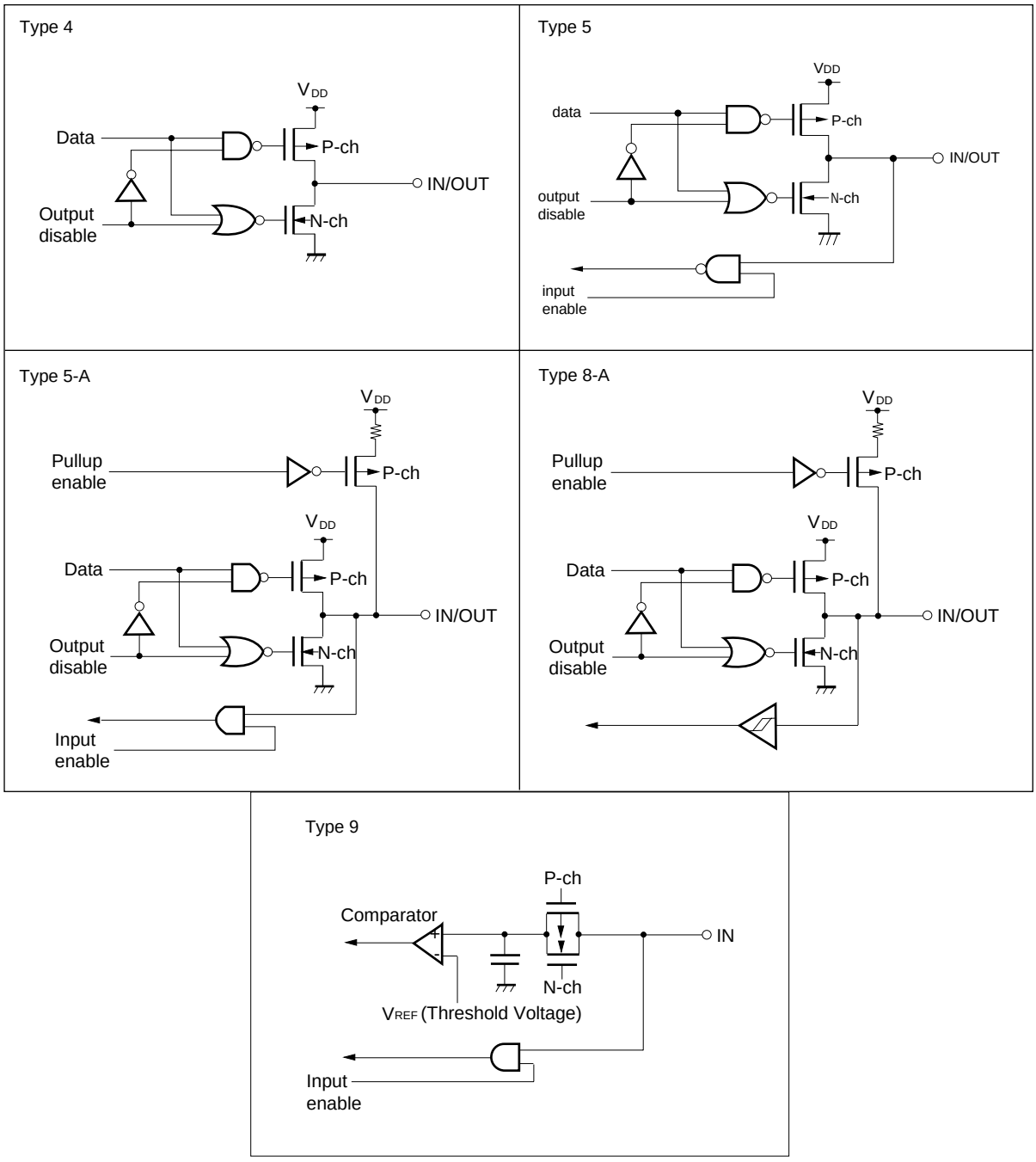
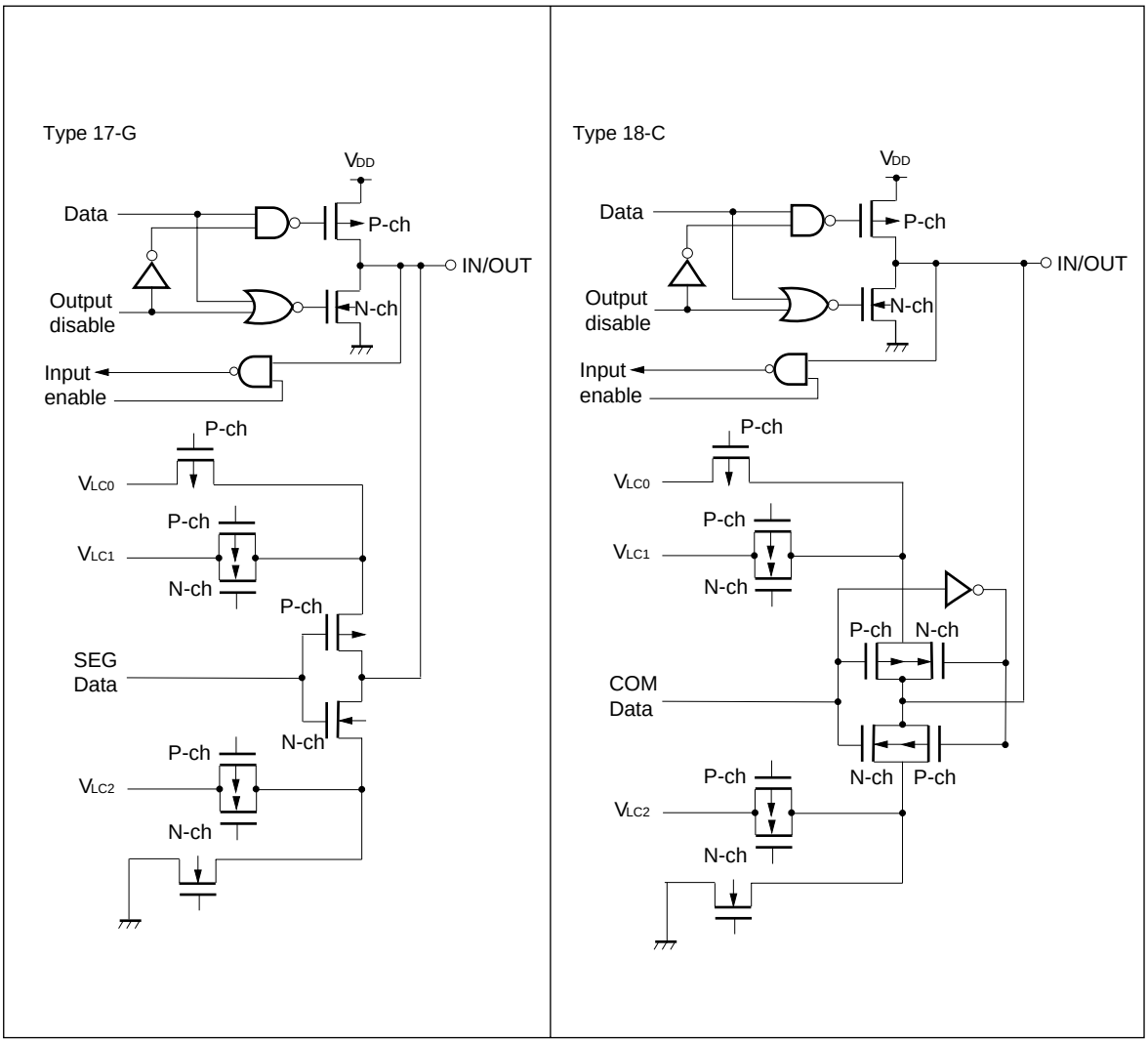


Figure 2-1: Pin I/O Circuits (2/2)



Chapter 3 CPU Functions

The CPU of the V850/DB1 is based on the RISC architecture and executes most instructions in one clock cycle by using a 5-stage pipeline.

3.1 Features

- Minimum instruction execution time: 62.5 ns (@ 16 MHz internal operation)
- Memory space:
  - Program space: 16 MB linear
  - Data Space: 4 GB linear
- Thirty-two 32-bit general-purpose registers
- Internal 32-bit architecture
- Five-stage pipeline control
- Multiplication/division instructions
- Saturated operation instructions
- One-clock 32-bit shift instruction
- Load/store instruction with long/short format
- Four types of bit manipulation instructions
  - Set
  - Clear
  - Not
  - Test

3.2 CPU Register Set

The CPU registers of the V850/DB1 can be classified into two categories: a general-purpose program register set and a dedicated system register set. All the registers are 32 bits width. For details, refer to V850 Family User's Manual Architecture.

Figure 3-1: CPU Register Set

Program register set

31	0
r0	Zero Register
r1	Reserved for Address Register
r2	Interrupt Stack Pointer
r3	Stack Pointer (SP)
r4	Global Pointer (GP)
r5	Text Pointer (TP)
r6	
r7	
r8	
r9	
r10	
r11	
r12	
r13	
r14	
r15	
r16	
r17	
r18	
r19	
r20	
r21	
r22	
r23	
r24	
r25	
r26	
r27	
r28	
r29	
r30	Element Pointer (EP)
r31	Link Pointer (LP)

31	0
PC	Program Counter

System register set

31	0
EIPC	Exception/Interrupt PC
EIPSW	Exception/Interrupt PSW

31	0
FEPC	Fatal Error PC
FEPSW	Fatal Error PSW

31	0
ECR	Exception Cause Register

31	0
PSW	Program Status Word

3.2.1 Program register set

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

(1) General-purpose registers

32 general-purpose registers, r0 to r31, are available. Any 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. 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. The contents must be restored to the registers after the registers have been used.

Table 3-1: Program Registers

Name	Usage	Operation
r0	Zero register	Always holds 0
r1	Assembler reserved register	Used as working register for address generation
r2	Interrupt stack pointer	Used as stack pointer for interrupt handler
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[Note]
r6 to r29		Address/data variable registers
r30	Element pointer	Base pointer when memory is accessed
r31	Link pointer	Used by compiler when calling function
PC	Program counter	Holds instruction address during program execution

Note: Area in which program code is mapped.

(2) Program counter (PC)

This register holds the address of the instruction under execution. The lower 24 bits of this register are valid, and bits 31 to 24 are fixed to 0. If a carry occurs from bit 23 to 24, it is ignored.
Bit 0 is fixed to 0, and branching to an odd address cannot be performed.

Figure 3-2: Program Counter (PC)

After reset: 00000000H

Symbol	31	24	23	1	0
PC	Fixed to 0		Instruction address under execution		0

3.2.2 System register set

System registers control the status of the CPU and hold interrupt information.

Table 3-2: System Register Numbers

No.	System Register Name	Usage	Operation
0	EIPC	Interrupt status saving registers	These registers save the PC and PSW when an exception or interrupt occurs. Because only one set of these registers is available, their contents must be saved when multiple interrupts are enabled.
1	EIPSW		
2	FEPC	NMI status saving registers	These registers save PC and PSW when NMI occurs.
3	FEPSW		
4	ECR	Interrupt source register	If exception, maskable interrupt, or NMI occurs, this register will contain information referencing the interrupt source. The high-order 16 bits of this register are called FECC, to which exception code of NMI is set. The low-order 16 bits are called EICC, to which exception code of exception/interrupt is set.
5	PSW	Program status word	A program status word is a collection of flags that indicate program status (instruction execution result) and CPU status.
6 to 31	Reserved		

Remark: To read/write these system registers, specify a system register number indicated by the system register load/store instruction (LDSR or STSR instruction).

(1) Interrupt source register (ECR)

Figure 3-3: Interrupt Source Register (ECR)

After reset: 00000000H

Symbol	31	16	15	0
ECR	FECC		EICC	

FECC	Exception code of NMI (For exception code, refer to **Table 7-1**.)
EICC	Exception code of exception/interrupt

(2) Program status word (PSW)

Figure 3-4: Program Status Word (PSW)

After reset: 00000020H

Symbol	31	8	7	6	5	4	3	2	1	0
PSW	RFU		NP	EP	ID	SAT	CY	OV	S	Z

RFU	Reserved field (fixed to 0).
NP	Indicates that NMI processing is in progress. This flag is set when NMI is accepted, and disables multiple interrupts.
EP	Indicates that trap processing is in progress. This flag is set when trap is generated. Moreover, interrupt requests can be accepted when this bit is sets.
ID	Indicates that accepting external interrupt request is disabled.
SAT	This flag is set if the result of executing saturated operation instruction overflows. If overflow does not occur, value of previous operation is held.
CY	This flag is set if carry or borrow occurs as result of operation. If carry or borrow does not occur, it is reset.
OV	This flag is set if overflow occurs during operation. If overflow does not occur, it is reset.
S	This flag is set if the result of operation is negative. It is reset if the result is positive.
Z	This flag is set if the result of operation is zero. If the result is not zero, it is reset.

3.3 Operation Modes

The V850/DB1 has the following operation modes.

(1) Normal operation mode (single-chip mode)

After the system has been released from the reset status, the pins related to the bus interface are set for port mode, execution branches to the reset entry address of the internal ROM, and instruction processing written in the internal ROM is started. However, external expansion mode that connects external device to external memory area is enabled by setting in the memory expansion mode register (MM) by instruction.

(2) Flash memory programming mode

This mode is provided only in the µPD70F3080 (Flash Version). The internal flash memory is programmable or erasable when the V_{PP} voltage is applied to the V_{PP} pin.

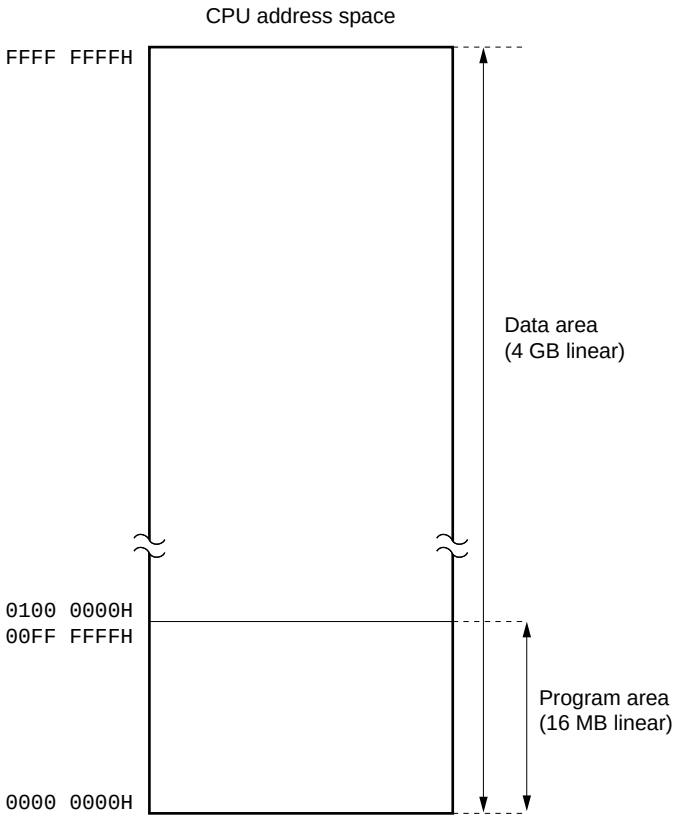
V_{PP}	Operation Mode
0	Normal operation mode
7.8 V	Flash memory programming mode
V_{DD}	Setting prohibited

3.4 Address Space

3.4.1 CPU address space

The CPU of the V850/DB1 is of 32-bit architecture and supports up to 4 GB of linear address space (data space) during operand addressing (data access). When referencing instruction addresses, linear address space (program space) of up to 16 MB is supported.
The CPU address space is shown below.

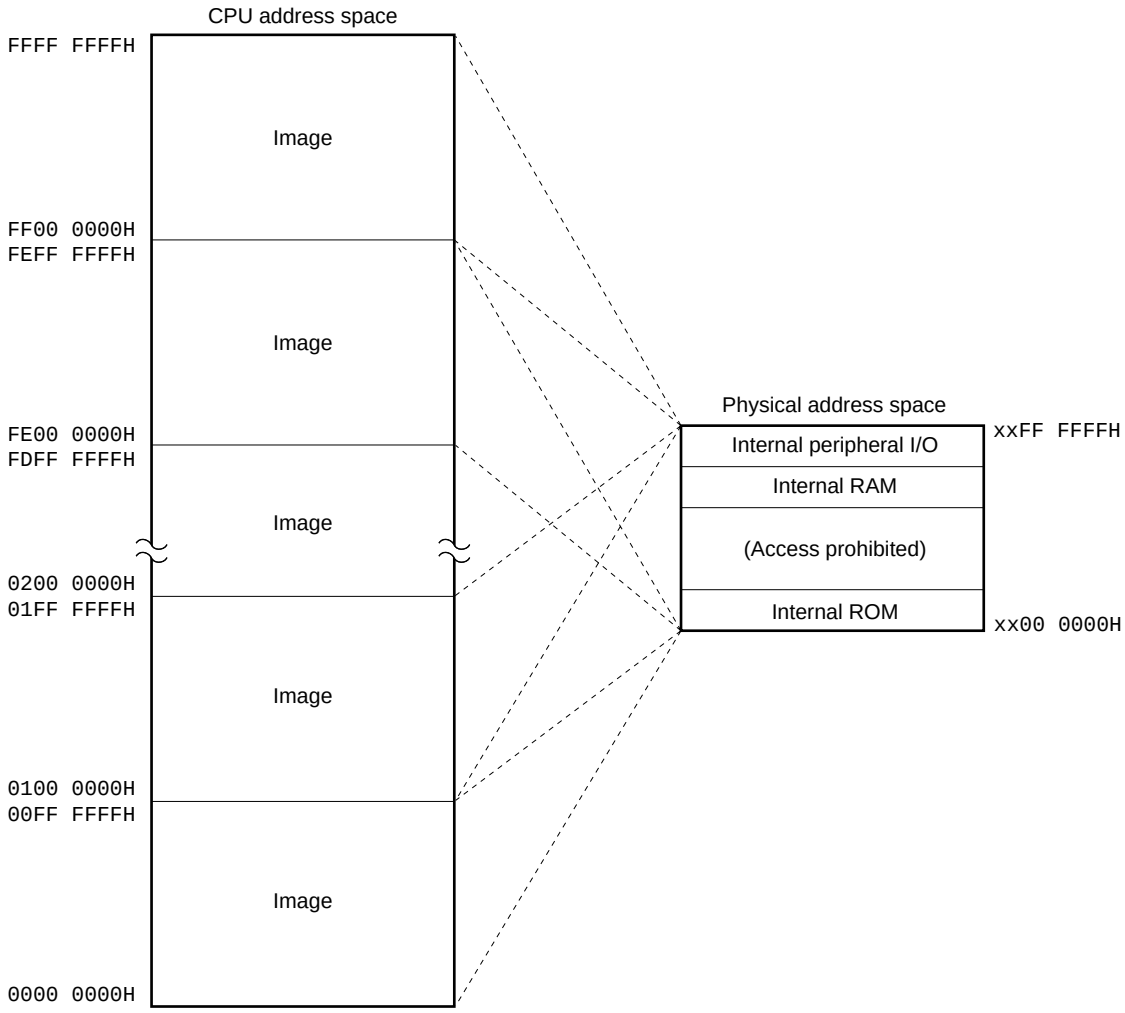
Figure 3-5: CPU Address Space



3.4.2 Image

The core CPU supports 4 GB of “virtual” addressing space, or 256 memory blocks, each containing 16 MB memory locations. In actuality, the same 16 MB block is accessed regardless of the values of bits 31 to 24 of the CPU address. The image of the virtual addressing space is shown below.
Because the higher 8 bits of a 32-bit CPU address are ignored and the CPU address is only seen as a 24-bit external physical address, the physical location xx000000H is equally referenced by multiple address values 00000000H, 01000000H, 02000000H,... FE000000H, FF000000H.

Figure 3-6: Image on Address Space

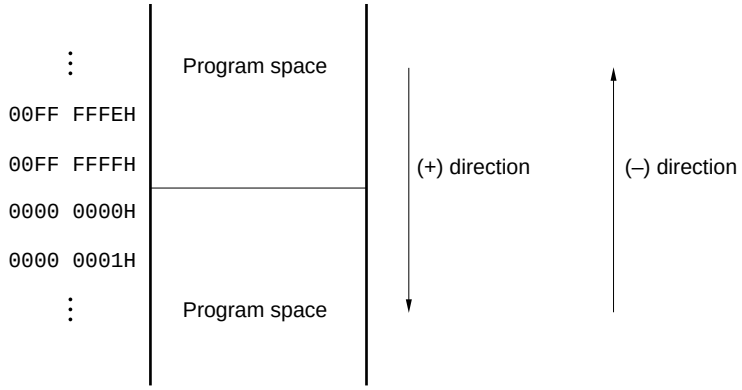


3.4.3 Wrap-around of CPU address space

(1) Program space

Of the 32 bits of the PC (program counter), the higher 8 bits are fixed to 0, and only the lower 24 bits are valid. Even if a carry or borrow occurs from bit 23 to 24 as a result of branch address calculation, the higher 8 bits ignore the carry or borrow and remain 0.
Therefore, the lower-limit address of the program space, address 00000000H, and the upper-limit address 00FFFFFFH are contiguous addresses, and the program space is wrapped around at the boundary of these addresses.

Figure 3-7: Wrap-around Program Space

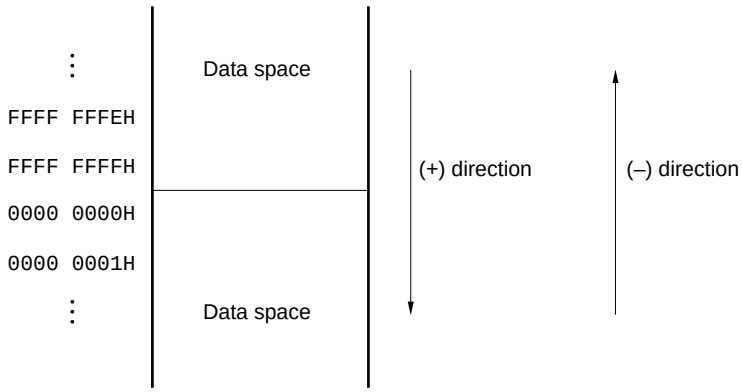


Caution: No instruction can be fetched from the 4 KB area of 00FFF000H to 00FFFFFFH because this area is defined as peripheral I/O area. Therefore, do not execute any branch operation instructions in which the destination address will reside in any part of this area.

(2) Data space

The result of operand address calculation that exceeds 32 bits is ignored.
Therefore, the lower-limit address of the program 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.

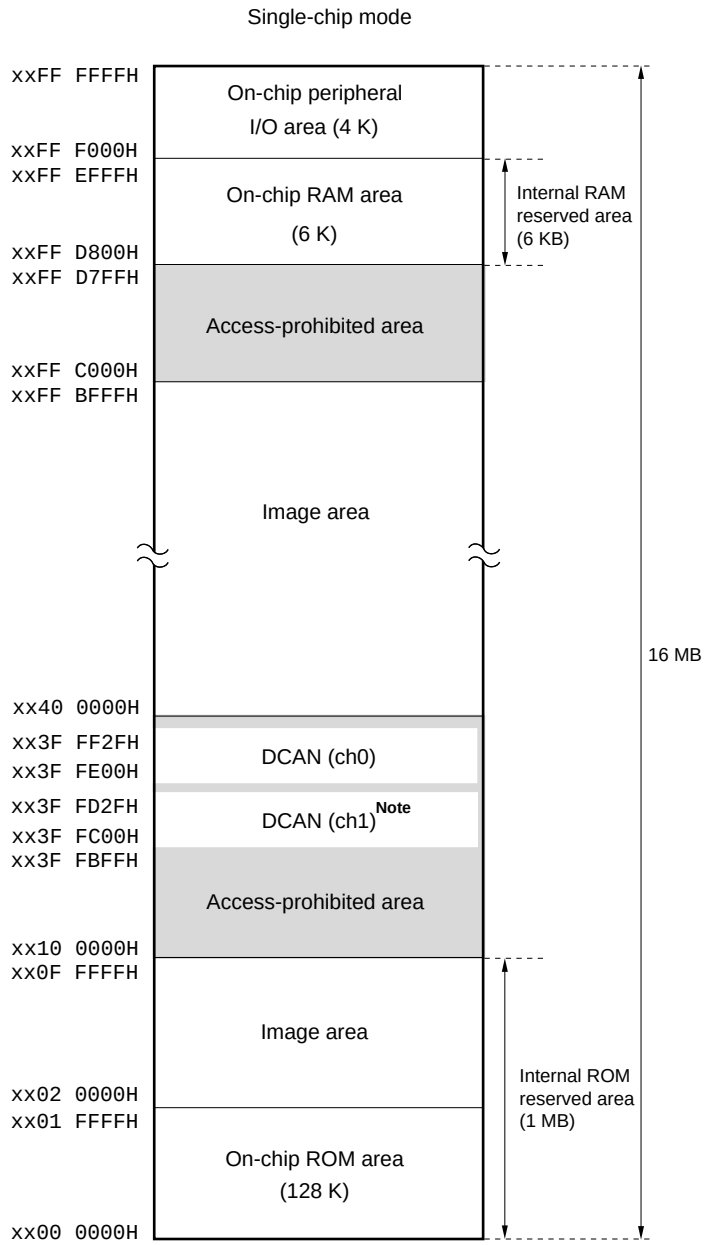
Figure 3-8: Data Space



3.4.4 Memory map

The V850/DB1 reserves areas as shown below.

Figure 3-9: Memory Map



Note: µPD70F3080 (Flash product) only.

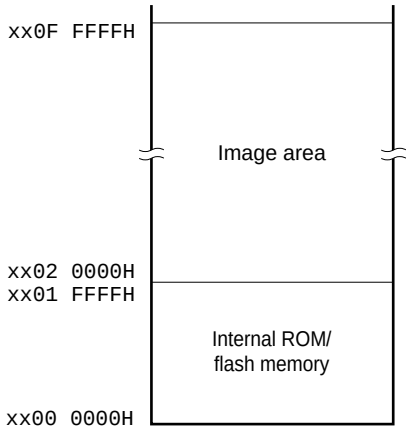
3.4.5 Area

(1) Internal ROM/flash memory area

An area of 1 MB maximum is reserved for the internal ROM/flash memory area.

128 KB are available for the addresses xx000000H to xx01FFFFH.
Addresses xx040000H to xx0FFFFFH is the image area

Figure 3-10: Internal ROM/Flash Memory Area



(2) Interrupt/exception table

The V850/DB1 increases the interrupt response speed by assigning handler addresses corresponding to interrupts/exceptions.

The collection of these handler addresses is called an interrupt/exception table, which is located in the internal ROM area. When an interrupt/exception request is granted, execution jumps to the handler address, and the program written at that memory address is executed. The sources of interrupts/exceptions, and the corresponding addresses are shown below.

Table 3-3: Interrupt/Exception Table (1/2)

Start Address of Interrupt/ Exception Table	Interrupt/Exception Source
0000 0000H	RESET
0000 0010H	NMI
0000 0020H	INTWDT
0000 0040H	TRAP0n (n = 0 to F)
0000 0050H	TRAP1n (n = 0 to F)
0000 0060H	ILGOP
0000 0080H	INTWDTM
0000 0090H	INTAD
0000 00A0H	INTGOVF0
0000 00B0H	INTGOVF1
0000 00C0H	INTGCC0
0000 00D0H	INTGCC1
0000 00E0H	INTGCC2
0000 00F0H	INTGCC3
0000 0100H	INTGCC4
0000 0110H	INTGCC5
0000 0120H	INTP0
0000 0130H	INTP1
0000 0140H	INTP2
0000 0150H	INTP3
0000 0160H	INTP4
0000 0170H	INTP5
0000 0180H	INTCE0
0000 0190H	INTCR0
0000 01A0H	INTCT00
0000 01B0H	INTCT10
0000 01C0H	INTCE1**Note**
0000 01D0H	INTCR1**Note**
0000 01E0H	INTCT01**Note**
0000 01F0H	INTCT11**Note**
0000 0200H	INTCSI00
0000 0210H	INTCSI01
0000 0220H	INTCSI02

Table 3-3: Interrupt/Exception Table (2/2)

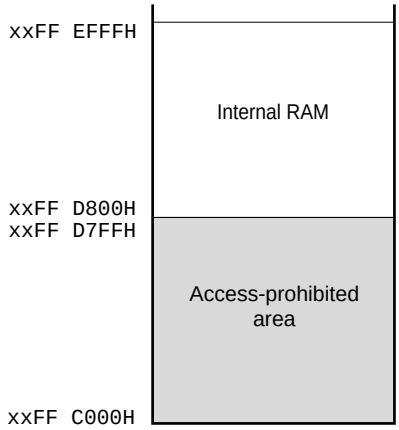
Start Address of Interrupt/ Exception Table	Interrupt/Exception Source
0000 0230H	INTSRE50
0000 0240H	INTSR50
0000 0250H	INTST50
0000 0260H	INTSRE51
0000 0270H	INTSR51
0000 0280H	INTST51
0000 0290H	INTCOVF0
0000 02A0H	INTCCC00
0000 02B0H	INTCCC10
0000 02C0H	INTCOVF1
0000 02D0H	INTCCC01
0000 02E0H	INTCCC11
0000 02F0H	INTTM50
0000 0300H	INTTM51
0000 0310H	INTWTI
0000 0320H	INTWT
0000 0330H	INTBRG

Note: Flash product only. These are Reserved for the Mask product.

(3) Internal RAM area

An area of 6 KB maximum is reserved for the internal RAM area at the addresses xxFFD800H to xxFFEFFFH.
(The addresses direct below this area is an access-prohibited area.)

Figure 3-11: Internal RAM Area

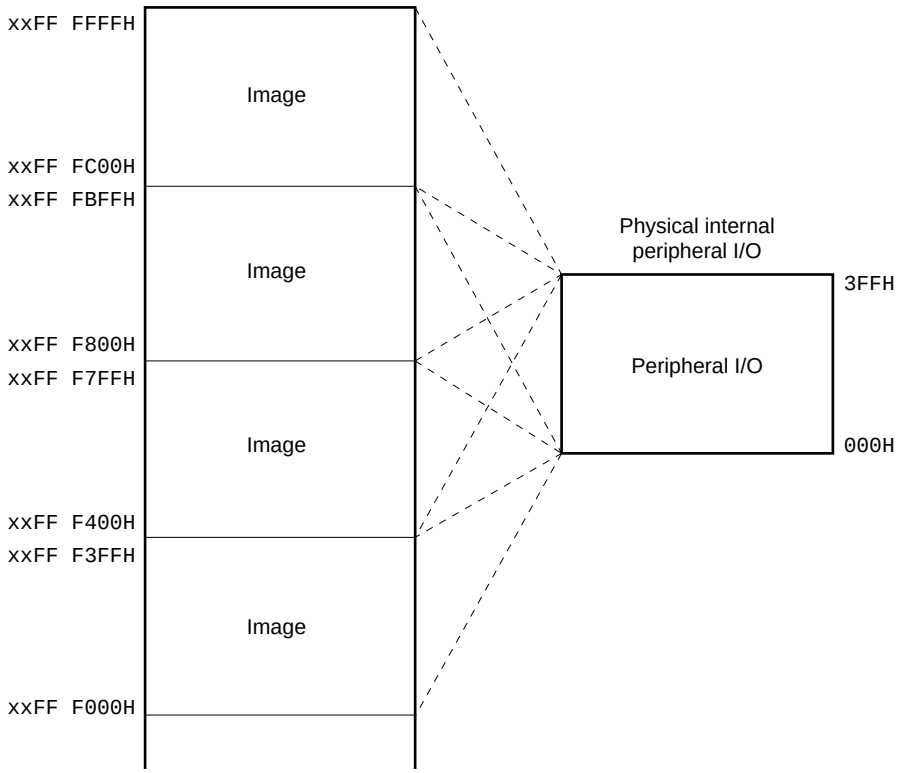


(4) Internal peripheral I/O area

A 4 KB area of addresses FFF000H to FFFFFFH is reserved as an internal peripheral I/O area. The V850/DB1 is provided with a 1 KB area of addresses FFF000H to FFF3FFH as a physical internal peripheral I/O area, and its image can be seen on the rest of the area (FFF400H to FFFFFFH).

Peripheral I/O registers associated with the operation mode specification and the state monitoring for the internal peripherals are all memory-mapped to the internal peripheral I/O area. Program fetches are not allowed in this area.

Figure 3-12: Internal Peripheral I/O Area



Cautions: 1. The least significant bit of an address is not decoded since all registers reside on an even address. If an odd address (2n + 1) in the peripheral I/O area is referenced (accessed in byte units), the register at the next lowest even address (2n) will be accessed.

2. If a register that can be accessed in byte units is accessed in half-word units, the higher 8 bits become undefined, if the access is a read operation. If a write access is made, only the data in the lower 8 bits is written to the register.

3. If a register with n address that can be accessed only in half-word units is accessed in word units, the operation is replaced with two half-word operations. The first operation (lower 16 bits) accesses to the register with n address and the second operation (higher 16 bits) accesses to the register with n + 2 address.

4. If a register with n address that can be accessed in word units is accessed with a word operation, the operation is replaced with two half-word operations. The first operation (lower 16 bits) accesses to the register with n address and the second operation (higher 16 bits) accesses to the register with n + 2 address.

5. 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 Recommended use of address space

The architecture of the V850/DB1 requires that a register that serves as a pointer be secured for address generation in operand data accessing for data space. The address in this pointer register ±32 KB can be accessed directly from instruction. However, general-purpose register used as a pointer register is limited. Therefore, by minimizing the deterioration of address calculation performance when changing the pointer value, the number of usable general-purpose registers for handling variables is maximized, and the program size can be saved because instructions for calculating pointer addresses are not required.
To enhance the efficiency of using the pointer in connection with the memory maps of the V850/DB1, the following points are recommended:

(1) Program space

Of the 32 bits of the PC (program counter), the higher 8 bits are fixed to 0, and only the lower 24 bits are valid. Therefore, a continuous 16 MB space, starting from address 00000000H, unconditionally corresponds to the memory map of the program space.

(2) Data space

For the efficient use of resources to be performed through the wrap-around feature of the data space, the continuous 8 MB address spaces 00000000H to 007FFFFFH and FF800000H to FFFFFFFFH of the 4 GB CPU are used as the data space. With the V850/DB1, 16 MB physical address space is seen as 256 images in the 4 GB CPU address space. The highest bit (bit 23) of this 24-bit address is assigned as address sign-extended to 32 bits.

(a) Application of wrap-around

For example, when R = r0 (zero register) is specified for the LD/ST disp16 [R] instruction, an addressing range of 00000000H ± 32 KB can be referenced with the sign-extended, 16-bit displacement value. Therefore all resources including on-chip hardware can be accessed with one pointer.

The zero register (r0) is a register set to 0 by the hardware, and eliminates the need for additional registers for the pointer.

Figure 3-13: Application of Wrap-Around

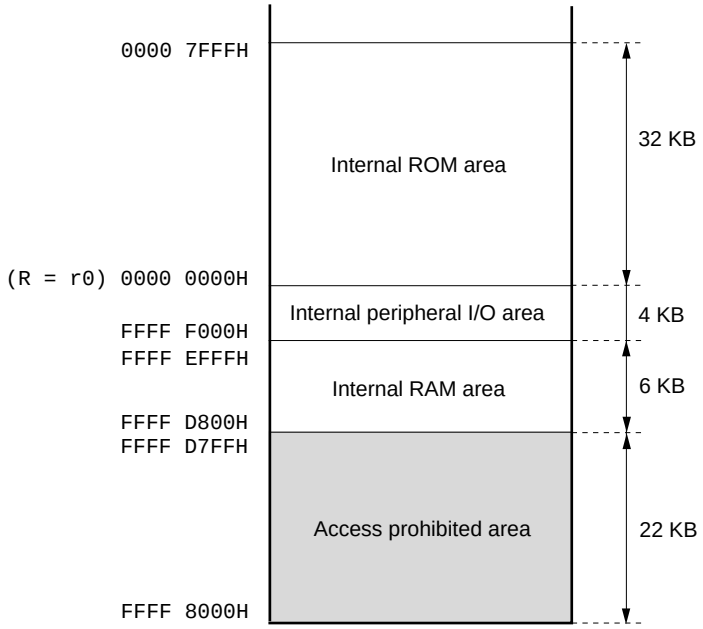
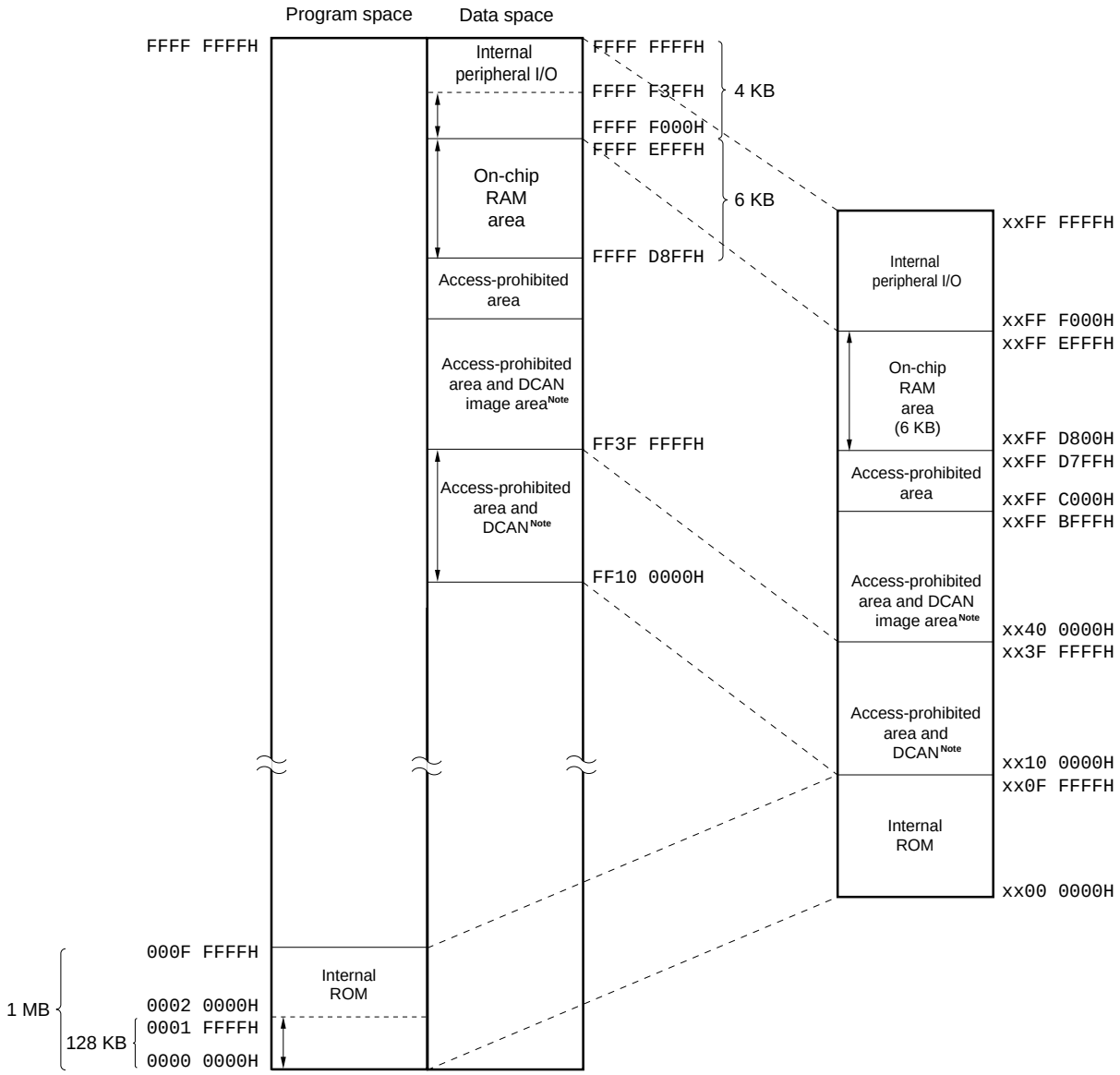


Figure 3-14: Recommended Memory Map (Flash Memory Version)



Note: This area cannot be used as a program area.
But in this adress area, the DCAN message buffers and DCAN SFRs are integrated.

Remarks: 1. The arrows indicate the recommended area.

2. This is as an example the recommended memory map for the µPD70F3080 (6 K RAM).

3.4.7 Peripheral I/O registers

The peripheral I/O register address map is shown below.

Table 3-4: Peripheral I/O Registers (1/7)

Address xxFF F... **Note 1**	Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
000H	Port 0	P0	R/W	×	×			00H
002H	Port 1	P1	R/W	×	×			00H
004H	Port 2	P2	R/W	×	×			00H
006H	Port 3	P3	R/W	×	×			00H
008H	Port 4	P4	R/W	×	×			00H
00AH	Port 5	P5	R/W	×	×			00H
00CH	Port 6	P6	R/W	×	×			00H
00EH	Port 7	P7	R	×	×			undefined
010H	Port 8	P8	R/W	×	×			00H
012H	Port 9	P9	R/W	×	×			00H
014H	Port 10	P10	R/W	×	×			00H
016H	Port 11	P11	R/W	×	×			00H
018H	Port 12	P12	R/W	×	×			00H
01AH	Port 13	P13	R/W	×	×			00H
01CH	Port 14	P14	R/W	×	×			00H
020H	Port 0 mode register	PM0	R/W	×	×			1FH
022H	Port 1 mode register	PM1	R/W	×	×			7FH
024H	Port 2 mode register	PM2	R/W	×	×			FFH
026H	Port 3 mode register	PM3	R/W	×	×			3FH
028H	Port 4 mode register	PM4	R/W	×	×			FFH
02AH	Port 5 mode register	PM5	R/W	×	×			FFH
02CH	Port 6 mode register	PM6	R/W	×	×			3FH
030H	Port 8 mode register	PM8	R/W	×	×			FFH
032H	Port 9 mode register	PM9	R/W	×	×			7FH
034H	Port 10 mode register **Note 2**	PM10	R/W	×	×			FFH
036H	Port 11 mode register **Note 2**	PM11	R/W	×	×			FFH
038H	Port 12 mode register	PM12	R/W	×	×			FFH
03AH	Port 13 mode register	PM13	R/W	×	×			FFH
03CH	Port 14 mode register	PM14	R/W	×	×			0FH
060H	Data Wait Control register	DWC	R/W			×		FFFFH
070H	Power save control register **Note 3**	PSC	R/W	×	×			C0H
074H	Processor clock control register **Note 3**	PCC	R/W	×	×			03H

Notes: 1. Prefix to the shown address. “xx” means “Don’t Care” (24 Bit-addressing)

2. Only performs output control (enables or disables output).

3. This is a special register. See Chapter 3.4.9 “Specific registers” on page 106.

Table 3-4: Peripheral I/O Registers (2/7)

Address xxFF F... **Note 1**	Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
078H	System status register	SYS	R/W	×	×			00H
080H	Pull-up resistor option register 0	PU0	R/W	×	×			00H
082H	Pull-up resistor option register 1	PU1	R/W	×	×			00H
084H	Pull-up resistor option register 2	PU2	R/W	×	×			00H
086H	Pull-up resistor option register 3	PU3	R/W	×	×			00H
090H	Pull-up resistor option register 8	PU8	R/W	×	×			00H
0C0H	Rising edge specification register 0	EGP0	R/W	×	×			00H
0C2H	Falling edge specification register 0	EGN0	R/W	×	×			00H
0E8H	Port 4 mode control register	PMC4	R/W	×	×			00H
0EAH	Port 5 mode control register	PMC5	R/W	×	×			00H
0ECH	Port 6 mode control register	PMC6	R/W	×	×			00H
0F2H	Port 9 mode control register	PMC9	R/W	×	×			00H
0F4H	Port 10 mode control register	PMC10	R/W	×	×			00H
0F6H	Port 11 mode control register	PMC11	R/W	×	×			00H
0F8H	Port 12 mode control register	PMC12	R/W	×	×			00H
0FAH	Port 13 mode control register	PMC13	R/W	×	×			00H
0FCH	Port 14 mode control register	PMC14	R/W	×	×			00H
100H	Interrupt control register	WDTIC	R/W	×	×			47H
102H	Interrupt control register	ADIC	R/W	×	×			47H
104H	Interrupt control register	GOVIC0	R/W	×	×			47H
106H	Interrupt control register	GOVIC1	R/W	×	×			47H
108H	Interrupt control register	GCCIC0	R/W	×	×			47H
10AH	Interrupt control register	GCCIC1	R/W	×	×			47H
10CH	Interrupt control register	GCCIC2	R/W	×	×			47H
10EH	Interrupt control register	GCCIC3	R/W	×	×			47H
110H	Interrupt control register	GCCIC4	R/W	×	×			47H
112H	Interrupt control register	GCCIC5	R/W	×	×			47H
114H	Interrupt control register	PIC0	R/W	×	×			47H
116H	Interrupt control register	PIC1	R/W	×	×			47H
118H	Interrupt control register	PIC2	R/W	×	×			47H
11AH	Interrupt control register	PIC3	R/W	×	×			47H
11CH	Interrupt control register	PIC4	R/W	×	×			47H
11EH	Interrupt control register	PIC5	R/W	×	×			47H
120H	Interrupt control register	CEIC0	R/W	×	×			47H
122H	Interrupt control register	CRIC0	R/W	×	×			47H
124H	Interrupt control register	CTIC00	R/W	×	×			47H
126H	Interrupt control register	CTIC10	R/W	×	×			47H

Notes: 1. Prefix to the shown address. “xx” means “Don’t Care” (24 Bit-addressing)

2. Only performs output control (enables or disables output).

3. This is a special register. See Chapter 3.4.9 “Specific registers” on page 106.

Table 3-4: Peripheral I/O Registers (3/7)

Address	Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
xxFF F... **Note 1**				1-bit	8-bit	16-bit	32-bit	
128H	Interrupt control register	CEIC1	R/W	×	×			47H
12AH	Interrupt control register	CRIC1	R/W	×	×			47H
12CH	Interrupt control register	CTIC01	R/W	×	×			47H
12EH	Interrupt control register	CTIC11	R/W	×	×			47H
130H	Interrupt control register	CSIC00	R/W	×	×			47H
132H	Interrupt control register	CSIC01	R/W	×	×			47H
134H	Interrupt control register	CSIC02	R/W	×	×			47H
136H	Interrupt control register	SREIC50	R/W	×	×			47H
138H	Interrupt control register	SRIC50	R/W	×	×			47H
13AH	Interrupt control register	STIC50	R/W	×	×			47H
13CH	Interrupt control register	SREIC51	R/W	×	×			47H
13EH	Interrupt control register	SRIC51	R/W	×	×			47H
140H	Interrupt control register	STIC51	R/W	×	×			47H
142H	Interrupt control register	COVIC0	R/W	×	×			47H
144H	Interrupt control register	CCCIC00	R/W	×	×			47H
146H	Interrupt control register	CCCIC10	R/W	×	×			47H
148H	Interrupt control register	COVIC1	R/W	×	×			47H
14AH	Interrupt control register	CCCIC01	R/W	×	×			47H
14CH	Interrupt control register	CCCIC11	R/W	×	×			47H
14EH	Interrupt control register	TMIC50	R/W	×	×			47H
150H	Interrupt control register	TMIC51	R/W	×	×			47H
152H	Interrupt control register	WTIIC0	R/W	×	×			47H
154H	Interrupt control register	WTIC0	R/W	×	×			47H
156H	Interrupt control register	BRGIC	R/W	×	×			47H
166H	In-service priority register	ISPR	R	×	×			00H
170H	Command register	PRCMD	W		×			Undefined
180H	Timer mode control register 0	MCNTC0	R/W	×	×			00H
184H	Compare register 10	MCMP10	R/W		×			00H
186H	Compare register 11	MCMP11	R/W		×			00H
188H	Compare register 20	MCMP20	R/W		×			00H
18AH	Compare register 21	MCMP21	R/W		×			00H
18CH	Compare register 30	MCMP30	R/W		×			00H
18EH	Compare register 31	MCMP31	R/W		×			00H
190H	Compare register 40	MCMP40	R/W		×			00H
192H	Compare register 41	MCMP41	R/W		×			00H
194H	Compare control register 1	MCMPC1	R/W	×	×			00H
198H	Compare control register 2	MCMPC2	R/W	×	×			00H

Notes: 1. Prefix to the shown address. “xx” means “Don’t Care” (24 Bit-addressing)

2. Only performs output control (enables or disables output).

3. This is a special register. See Chapter 3.4.9 “Specific registers” on page 106.

Table 3-4: Peripheral I/O Registers (4/7)

Address xxFF F... **Note 1**	Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
19CH	Compare control register 3	MCMPC3	R/W	×	×			00H
1A0H	Compare control register 4	MCMPC4	R/W	×	×			00H
1A8H	Timer mode control register 1	MCNTC1	R/W	×	×			00H
1ACH	Compare register 50	MCMP50	R/W		×			00H
1AEH	Compare register 51	MCMP51	R/W		×			00H
1B0H	Compare register 60	MCMP60	R/W		×			00H
1B2H	Compare register 61	MCMP61	R/W		×			00H
1B4H	Compare control register 5	MCMPC5	R/W	×	×			00H
1B8H	Compare control register 6	MCMPC6	R/W	×	×			00H
200H	Control register GL	TMGML	R/W	×	×			00H
202H	Control register GH	TMGMH	R/W	×	×			00H
204H	Channel mode register GL	TMGCML	R/W	×	×			00H
206H	Channel mode register GH	TMGCMH	R/W	×	×			00H
208H	Output control register GL	OCTLGL	R/W	×	×			44H
20AH	Output control register GH	OCTLGH	R/W	×	×			44H
20CH	Status register	TMGST	R	×	×			00H
210H	16-bit timer counter G0	TMG0	R			×		0000H
214H	16-bit timer counter G1	TMG1	R			×		0000H
218H	Capture/compare register G0	GCC0	R/W			×		0000H
21CH	Capture/compare register G1	GCC1	R/W			×		0000H
220H	Capture/compare register G2	GCC2	R/W			×		0000H
224H	Capture/compare register G3	GCC3	R/W			×		0000H
228H	Capture/compare register G4	GCC4	R/W			×		0000H
22CH	Capture/compare register G5	GCC5	R/W			×		0000H
260H	16-bit timer counter C0	TMC0	R			×		0000H
264H	Capture/compare register C00	CCC00	R/W			×		0000H
268H	Capture/compare register C10	CCC10	R/W			×		0000H
26CH	Control register C00	TMCC00	R/W		×			00H
270H	Control register C10	TMCC10	R/W	×	×			20H
272H	Signal edge select register C0	SESC0	R/W	×	×			00H
280H	16-bit timer counter C1	TMC1	R			×		0000H
284H	Capture/compare register C01	CCC01	R/W			×		0000H
288H	Capture/compare register C11	CCC11	R/W			×		0000H
28CH	Control register C01	TMCC01	R/W		×			00H
290H	Control register C11	TMCC11	R/W	×	×			20H
292H	Signal edge select register C1	SESC1	R/W	×	×			00H
2A0H	8-bit counter 50	TM50	R		×			00H

Notes: 1. Prefix to the shown address. “xx” means “Don’t Care” (24 Bit-addressing)

2. Only performs output control (enables or disables output).

3. This is a special register. See Chapter 3.4.9 “Specific registers” on page 106.

Table 3-4: Peripheral I/O Registers (5/7)

Address xxFF F... **Note 1**	Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
2A2H	8-bit counter 51	TM51	R		×			00H
2A4H	Compare register 50	CR50	R/W		×			00H
2A6H	Compare register 51	CR51	R/W		×			00H
2A8H	Timer clock selection register 50	TCL50	R/W	×	×			00H
2AAH	Timer clock selection register 51	TCL51	R/W	×	×			00H
2ACH	8-bit timer mode control register 50	TMC50	R/W	×	×			00H
2AEH	8-bit timer mode control register 51	TMC51	R/W	×	×			00H
2C0H	Serial interface mode register 00	CSIM00	R/W	×	×			00H
2C2H	Serial interface clock selection register 00	CSICK00	R/W	×	×			00H
2C4H	Serial interface reception buffer register 00	SIRB00	R			×		0000H
		SIRBL00	R		×			00H
2C8H	Serial interface transmission buffer register 00	SOTB00	R/W			×		0000H
		SOTBL00	R/W		×			00H
2D0H	Serial interface first stage transmission buffer register 00	SOTBF00	R/W			×		0000H
		SOTBFL00	R/W		×			00H
2D4H	Serial I/O shift register 00	SIO00	R			×		0000H
		SIOL00	R		×			00H
2E0H	Serial interface mode register 01	CSIM01	R/W	×	×			00H
2E2H	Serial interface clock selection register 01	CSICK01	R/W	×	×			00H
2E4H	Serial interface reception buffer register 01	SIRB01	R			×		0000H
		SIRBL01	R		×			00H
2E8H	Serial interface transmission buffer register 01	SOTB01	R/W			×		0000H
		SOTBL01	R/W		×			00H
2F0H	Serial interface first stage transmission buffer register 01	SOTBF01	R/W			×		0000H
		SOTBFL01	R/W		×			00H
2F4H	Serial I/O shift register 01	SIO01	R			×		0000H
		SIOL01	R		×			00H
300H	Serial interface mode register 02	CSIM02	R/W	×	×			00H
302H	Serial interface clock selection register 02	CSICK02	R/W	×	×			00H
304H	Serial interface reception buffer register 02	SIRB02	R			×		0000H
		SIRBL02	R		×			00H
308H	Serial interface transmission buffer register 02	SOTB02	R/W			×		0000H
		SOTBL02	R/W		×			00H

Notes: 1. Prefix to the shown address. “xx” means “Don’t Care” (24 Bit-addressing)

2. Only performs output control (enables or disables output).

3. This is a special register. See Chapter 3.4.9 “Specific registers” on page 106.

Table 3-4: Peripheral I/O Registers (6/7)

Address xxFF F... **Note 1**	Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
310H	Serial interface first stage transmission buffer register 02	SOTBF02	R/W			×		0000H
		SOTBFL02	R/W		×			00H
314H	Serial I/O shift register 02	SIO02	R			×		0000H
		SIOL02	R		×			00H
320H	UART mode register 50	ASIM50	R/W	×	×			01H
324H	Reception buffer register 50	RXB50	R		×			FFH
326H	UART status register 50	ASIS50	R		×			00H
328H	Transmission buffer register 50	TXB50	R/W		×			FFH
32AH	UART transmission status register 50	ASIF50	R		×			00H
32CH	Clock selection register 50	CKSR50	R/W		×			00H
32EH	Baud rate generator control register 50	BRGC50	R/W		×			FFH
340H	UART mode register 51	ASIM51	R/W	×	×			01H
344H	Reception buffer register 51	RXB51	R		×			FFH
346H	UART status register 51	ASIS51	R		×			00H
348H	Transmission buffer register 51	TXB51	R/W		×			FFH
34AH	UART transmission status register 51	ASIF51	R		×			00H
34CH	Clock selection register 51	CKSR51	R/W		×			00H
34EH	Baud rate generator control register 51	BRGC51	R/W		×			FFH
360H	Watch timer mode register	WTM	R/W	×	×			00H
368H	Pre-scalar mode register	PRSM	R/W	×	×			00H
36AH	Pre-scalar compare register	PRSCM	R/W		×			00H
370H	A/D converter mode register	ADM	R/W	×	×			00H
372H	A/D select register	ADS	R/W		×			00H
374H	Power fail mode register	PFM	R/W	×	×			00H
376H	Power fail threshold register	PFT	R/W		×			00H
378H	A/D conversion result register	ADCR	R			×		undefined
380H	Oscillation stabilization time selection register	OSTS	R/W		×			04H
382H	Watchdog timer clock selection register	WDCS	R/W		×			00H
384H	Watchdog timer mode register	WDTM	R/W	×	×			00H
38AH	Clock output mode register	CLOM	R/W	×	×			00H
390H	LCD clock control register	LCDC	R/W		×			00H
392H	LCD display mode control register	LCDM	R/W	×	×			00H

Notes: 1. Prefix to the shown address. “xx” means “Don’t Care” (24 Bit-addressing)

2. Only performs output control (enables or disables output).

3. This is a special register. See Chapter 3.4.9 “Specific registers” on page 106.

Table 3-4: Peripheral I/O Registers (7/7)

Address xxFF F... **Note 1**	Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
3A0H	LCD display data SEG0	SEGREG00	R/W	×	×			00H
3A2H	LCD display data SEG1	SEGREG01	R/W	×	×			00H
3A4H	LCD display data SEG2	SEGREG02	R/W	×	×			00H
3A6H	LCD display data SEG3	SEGREG03	R/W	×	×			00H
3A8H	LCD display data SEG4	SEGREG04	R/W	×	×			00H
3AAH	LCD display data SEG5	SEGREG05	R/W	×	×			00H
3ACH	LCD display data SEG6	SEGREG06	R/W	×	×			00H
3AEH	LCD display data SEG7	SEGREG07	R/W	×	×			00H
3B0H	LCD display data SEG8	SEGREG08	R/W	×	×			00H
3B2H	LCD display data SEG9	SEGREG09	R/W	×	×			00H
3B4H	LCD display data SEG10	SEGREG10	R/W	×	×			00H
3B6H	LCD display data SEG11	SEGREG11	R/W	×	×			00H
3B8H	LCD display data SEG12	SEGREG12	R/W	×	×			00H
3BAH	LCD display data SEG13	SEGREG13	R/W	×	×			00H
3BCH	LCD display data SEG14	SEGREG14	R/W	×	×			00H
3BEH	LCD display data SEG15	SEGREG15	R/W	×	×			00H
3C0H	LCD display data SEG16	SEGREG16	R/W	×	×			00H
3C2H	LCD display data SEG17	SEGREG17	R/W	×	×			00H
3C4H	LCD display data SEG18	SEGREG18	R/W	×	×			00H
3C6H	LCD display data SEG19	SEGREG19	R/W	×	×			00H
3C8H	LCD display data SEG20	SEGREG20	R/W	×	×			00H
3CAH	LCD display data SEG21	SEGREG21	R/W	×	×			00H
3CCH	LCD display data SEG22	SEGREG22	R/W	×	×			00H
3CEH	LCD display data SEG23	SEGREG23	R/W	×	×			00H
3D0H	LCD display data SEG24	SEGREG24	R/W	×	×			00H
3D2H	LCD display data SEG25	SEGREG25	R/W	×	×			00H
3D4H	LCD display data SEG26	SEGREG26	R/W	×	×			00H
3D6H	LCD display data SEG27	SEGREG27	R/W	×	×			00H
3D8H	LCD display data SEG28	SEGREG28	R/W	×	×			00H
3DAH	LCD display data SEG29	SEGREG29	R/W	×	×			00H
3DCH	LCD display data SEG30	SEGREG30	R/W	×	×			00H
3DEH	LCD display data SEG31	SEGREG31	R/W	×	×			00H
3E0H	LCD display data SEG32	SEGREG32	R/W	×	×			00H
3E2H	LCD display data SEG33	SEGREG33	R/W	×	×			00H
3E4H	LCD display data SEG34	SEGREG34	R/W	×	×			00H
3E6H	LCD display data SEG35	SEGREG35	R/W	×	×			00H

Notes: 1. Prefix to the shown address. “xx” means “Don’t Care” (24 Bit-addressing)

2. Only performs output control (enables or disables output).

3. This is a special register. See Chapter 3.4.9 “Specific registers” on page 106.

3.4.8 Non-Peripheral I/O Registers for DCAN1 and DCAN0

The non-peripheral I/O register for the address map of the DCAN1 and DCAN0 Peripheral is shown below.

Table 3-5: Non-Peripheral I/O Registers for DCAN1 Note 2 (1/10)

Address xxnFF.... Note 1	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
C00H	Transmission buffer 01 Note 2		R/W	×	×			Undefined
C01H			R/W	×	×			Undefined
C02H			R/W	×	×			Undefined
C03H			R/W	×	×			Undefined
C04H			R/W	×	×			Undefined
C05H			R/W	×	×			Undefined
C06H			R/W	×	×			Undefined
C07H			R/W	×	×			Undefined
C08H			R/W	×	×			Undefined
C09H			R/W	×	×			Undefined
C0AH			R/W	×	×			Undefined
C0BH			R/W	×	×			Undefined
C0CH			R/W	×	×			Undefined
C0DH			R/W	×	×			Undefined
C0EH			R/W	×	×			Undefined
C0FH			R/W	×	×			Undefined
C10H	Transmission buffer 11 Note 2		R/W	×	×			Undefined
C11H			R/W	×	×			Undefined
C12H			R/W	×	×			Undefined
C13H			R/W	×	×			Undefined
C14H			R/W	×	×			Undefined
C15H			R/W	×	×			Undefined
C16H			R/W	×	×			Undefined
C17H			R/W	×	×			Undefined
C18H			R/W	×	×			Undefined
C19H			R/W	×	×			Undefined
C1AH			R/W	×	×			Undefined
C1BH			R/W	×	×			Undefined
C1CH			R/W	×	×			Undefined
C1DH			R/W	×	×			Undefined
C1EH			R/W	×	×			Undefined
C1FH			R/W	×	×			Undefined

Notes: 1. Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

2. µPD10F3080 (Flash product) only.

Table 3-5: Non-Peripheral I/O Registers for DCAN1 Note 2 (2/10)

Address xxnFF.... Note 1	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
C20H	Reception buffer 01 / Mask buffer 01		R/W	×	×			Undefined
C21H			R/W	×	×			Undefined
C22H			R/W	×	×			Undefined
C23H			R/W	×	×			Undefined
C24H			R/W	×	×			Undefined
C25H			R/W	×	×			Undefined
C26H			R/W	×	×			Undefined
C27H			R/W	×	×			Undefined
C28H			R/W	×	×			Undefined
C29H			R/W	×	×			Undefined
C2AH			R/W	×	×			Undefined
C2BH			R/W	×	×			Undefined
C2CH			R/W	×	×			Undefined
C2DH			R/W	×	×			Undefined
C2EH			R/W	×	×			Undefined
C2FH			R/W	×	×			Undefined
C30H	Reception buffer 11		R/W	×	×			Undefined
C31H			R/W	×	×			Undefined
C32H			R/W	×	×			Undefined
C33H			R/W	×	×			Undefined
C34H			R/W	×	×			Undefined
C35H			R/W	×	×			Undefined
C36H			R/W	×	×			Undefined
C37H			R/W	×	×			Undefined
C38H			R/W	×	×			Undefined
C39H			R/W	×	×			Undefined
C3AH			R/W	×	×			Undefined
C3BH			R/W	×	×			Undefined
C3CH			R/W	×	×			Undefined
C3DH			R/W	×	×			Undefined
C3EH			R/W	×	×			Undefined
C3FH			R/W	×	×			Undefined

Notes: 1. Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

2. µPD10F3080 (Flash product) only.

Table 3-5: Non-Peripheral I/O Registers for DCAN1 Note 2 (3/10)

Address xxnFF.... Note 1	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
C40H	Reception buffer 21 / Mask buffer 11		R/W	×	×			Undefined
C41H			R/W	×	×			Undefined
C42H			R/W	×	×			Undefined
C43H			R/W	×	×			Undefined
C44H			R/W	×	×			Undefined
C45H			R/W	×	×			Undefined
C46H			R/W	×	×			Undefined
C47H			R/W	×	×			Undefined
C48H			R/W	×	×			Undefined
C49H			R/W	×	×			Undefined
C4AH			R/W	×	×			Undefined
C4BH			R/W	×	×			Undefined
C4CH			R/W	×	×			Undefined
C4DH			R/W	×	×			Undefined
C4EH			R/W	×	×			Undefined
C4FH			R/W	×	×			Undefined
C50H	Reception buffer 31		R/W	×	×			Undefined
C51H			R/W	×	×			Undefined
C52H			R/W	×	×			Undefined
C53H			R/W	×	×			Undefined
C54H			R/W	×	×			Undefined
C55H			R/W	×	×			Undefined
C56H			R/W	×	×			Undefined
C57H			R/W	×	×			Undefined
C58H			R/W	×	×			Undefined
C59H			R/W	×	×			Undefined
C5AH			R/W	×	×			Undefined
C5BH			R/W	×	×			Undefined
C5CH			R/W	×	×			Undefined
C5DH			R/W	×	×			Undefined
C5EH			R/W	×	×			Undefined
C5FH			R/W	×	×			Undefined

Notes: 1. Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

2. µPD10F3080 (Flash product) only.

Table 3-5: Non-Peripheral I/O Registers for DCAN1 Note 2 (4/10)

Address xxnFF.... Note 1	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
C60H	Reception buffer 41		R/W	×	×			Undefined
C61H			R/W	×	×			Undefined
C62H			R/W	×	×			Undefined
C63H			R/W	×	×			Undefined
C64H			R/W	×	×			Undefined
C65H			R/W	×	×			Undefined
C66H			R/W	×	×			Undefined
C67H			R/W	×	×			Undefined
C68H			R/W	×	×			Undefined
C69H			R/W	×	×			Undefined
C6AH			R/W	×	×			Undefined
C6BH			R/W	×	×			Undefined
C6CH			R/W	×	×			Undefined
C6DH			R/W	×	×			Undefined
C6EH			R/W	×	×			Undefined
C6FH			R/W	×	×			Undefined
C70H	Reception buffer 51		R/W	×	×			Undefined
C71H			R/W	×	×			Undefined
C72H			R/W	×	×			Undefined
C73H			R/W	×	×			Undefined
C74H			R/W	×	×			Undefined
C75H			R/W	×	×			Undefined
C76H			R/W	×	×			Undefined
C77H			R/W	×	×			Undefined
C78H			R/W	×	×			Undefined
C79H			R/W	×	×			Undefined
C7AH			R/W	×	×			Undefined
C7BH			R/W	×	×			Undefined
C7CH			R/W	×	×			Undefined
C7DH			R/W	×	×			Undefined
C7EH			R/W	×	×			Undefined
C7FH			R/W	×	×			Undefined

Notes: 1. Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

2. µPD10F3080 (Flash product) only.

Table 3-5: Non-Peripheral I/O Registers for DCAN1 Note 2 (5/10)

Address xxnFF.... Note 1	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
C80H	Reception buffer 61		R/W	×	×			Undefined
C81H			R/W	×	×			Undefined
C82H			R/W	×	×			Undefined
C83H			R/W	×	×			Undefined
C84H			R/W	×	×			Undefined
C85H			R/W	×	×			Undefined
C86H			R/W	×	×			Undefined
C87H			R/W	×	×			Undefined
C88H			R/W	×	×			Undefined
C89H			R/W	×	×			Undefined
C8AH			R/W	×	×			Undefined
C8BH			R/W	×	×			Undefined
C8CH			R/W	×	×			Undefined
C8DH			R/W	×	×			Undefined
C8EH			R/W	×	×			Undefined
C8FH			R/W	×	×			Undefined
C90H	Reception buffer 71		R/W	×	×			Undefined
C91H			R/W	×	×			Undefined
C92H			R/W	×	×			Undefined
C93H			R/W	×	×			Undefined
C94H			R/W	×	×			Undefined
C95H			R/W	×	×			Undefined
C96H			R/W	×	×			Undefined
C97H			R/W	×	×			Undefined
C98H			R/W	×	×			Undefined
C99H			R/W	×	×			Undefined
C9AH			R/W	×	×			Undefined
C9BH			R/W	×	×			Undefined
C9CH			R/W	×	×			Undefined
C9DH			R/W	×	×			Undefined
C9EH			R/W	×	×			Undefined
C9FH			R/W	×	×			Undefined

Notes: 1. Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

2. µPD10F3080 (Flash product) only.

Table 3-5: Non-Peripheral I/O Registers for DCAN1 Note 2 (6/10)

Address xxnFF.... Note 1	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
CA0H	Reception buffer 81		R/W	×	×			Undefined
CA1H			R/W	×	×			Undefined
CA2H			R/W	×	×			Undefined
CA3H			R/W	×	×			Undefined
CA4H			R/W	×	×			Undefined
CA5H			R/W	×	×			Undefined
CA6H			R/W	×	×			Undefined
CA7H			R/W	×	×			Undefined
CA8H			R/W	×	×			Undefined
CA9H			R/W	×	×			Undefined
CAAH			R/W	×	×			Undefined
CABH			R/W	×	×			Undefined
CACH			R/W	×	×			Undefined
CADH			R/W	×	×			Undefined
CAEH			R/W	×	×			Undefined
CAFH			R/W	×	×			Undefined
CB0H	Reception buffer 91		R/W	×	×			Undefined
CB1H			R/W	×	×			Undefined
CB2H			R/W	×	×			Undefined
CB3H			R/W	×	×			Undefined
CB4H			R/W	×	×			Undefined
CB5H			R/W	×	×			Undefined
CB6H			R/W	×	×			Undefined
CB7H			R/W	×	×			Undefined
CB8H			R/W	×	×			Undefined
CB9H			R/W	×	×			Undefined
CBAH			R/W	×	×			Undefined
CBBH			R/W	×	×			Undefined
CBCH			R/W	×	×			Undefined
CBDH			R/W	×	×			Undefined
CBEH			R/W	×	×			Undefined
CBFH			R/W	×	×			Undefined

Notes: 1. Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

2. µPD10F3080 (Flash product) only.

Table 3-5: Non-Peripheral I/O Registers for DCAN1 Note 2 (7/10)

Address xxnFF.... Note 1	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
CC0H	Reception buffer 101		R/W	×	×			Undefined
CC1H			R/W	×	×			Undefined
CC2H			R/W	×	×			Undefined
CC3H			R/W	×	×			Undefined
CC4H			R/W	×	×			Undefined
CC5H			R/W	×	×			Undefined
CC6H			R/W	×	×			Undefined
CC7H			R/W	×	×			Undefined
CC8H			R/W	×	×			Undefined
CC9H			R/W	×	×			Undefined
CCAH			R/W	×	×			Undefined
CCBH			R/W	×	×			Undefined
CCCH			R/W	×	×			Undefined
CCDH			R/W	×	×			Undefined
CCEH			R/W	×	×			Undefined
CCFH			R/W	×	×			Undefined
CD0H	Reception buffer 111		R/W	×	×			Undefined
CD1H			R/W	×	×			Undefined
CD2H			R/W	×	×			Undefined
CD3H			R/W	×	×			Undefined
CD4H			R/W	×	×			Undefined
CD5H			R/W	×	×			Undefined
CD6H			R/W	×	×			Undefined
CD7H			R/W	×	×			Undefined
CD8H			R/W	×	×			Undefined
CD9H			R/W	×	×			Undefined
CDAH			R/W	×	×			Undefined
CDBH			R/W	×	×			Undefined
CDCH			R/W	×	×			Undefined
CDDH			R/W	×	×			Undefined
CDEH			R/W	×	×			Undefined
CDFH			R/W	×	×			Undefined

Notes: 1. Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

2. µPD10F3080 (Flash product) only.

Table 3-5: Non-Peripheral I/O Registers for DCAN1 Note 2 (8/10)

Address xxnFF.... Note 1	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
CE0H	Reception buffer 121		R/W	×	×			Undefined
CE1H			R/W	×	×			Undefined
CE2H			R/W	×	×			Undefined
CE3H			R/W	×	×			Undefined
CE4H			R/W	×	×			Undefined
CE5H			R/W	×	×			Undefined
CE6H			R/W	×	×			Undefined
CE7H			R/W	×	×			Undefined
CE8H			R/W	×	×			Undefined
CE9H			R/W	×	×			Undefined
CEAH			R/W	×	×			Undefined
CEBH			R/W	×	×			Undefined
CECH			R/W	×	×			Undefined
CEDH			R/W	×	×			Undefined
CEEH			R/W	×	×			Undefined
CEFH			R/W	×	×			Undefined
CF0H	Reception buffer 131		R/W	×	×			Undefined
CF1H			R/W	×	×			Undefined
CF2H			R/W	×	×			Undefined
CF3H			R/W	×	×			Undefined
CF4H			R/W	×	×			Undefined
CF5H			R/W	×	×			Undefined
CF6H			R/W	×	×			Undefined
CF7H			R/W	×	×			Undefined
CF8H			R/W	×	×			Undefined
CF9H			R/W	×	×			Undefined
CFAH			R/W	×	×			Undefined
CFBH			R/W	×	×			Undefined
CFCH			R/W	×	×			Undefined
CFDH			R/W	×	×			Undefined
CFEH			R/W	×	×			Undefined
CFFH			R/W	×	×			Undefined
D00H	Reception buffer 141		R/W	×	×			Undefined
D01H			R/W	×	×			Undefined
D02H			R/W	×	×			Undefined
D03H			R/W	×	×			Undefined
D04H			R/W	×	×			Undefined

Notes: 1. Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

2. µPD10F3080 (Flash product) only.

Table 3-5: Non-Peripheral I/O Registers for DCAN1 Note 2 (9/10)

Address xxnFF.... Note 1	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
D05H	Reception buffer 141		R/W	×	×			Undefined
D06H			R/W	×	×			Undefined
D07H			R/W	×	×			Undefined
D08H			R/W	×	×			Undefined
D09H			R/W	×	×			Undefined
D0AH			R/W	×	×			Undefined
D0BH			R/W	×	×			Undefined
D0CH			R/W	×	×			Undefined
D0DH			R/W	×	×			Undefined
D0EH			R/W	×	×			Undefined
D0FH			R/W	×	×			Undefined
D10H	Reception buffer 151		R/W	×	×			Undefined
D11H			R/W	×	×			Undefined
D12H			R/W	×	×			Undefined
D13H			R/W	×	×			Undefined
D14H			R/W	×	×			Undefined
D15H			R/W	×	×			Undefined
D16H			R/W	×	×			Undefined
D17H			R/W	×	×			Undefined
D18H			R/W	×	×			Undefined
D19H			R/W	×	×			Undefined
D1AH			R/W	×	×			Undefined
D1BH			R/W	×	×			Undefined
D1CH			R/W	×	×			Undefined
D1DH			R/W	×	×			Undefined
D1EH			R/W	×	×			Undefined
D1FH			R/W	×	×			Undefined
D20H	DCAN control register 1 Note 2	DCANC1	R/W	×	×			00H
D21H	CAN control register	CANC1	R/W	×	×			01H
D22H	Transmission control register 1	TCR1	R/W		×			00H
D23H	Reception message register 1	RMES1	R		×			00H
D24H	Redefinition control register 1	REDEF1	R/W	×	×			00H
D25H	CAN error status register 1	CANES1	R/W		×			00H
D26H	Transmission error counter 1	TEC1	R		×			00H

Notes: 1. Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

2. µPD10F3080 (Flash product) only.

Table 3-5: Non-Peripheral I/O Registers for DCAN1 Note 2 (10/10)

Address xxnFF.... Note 1	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
D27H	Reception error counter 1	REC1	R		×			00H
D28H	Message count register 1	MCNT1	R/W		×			C0H
D29H	Bit rate prescaler 1	BRPRS1	R/W		×			00H
D2AH	Synchronous control register 01	SYNC01	R/W		×			18H
D2BH	Synchronous control register 11	SYNC11	R/W		×			0EH
D2CH	Mask control register 1	MASKC1	R/W		×			00H

Notes: 1. Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

2. µPD10F3080 (Flash product) only.

Table 3-6: Non-Peripheral I/O Registers for DCAN0 (1/10)

Address xxnFF.... **Note 1**	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
E00H	Transmission buffer 00		R/W	×	×			Undefined
E01H			R/W	×	×			Undefined
E02H			R/W	×	×			Undefined
E03H			R/W	×	×			Undefined
E04H			R/W	×	×			Undefined
E05H			R/W	×	×			Undefined
E06H			R/W	×	×			Undefined
E07H			R/W	×	×			Undefined
E08H			R/W	×	×			Undefined
E09H			R/W	×	×			Undefined
E0AH			R/W	×	×			Undefined
E0BH			R/W	×	×			Undefined
E0CH			R/W	×	×			Undefined
E0DH			R/W	×	×			Undefined
E0EH			R/W	×	×			Undefined
E0FH			R/W	×	×			Undefined
E10H	Transmission buffer 10		R/W	×	×			Undefined
E11H			R/W	×	×			Undefined
E12H			R/W	×	×			Undefined
E13H			R/W	×	×			Undefined
E14H			R/W	×	×			Undefined
E15H			R/W	×	×			Undefined
E16H			R/W	×	×			Undefined
E17H			R/W	×	×			Undefined
E18H			R/W	×	×			Undefined
E19H			R/W	×	×			Undefined
E1AH			R/W	×	×			Undefined
E1BH			R/W	×	×			Undefined
E1CH			R/W	×	×			Undefined
E1DH			R/W	×	×			Undefined
E1EH			R/W	×	×			Undefined
E1FH			R/W	×	×			Undefined

Note: Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

Table 3-6: Non-Peripheral I/O Registers for DCAN0 (2/10)

Address xxnFF.... **Note 1**	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
E20H	Reception buffer 00 / Mask buffer 00		R/W	×	×			Undefined
E21H			R/W	×	×			Undefined
E22H			R/W	×	×			Undefined
E23H			R/W	×	×			Undefined
E24H			R/W	×	×			Undefined
E25H			R/W	×	×			Undefined
E26H			R/W	×	×			Undefined
E27H			R/W	×	×			Undefined
E28H			R/W	×	×			Undefined
E29H			R/W	×	×			Undefined
E2AH			R/W	×	×			Undefined
E2BH			R/W	×	×			Undefined
E2CH			R/W	×	×			Undefined
E2DH			R/W	×	×			Undefined
E2EH			R/W	×	×			Undefined
E2FH			R/W	×	×			Undefined
E30H	Reception buffer 10		R/W	×	×			Undefined
E31H			R/W	×	×			Undefined
E32H			R/W	×	×			Undefined
E33H			R/W	×	×			Undefined
E34H			R/W	×	×			Undefined
E35H			R/W	×	×			Undefined
E36H			R/W	×	×			Undefined
E37H			R/W	×	×			Undefined
E38H			R/W	×	×			Undefined
E39H			R/W	×	×			Undefined
E3AH			R/W	×	×			Undefined
E3BH			R/W	×	×			Undefined
E3CH			R/W	×	×			Undefined
E3DH			R/W	×	×			Undefined
E3EH			R/W	×	×			Undefined
E3FH			R/W	×	×			Undefined

Note: Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

Table 3-6: Non-Peripheral I/O Registers for DCAN0 (3/10)

Address xxnFF.... **Note 1**	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
E40H	Reception buffer 20 / Mask buffer 10		R/W	×	×			Undefined
E41H			R/W	×	×			Undefined
E42H			R/W	×	×			Undefined
E43H			R/W	×	×			Undefined
E44H			R/W	×	×			Undefined
E45H			R/W	×	×			Undefined
E46H			R/W	×	×			Undefined
E47H			R/W	×	×			Undefined
E48H			R/W	×	×			Undefined
E49H			R/W	×	×			Undefined
E4AH			R/W	×	×			Undefined
E4BH			R/W	×	×			Undefined
E4CH			R/W	×	×			Undefined
E4DH			R/W	×	×			Undefined
E4EH			R/W	×	×			Undefined
E4FH			R/W	×	×			Undefined
E50H	Reception buffer 30		R/W	×	×			Undefined
E51H			R/W	×	×			Undefined
E52H			R/W	×	×			Undefined
E53H			R/W	×	×			Undefined
E54H			R/W	×	×			Undefined
E55H			R/W	×	×			Undefined
E56H			R/W	×	×			Undefined
E57H			R/W	×	×			Undefined
E58H			R/W	×	×			Undefined
E59H			R/W	×	×			Undefined
E5AH			R/W	×	×			Undefined
E5BH			R/W	×	×			Undefined
E5CH			R/W	×	×			Undefined
E5DH			R/W	×	×			Undefined
E5EH			R/W	×	×			Undefined
E5FH			R/W	×	×			Undefined

Note: Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

Table 3-6: Non-Peripheral I/O Registers for DCAN0 (4/10)

Address xxnFF.... **Note 1**	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
E60H	Reception buffer 40		R/W	×	×			Undefined
E61H			R/W	×	×			Undefined
E62H			R/W	×	×			Undefined
E63H			R/W	×	×			Undefined
E64H			R/W	×	×			Undefined
E65H			R/W	×	×			Undefined
E66H			R/W	×	×			Undefined
E67H			R/W	×	×			Undefined
E68H			R/W	×	×			Undefined
E69H			R/W	×	×			Undefined
E6AH			R/W	×	×			Undefined
E6BH			R/W	×	×			Undefined
E6CH			R/W	×	×			Undefined
E6DH			R/W	×	×			Undefined
E6EH			R/W	×	×			Undefined
E6FH			R/W	×	×			Undefined
E70H	Reception buffer 50		R/W	×	×			Undefined
E71H			R/W	×	×			Undefined
E72H			R/W	×	×			Undefined
E73H			R/W	×	×			Undefined
E74H			R/W	×	×			Undefined
E75H			R/W	×	×			Undefined
E76H			R/W	×	×			Undefined
E77H			R/W	×	×			Undefined
E78H			R/W	×	×			Undefined
E79H			R/W	×	×			Undefined
E7AH			R/W	×	×			Undefined
E7BH			R/W	×	×			Undefined
E7CH			R/W	×	×			Undefined
E7DH			R/W	×	×			Undefined
E7EH			R/W	×	×			Undefined
E7FH			R/W	×	×			Undefined

Note: Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

Table 3-6: Non-Peripheral I/O Registers for DCAN0 (5/10)

Address xxnFF.... **Note 1**	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
E80H	Reception buffer 60		R/W	×	×			Undefined
E81H			R/W	×	×			Undefined
E82H			R/W	×	×			Undefined
E83H			R/W	×	×			Undefined
E84H			R/W	×	×			Undefined
E85H			R/W	×	×			Undefined
E86H			R/W	×	×			Undefined
E87H			R/W	×	×			Undefined
E88H			R/W	×	×			Undefined
E89H			R/W	×	×			Undefined
E8AH			R/W	×	×			Undefined
E8BH			R/W	×	×			Undefined
E8CH			R/W	×	×			Undefined
E8DH			R/W	×	×			Undefined
E8EH			R/W	×	×			Undefined
E8FH			R/W	×	×			Undefined
E90H	Reception buffer 70		R/W	×	×			Undefined
E91H			R/W	×	×			Undefined
E92H			R/W	×	×			Undefined
E93H			R/W	×	×			Undefined
E94H			R/W	×	×			Undefined
E95H			R/W	×	×			Undefined
E96H			R/W	×	×			Undefined
E97H			R/W	×	×			Undefined
E98H			R/W	×	×			Undefined
E99H			R/W	×	×			Undefined
E9AH			R/W	×	×			Undefined
E9BH			R/W	×	×			Undefined
E9CH			R/W	×	×			Undefined
E9DH			R/W	×	×			Undefined
E9EH			R/W	×	×			Undefined
E9FH			R/W	×	×			Undefined

Note: Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

Table 3-6: Non-Peripheral I/O Registers for DCAN0 (6/10)

Address xxnFF.... **Note 1**	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
EA0H	Reception buffer 80		R/W	×	×			Undefined
EA1H			R/W	×	×			Undefined
EA2H			R/W	×	×			Undefined
EA3H			R/W	×	×			Undefined
EA4H			R/W	×	×			Undefined
EA5H			R/W	×	×			Undefined
EA6H			R/W	×	×			Undefined
EA7H			R/W	×	×			Undefined
EA8H			R/W	×	×			Undefined
EA9H			R/W	×	×			Undefined
EAAH			R/W	×	×			Undefined
EABH			R/W	×	×			Undefined
EACH			R/W	×	×			Undefined
EADH			R/W	×	×			Undefined
EAEH			R/W	×	×			Undefined
EAFH			R/W	×	×			Undefined
EB0H	Reception buffer 90		R/W	×	×			Undefined
EB1H			R/W	×	×			Undefined
EB2H			R/W	×	×			Undefined
EB3H			R/W	×	×			Undefined
EB4H			R/W	×	×			Undefined
EB5H			R/W	×	×			Undefined
EB6H			R/W	×	×			Undefined
EB7H			R/W	×	×			Undefined
EB8H			R/W	×	×			Undefined
EB9H			R/W	×	×			Undefined
EBAH			R/W	×	×			Undefined
EBBH			R/W	×	×			Undefined
EBCH			R/W	×	×			Undefined
EBDH			R/W	×	×			Undefined
EBEH			R/W	×	×			Undefined
EBFH			R/W	×	×			Undefined

Note: Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

Table 3-6: Non-Peripheral I/O Registers for DCAN0 (7/10)

Address xxnFF.... **Note 1**	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
EC0H	Reception buffer 100		R/W	×	×			Undefined
EC1H			R/W	×	×			Undefined
EC2H			R/W	×	×			Undefined
EC3H			R/W	×	×			Undefined
EC4H			R/W	×	×			Undefined
EC5H			R/W	×	×			Undefined
EC6H			R/W	×	×			Undefined
EC7H			R/W	×	×			Undefined
EC8H			R/W	×	×			Undefined
EC9H			R/W	×	×			Undefined
ECAH			R/W	×	×			Undefined
ECBH			R/W	×	×			Undefined
ECCH			R/W	×	×			Undefined
ECDH			R/W	×	×			Undefined
ECEH			R/W	×	×			Undefined
ECFH			R/W	×	×			Undefined
ED0H	Reception buffer 110		R/W	×	×			Undefined
ED1H			R/W	×	×			Undefined
ED2H			R/W	×	×			Undefined
ED3H			R/W	×	×			Undefined
ED4H			R/W	×	×			Undefined
ED5H			R/W	×	×			Undefined
ED6H			R/W	×	×			Undefined
ED7H			R/W	×	×			Undefined
ED8H			R/W	×	×			Undefined
ED9H			R/W	×	×			Undefined
EDAH			R/W	×	×			Undefined
EDBH			R/W	×	×			Undefined
EDCH			R/W	×	×			Undefined
EDDH			R/W	×	×			Undefined
EDEH			R/W	×	×			Undefined
EDFH			R/W	×	×			Undefined

Note: Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

Table 3-6: Non-Peripheral I/O Registers for DCAN0 (8/10)

Address xxnFF.... **Note 1**	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
EE0H	Reception buffer 120		R/W	×	×			Undefined
EE1H			R/W	×	×			Undefined
EE2H			R/W	×	×			Undefined
EE3H			R/W	×	×			Undefined
EE4H			R/W	×	×			Undefined
EE5H			R/W	×	×			Undefined
EE6H			R/W	×	×			Undefined
EE7H			R/W	×	×			Undefined
EE8H			R/W	×	×			Undefined
EE9H			R/W	×	×			Undefined
EEAH			R/W	×	×			Undefined
EEBH			R/W	×	×			Undefined
EECH			R/W	×	×			Undefined
EEDH			R/W	×	×			Undefined
EEEH			R/W	×	×			Undefined
EEFH			R/W	×	×			Undefined
EF0H	Reception buffer 130		R/W	×	×			Undefined
EF1H			R/W	×	×			Undefined
EF2H			R/W	×	×			Undefined
EF3H			R/W	×	×			Undefined
EF4H			R/W	×	×			Undefined
EF5H			R/W	×	×			Undefined
EF6H			R/W	×	×			Undefined
EF7H			R/W	×	×			Undefined
EF8H			R/W	×	×			Undefined
EF9H			R/W	×	×			Undefined
EFAH			R/W	×	×			Undefined
EFBH			R/W	×	×			Undefined
EFCH			R/W	×	×			Undefined
EFDH			R/W	×	×			Undefined
EFEH			R/W	×	×			Undefined
EFFH			R/W	×	×			Undefined

Note: Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

Table 3-6: Non-Peripheral I/O Registers for DCAN0 (9/10)

Address xxnFF.... **Note 1**	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
F00H	Reception buffer 140		R/W	×	×			Undefined
F01H			R/W	×	×			Undefined
F02H			R/W	×	×			Undefined
F03H			R/W	×	×			Undefined
F04H			R/W	×	×			Undefined
F05H			R/W	×	×			Undefined
F06H			R/W	×	×			Undefined
F07H			R/W	×	×			Undefined
F08H			R/W	×	×			Undefined
F09H			R/W	×	×			Undefined
F0AH			R/W	×	×			Undefined
F0BH			R/W	×	×			Undefined
F0CH			R/W	×	×			Undefined
F0DH			R/W	×	×			Undefined
F0EH			R/W	×	×			Undefined
F0FH			R/W	×	×			Undefined
F10H	Reception buffer 150		R/W	×	×			Undefined
F11H			R/W	×	×			Undefined
F12H			R/W	×	×			Undefined
F13H			R/W	×	×			Undefined
F14H			R/W	×	×			Undefined
F15H			R/W	×	×			Undefined
F16H			R/W	×	×			Undefined
F17H			R/W	×	×			Undefined
F18H			R/W	×	×			Undefined
F19H			R/W	×	×			Undefined
F1AH			R/W	×	×			Undefined
F1BH			R/W	×	×			Undefined
F1CH			R/W	×	×			Undefined
F1DH			R/W	×	×			Undefined
F1EH			R/W	×	×			Undefined
F1FH			R/W	×	×			Undefined
F20H	DCAN control register 0	DCANC0	R/W	×	×			00H
F21H	CAN control register 0	CANC0	R/W	×	×			01H
F22H	Transmission control register 0	TCR0	R/W		×			00H
F23H	Reception message register 0	RMES0	R		×			00H
F24H	Redefinition control register 0	REDEF0	R/W	×	×			00H
F25H	CAN error status register 0	CANES0	R/W		×			00H
F26H	Transmission error counter 0	TEC0	R		×			00H

Note: Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

Table 3-6: Non-Peripheral I/O Registers for DCAN0 (10/10)

Address xxnFF.... **Note 1**	Non-Peripheral I/O Register Name	Symbol	R/W	Bit Unit Accessibility				After Reset
				1-bit	8-bit	16-bit	32-bit	
F27H	Reception error counter 0	REC0	R		×			00H
F28H	Message count register 0	MCNT0	R/W		×			C0H
F29H	Bit rate pre-scalar 0	BRPRS0	R/W		×			00H
F2AH	Synchronous control register 00	SYNC00	R/W		×			18H
F2BH	Synchronous control register 10	SYNC10	R/W		×			0EH
F2CH	Mask control register 0	MASKC0	R/W		×			00H

Note: Prefix to the shown adress. “n” is set to 3, 7, or B. “xx” means “Don’t Care”

3.4.9 Specific registers

Specific registers are registers that are protected from being written with illegal data due to erroneous program execution, etc. The write access of these specific registers is executed in a specific sequence, and if abnormal store operations occur, it is notified by the system status register (SYS). The V850/DB1 has two specific registers, the power save control register (PSC) and processor clock control register (PCC). For details of the PSC register, refer to **5.4.2 "Power Save Control Register (PSC)" on page 142,** and for details of the PCC register, refer to **5.4.1 "Processor Clock Control Register (PCC)" on page 141.**
The following sequence shows the data setting of the specific registers.

No special sequence is required when reading the specific registers.

Caution: Moreover, to ensure that the execution routine following release of the STOP or WATCH mode is performed correctly, insert the NOP instruction as a dummy instruction (<6>). If the value of the ID bit of PSW does not change as the result of execution of the instruction to return the NP bit to 0 (<5>), insert two NOP instructions, and if the value of the ID bit of PSW changes, insert five NOP instructions.

A description example is given below.

[Description example]: In case of PSC register

```
<1> LDSR rX,5          ;NP bit = 1
<2> ST.B r0,PRCMD [r0] ;Write to PRCMD
<3> ST.B rD,PSC [r0]   ;PSC register setting
<4> LDSR rY,5          ;NP bit = 0
<5> NOP                ;Dummy instruction (2 or 5 instructions)
    .
    .
    .
<6> NOP
(next instruction)     ;Execution routine following cancellation of
                       ;STOP/WATCH mode
    .
    .
    .
```

Remarks: 1. rX: Value to be written to PSW

2. rY: Value to be written back to PSW

3. rD: Value to be set to PSC

When saving the value of PSW, the value of PSW prior to setting the NP bit must be transferred to the rY register.

Caution: The instructions (<5> interrupt disable cancel, <6> NOP instruction) following the store instruction for the PSC register for setting the software STOP mode and WATCH mode are executed before a power save mode is entered.

(1) Command register (PRCMD)

The command register (PRCMD) is a register used when write-accessing the specific register to prevent incorrect writing to the specific registers due to the erroneous program execution.
This register can be written in 8-bit units. It becomes undefined values in a read cycle.
Occurrence of illegal store operations can be checked by the PRERR bit of the SYS register.

Figure 3-15: Command Register (PRCMD)

	7	6	5	4	3	2	1	0	R/W	Address	After reset
PRCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0	W	FFFFF170H	Undefined

REGn	Registration Code
X	Any 8-bit data

(2) System status register (SYS)

This register is allocated with status flags showing the operating state of the entire system. This register can be read/written in 8- or 1-bit units.

Figure 3-16: System Status Register (SYS)

	7	6	5	4	3	2	1	0	R/W	Address	After reset
SYS	0	0	0	PRERR	0	0	0	0	R/W	FFFFF078H	00H

PRERR	Detection of Protection Error
0	Protection error does not occur
1	Protection error occurs

Operation conditions of PRERR flag are shown as follows.

(a) Set conditions (PRERR = 1)

(1) When a write operation to the specific register took place in a state where the store instruction operation for the recent peripheral I/O was not a write operation to the PRCMD register.
(2) When the first store instruction operation following a write operation to the PRCMD register is to any peripheral I/O register apart from specific registers.

(b) Reset conditions: (PRERR = 0)

(1) When 0 is written to the PRERR flag of the SYS register.
(2) At system reset.

[MEMO]

Chapter 4 Interrupt/Exception Processing Function

The V850/DB1 is provided with a dedicated interrupt controller (INTC) for interrupt servicing and can process a total of 46 interrupt requests: 2 non maskable and 44 maskable.
Furthermore a $\overline{\text{RESET}}$-function is supplied, but this is explained in a special chapter.
An interrupt is an event that occurs independently of program execution, and an exception is an event whose occurrence is dependent on program execution. Generally, an exception takes precedence over an interrupt.
The V850/DB1 can process interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).
Eight levels of software-programmable priorities can be specified for each interrupt request. Interrupt servicing starts after no fewer than 5 system clocks (5 x 62.5 ns (@ 16 MHz)) following the generation of an interrupt request.

4.1 Features

- Interrupts
  - RESET interrupts: 2 sources (1 internal from the Watchdog Timer, 1 external from $\overline{\text{RESET}}$-pin)
  - Non-maskable interrupts: 2 sources (1 internal from the Watchdog Timer, 1 external from
  - NMI-pin)
  - Maskable interrupts: 44 sources (37 internal, 6 external from the INTP0 to INTP5-pin)
  - Noise elimination**Note** and valid edge specification for maskable external interrupt request signals.
  - 8 levels of programmable priorities (maskable interrupts)
  - Multiple interrupt control according to priority
  - Masks can be specified for each maskable interrupt request.

Note: An analog filter with a time constant of 500 ns is used for the INTPn pins to suppressed noise.

- Exceptions
  - Software exceptions: 32 sources
  - Exception trap: 1 source (illegal opcode exception)

Interrupt/exception sources are listed in Table 4-1.

Table 4-1: Interrupt/Exception Sources (1/3)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Controlling Register	Generating Source	Generating Unit				
Reset	Interrupt	RESET	-	External reset input from external pin or Internal reset input from WDT	RESET	-	0000H	00000000H	Undefined
Non-maskable	Interrupt	NMI	-	NMI pin input	NMI-pin	-	0010H	00000010H	nextPC
		INTWDT	-	Watchdog timer overflow	WDT	-	0020H	00000020H	nextPC
Software exception	Exception	TRAP0n **Note 1**	-	TRAP instruction	-	-	004nH	00000040H	nextPC
		TRAP1n **Note 1**	-	TRAP instruction	-	-	005nH	00000050H	nextPC
Exception trap	Exception	ILGOP	-	Illegal instruction code	-	-	0060H	00000060H	nextPC
Maskable	Interrupt	INTWDTM	WDTIC	Watchdog timer overflow	WDT	0	0080H	00000080H	nextPC
		INTAD	ADIC	End of A/D conversion	A/D Converter	1	0090H	00000090H	nextPC
		INTGOVF0	GOVIC0	TMG0 overflow	Timer G	2	00A0H	000000A0H	nextPC
		INTGOVF1	GOVIC1	TMG1 overflow	Timer G	3	00B0H	000000B0H	nextPC
		INTGCC0	GCCIC0	TIG0 valid edge detection (capture). Match of TMG0 and GCC0 (compare)	Timer G	4	00C0H	000000C0H	nextPC
		INTGCC1	GCCIC1	TIG1 valid edge detection (capture). Match of TMG0 or TMG1 and GCC1 (compare)	Timer G	5	00D0H	000000D0H	nextPC
		INTGCC2	GCCIC2	TIG2 valid edge detection (capture). Match of TMG0 or TMG1 and GCC2 (compare)	Timer G	6	00E0H	000000E0H	nextPC
		INTGCC3	GCCIC3	TIG3 valid edge detection (capture). Match of TMG0 or TMG1 and GCC3 (compare)	Timer G	7	00F0H	000000F0H	nextPC
		INTGCC4	GCCIC4	TIG4 valid edge detection (capture). Match of TMG0 or TMG1 and GCC4 (compare)	Timer G	8	0100H	00000100H	nextPC
		INTGCC5	GCCIC5	TIG5 valid edge detection (capture). Match of TMG0 and GCC5 (compare)	Timer G	9	0110H	00000110H	nextPC
		INTP0	PIC0	External interrupt pin input edge detection (INTP0)	Pin	10	0120H	00000120H	nextPC
		INTP1	PIC1	External interrupt pin input edge detection (INTP1)	Pin	11	0130H	00000130H	nextPC
		INTP2	PIC2	External interrupt pin input edge detection (INTP2)	Pin	12	0140H	00000140H	nextPC
		INTP3	PIC3	External interrupt pin input edge detection (INTP3)	Pin	13	0150H	00000150H	nextPC
		INTP4	PIC4	External interrupt pin input edge detection (INTP4)	Pin	14	0160H	00000160H	nextPC

Table 4-1: Interrupt/Exception Sources (2/3)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Controlling Register	Generating Source	Generating Unit				
Maskable	Interrupt	INTP5	PIC5	External interrupt pin input edge detection (INTP5)	Pin	15	0170H	00000170H	nextPC
		INTCE0	CEIC0	DCAN1 (ch0) error generation	DCAN	16	0180H	00000180H	nextPC
		INTCR0	CRIC0	Completion of DCAN (ch0) reception	DCAN	17	0190H	00000190H	nextPC
		INTCT00	CTIC00	Completion of DCAN (ch0) transmission (buffer 0)	DCAN	18	01A0H	000001A0H	nextPC
		INTCT10	CTIC10	Completion of DCAN (ch0) transmission (buffer 1)	DCAN	19	01B0H	000001B0H	nextPC
		INTCE1 **Note 2**	CEIC1	DCAN1 error generation	DCAN	20	01C0H	000001C0H	nextPC
		INTCR1 **Note 2**	CRIC1	Completion of DCAN1reception	DCAN	21	01D0H	000001D0H	nextPC
		INTCT01 **Note 2**	CTIC01	Completion of DCAN1 transmission (buffer 0)	DCAN	22	01E0H	000001E0H	nextPC
		INTCT11 **Note 2**	CTIC11	Completion of DCAN1 transmission (buffer 1)	DCAN	23	01F0H	000001F0H	nextPC
		INTCSI00	CSIC00	End of CSI00 transfer	CSI0	24	0200H	00000200H	nextPC
		INTCSI01	CSIC01	End of CSI01 transfer	CSI0	25	0210H	00000210H	nextPC
		INTCSI02	CSIC02	End of CSI02 transfer	CSI0	26	0220H	00000220H	nextPC
		INTSRE50	SREIC50	Generation of UART50 reception error	UART5	27	0230H	00000230H	nextPC
		INTSR50	SRIC50	Completion of UART50 reception	UART5	28	0240H	00000240H	nextPC
		INTST50	STIC50	Completion of UART50 transmission	UART5	29	0250H	00000250H	nextPC
		INTSRE51	SREIC51	Generation of UART51 reception error	UART5	30	0260H	00000260H	nextPC
		INTSR51	SRIC51	Completion of UART51reception	UART5	31	0270H	00000270H	nextPC
		INTST51	STIC51	Completion of UART51 transmission	UART5	32	0280H	00000280H	nextPC
		INTCOVF0	COVIC0	TMC00 overflow	TMC	33	0290H	00000290H	nextPC
		INTCCC00	CCCIC00	TIC00 valid edge detection (capture). Match of TMC0 and CCC00 (compare)	TMC	34	02A0H	000002A0H	nextPC
		INTCCC10	CCCIC10	TIC10 valid edge detection (capture). Match of TMC0 and CCC10 (compare)	TMC	35	02B0H	000002B0H	nextPC
		INTCOVF1	COVIC1	TMC11 overflow	TMC	36	02C0H	000002C0H	nextPC
		INTCCC01	CCCIC01	TIC01 valid edge detection (capture) Match of TMC1 and CCC01 (compare)	TMC	37	02D0H	000002D0H	nextPC
		INTCCC11	CCCIC11	TIC11 valid edge detection (capture) Match of TMC1 and CCC11 (compare)	TMC	38	02E0H	000002E0H	nextPC
		INTTM50	TMIC50	Match of TM50 and CR50	TM5	39	02F0H	000002F0H	nextPC
		INTTM51	TMIC51	Match of TM51 and CR51	TM5	40	0300H	00000300H	nextPC

Table 4-1: Interrupt/Exception Sources (3/3)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Controlling Register	Generating Source	Generating Unit				
Maskable	Interrupt	INTWTI	WTIIC0	Watch timer interval	WT	41	0310H	00000310H	nextPC
		INTWT	WTIC0	Watch timer reference time	WT	42	0320H	00000320H	nextPC
		INTBRG	BRGIC	Match of 8-bit counter and PRSCM	BRG	43	0330H	00000330H	nextPC

Notes: 1. n = 0 to FH

2. µPD70F3080 (Flash product) only.

Remarks: 1. Default priority: The priority order when two or more maskable interrupt requests are generated at the same time. The highest priority is 0.

2. Restored PC: The value of the PC saved to EIPC or FEPC when interrupt/exception processing is started. However, the value of the PC saved when an interrupt is acknowledged during division (DIVH) instruction execution is the value of the PC of the current instruction (DIVH).

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

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

4.2 Non-Maskable Interrupts

A non-maskable interrupt request is acknowledged unconditionally, even when interrupts are in the interrupt disabled (DI) status.
An NMI is not subject to priority control and takes precedence over all the other interrupts.
Non-maskable interrupts of V850/DB1 are available for the following two requests:

- NMI pin input (NMI)
- Non-maskable watchdog timer interrupt request (INTWDT)

When the valid edge specified by bits EGP00, EGN00 of the edge specification registers (EGP0, EGN0) is detected on the NMI pin, the interrupt occurs.
The watchdog timer interrupt request (INTWDT) is only effective as non-maskable interrupt if the Watchdog timer mode 2 of the watchdog timer mode register (WDTM) is set.
If multiple non-maskable interrupts are generated at the same time, the highest priority servicing is executed according to the following priority order (the lower priority interrupt is pending):

NMI > INTWDT

Note that if an NMI or INTWDT request is generated while NMI is being serviced, the service is executed as follows.

(1) If an NMI is generated while NMI is being serviced

The new NMI request is held pending regardless of the value of the PSW.NP bit. The pending NMI request is acknowledged after servicing of the current NMI request has finished (after execution of the RETI instruction).

(2) If an INTWDT request is generated while NMI is being serviced

If the PSW.NP bit remains set (1) while NMI is being serviced, the new INTWDT request is held pending. The pending INTWDT request is acknowledge after servicing of the current NMI request has finished (after execution of the RETI instruction).
If the PSW.NP bit is cleared (0) while NMI is being serviced, the newly generated INTWDT request is executed (NMI servicing is halted).

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

Cautions: 1. Although the values of the PC and PSW are saved to an NMI status save register (FEPC, FEPSW) when a non-maskable interrupt request is generated, only the NMI can be restored by the RETI instruction at this time. Because INTWDT cannot be restored by the RETI instruction, the system must be reset after servicing this interrupt.

2. If PSW.NP is cleared to 0 by the LDSR instruction during non-maskable interrupt servicing, a NMI interrupt afterwards cannot be acknowledged correctly.

4.2.1 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 halfword (FECC) of ECR.
(4) Sets the NP and ID bits of the PSW and clears the EP bit.
(5) Sets the handler address (0000 0010H) corresponding to the non-maskable interrupt to the PC, and transfers control.

The processing configuration of a non-maskable interrupt is shown in Figure 4-1.

Figure 4-1: Processing Configuration of Non-Maskable Interrupt

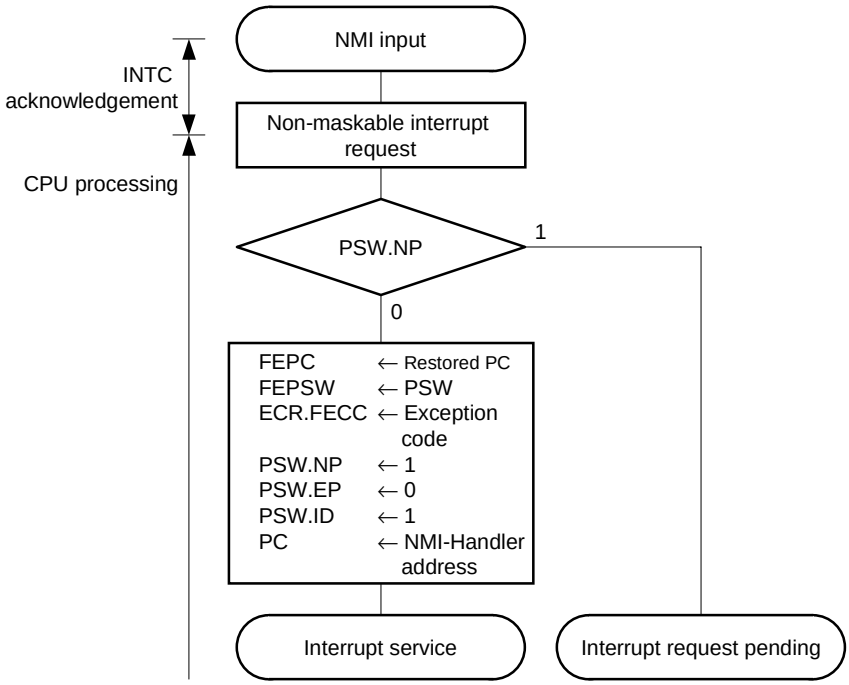
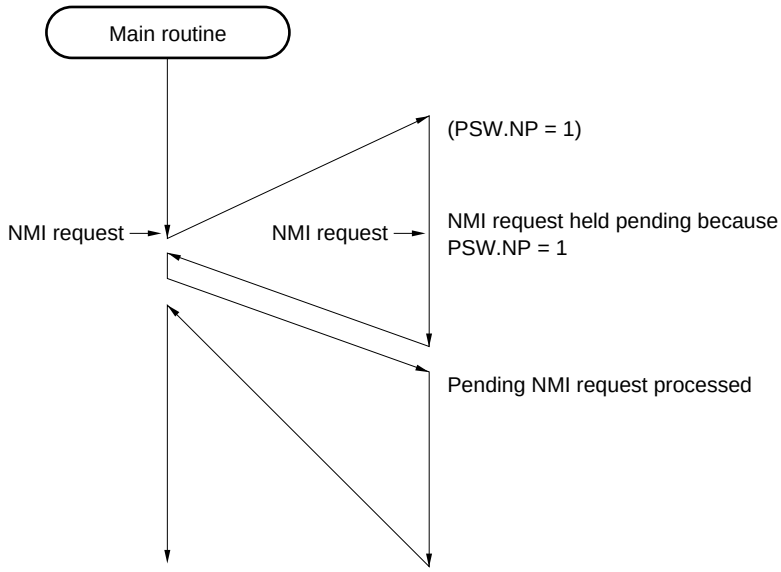
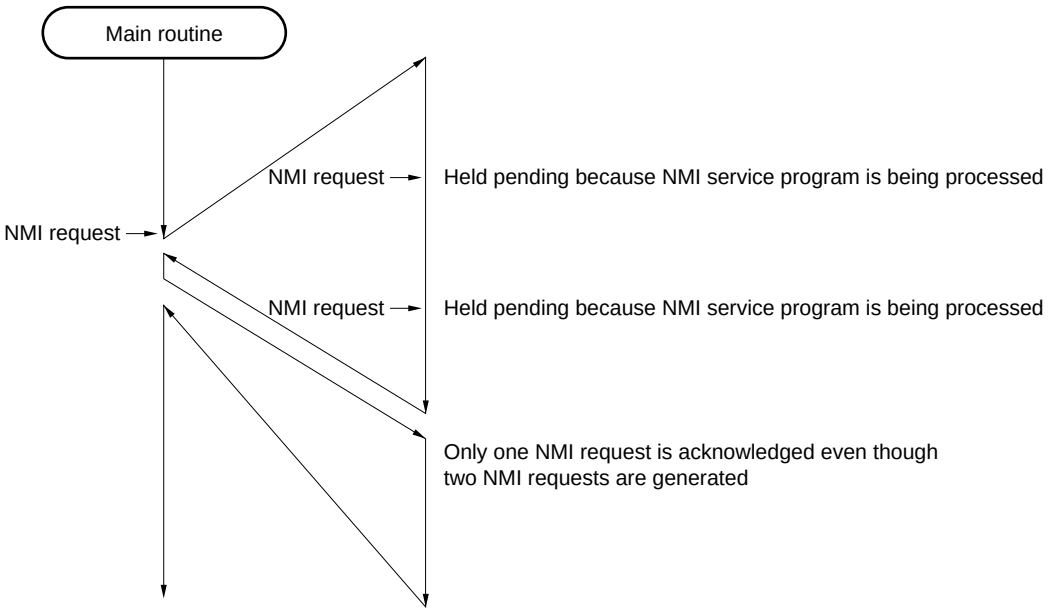


Figure 4-2: Acknowledging Non-Maskable Interrupt Request

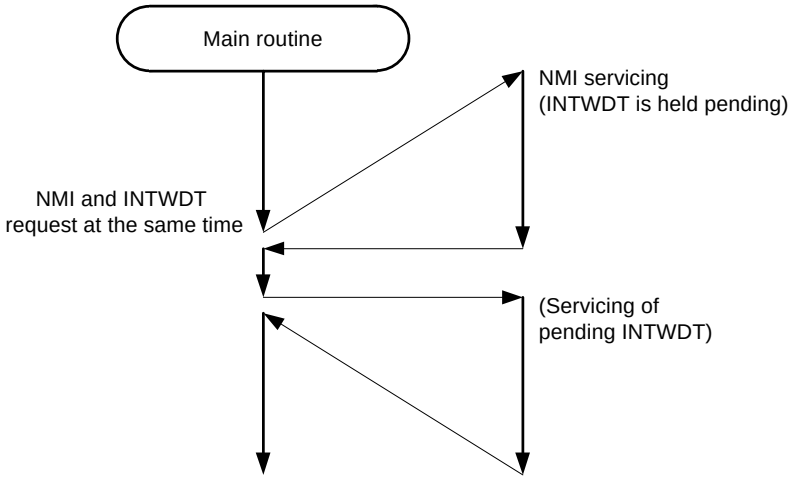
(a) If a new NMI request is generated while an NMI service program is being executed



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



(c) Multiple Non Maskable Interrupts requests generated at the same time


Main routine
NMI servicing
(INTWDT is held pending)
NMI and INTWDT
request at the same time
(Servicing of
pending INTWDT)

4.2.2 Restore

(1) NMI

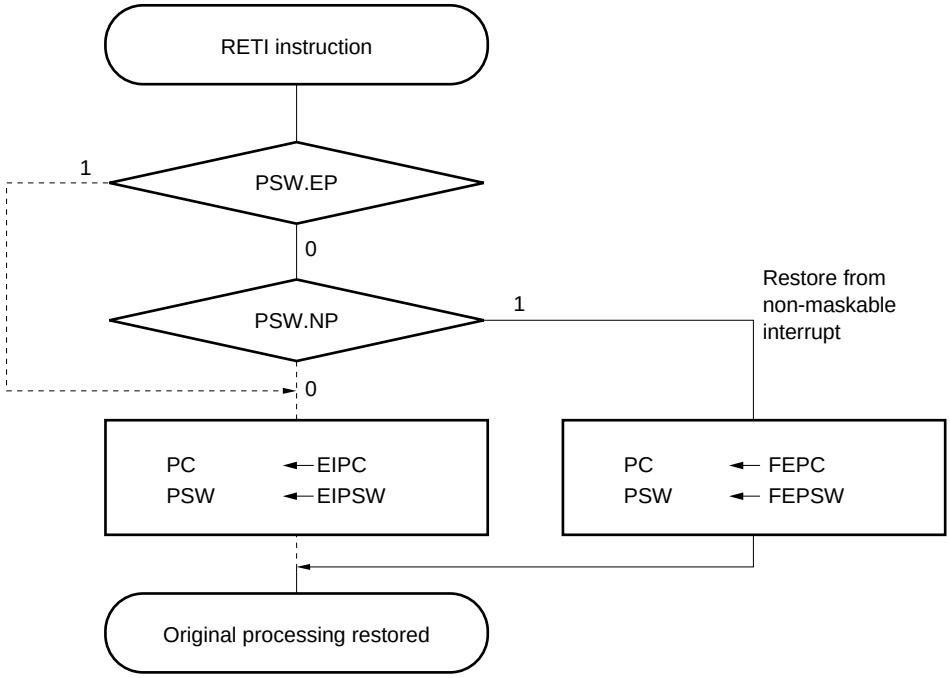
Execution is restored from the non-maskable interrupt (NMI) processing by the RETI instruction. When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<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 4-3 illustrates how the RETI instruction is processed.

Figure 4-3: 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.

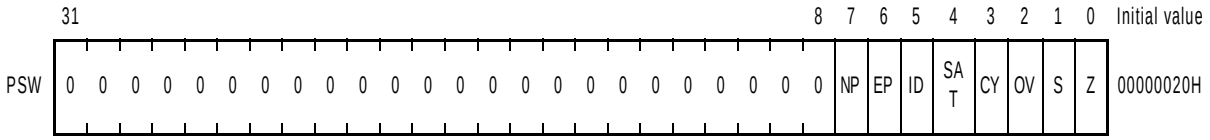
(2) INTWDT

Restoring by RETI instruction is not possible. Perform a system reset after interrupt servicing.

4.2.3 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 an NMI interrupt has been acknowledged, and masks all interrupt requests and exceptions to prohibit multiple interrupts from being acknowledged.

Figure 4-4: Non-maskable Interrupt Status Flag (NP)



Bit Position	Bit Name	Function
7	NP	Indicates whether NMI interrupt processing is in progress. 0: No NMI interrupt processing 1: NMI interrupt currently being processed

4.2.4 Edge detection function of NMI pin

The NMI pin valid edge can be selected from the following four types: falling edge, rising edge, both edges, detects neither rising nor falling edge (see 4.4 “Noise Elimination Circuit” on page 130).
The rising edge specification register 0 (EGP0) and falling edge specification register 0 (EGN0) specify the valid edge of the non-maskable interrupt (NMI). These two registers can be read/written in 1-bit or 8-bit units.
After reset, the valid edge of the NMI pin is set to the “detects neither rising nor falling edge” state. Therefore, the NMI pin functions as a normal port and an interrupt request cannot be acknowledged, unless a valid edge is specified by using the EGP0 and EGN0 registers. When using P00 as an output port, set the NMI valid edge to “detects neither rising nor falling edge”.

4.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The V850/DB1 has 44 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).

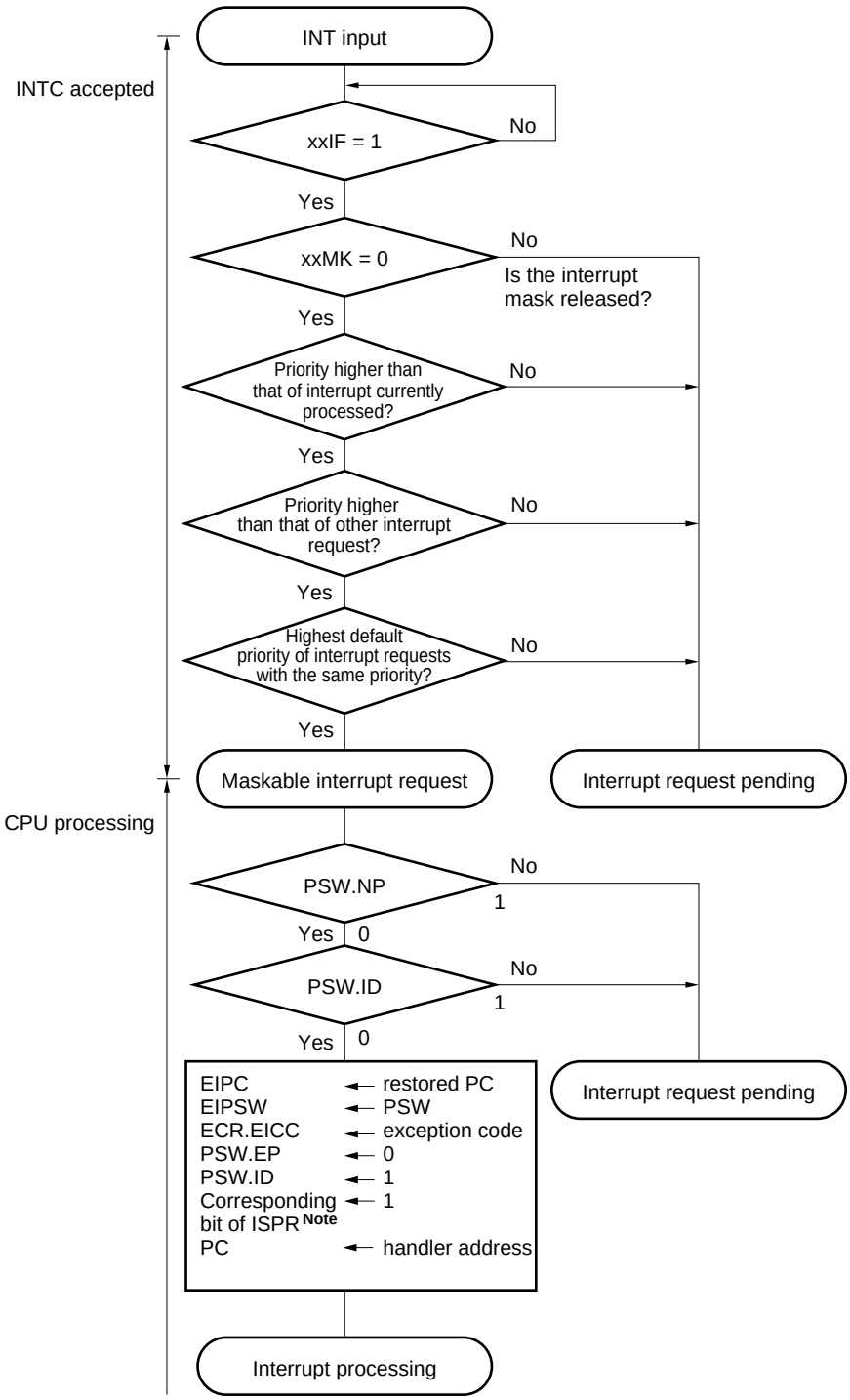
4.3.1 Operation

If a maskable interrupt occurs by INTP0- to INTP5-pin 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 halfword 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 4-5.

Figure 4-5: Maskable Interrupt Processing



Note: For the ISPR register, see Chapter 4.3.5 “In-service priority register (ISPR)” on page 128.

An INTPn input interrupt masked by the interrupt controller and an INTPn input interrupt 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 interrupt starts the new maskable interrupt processing.

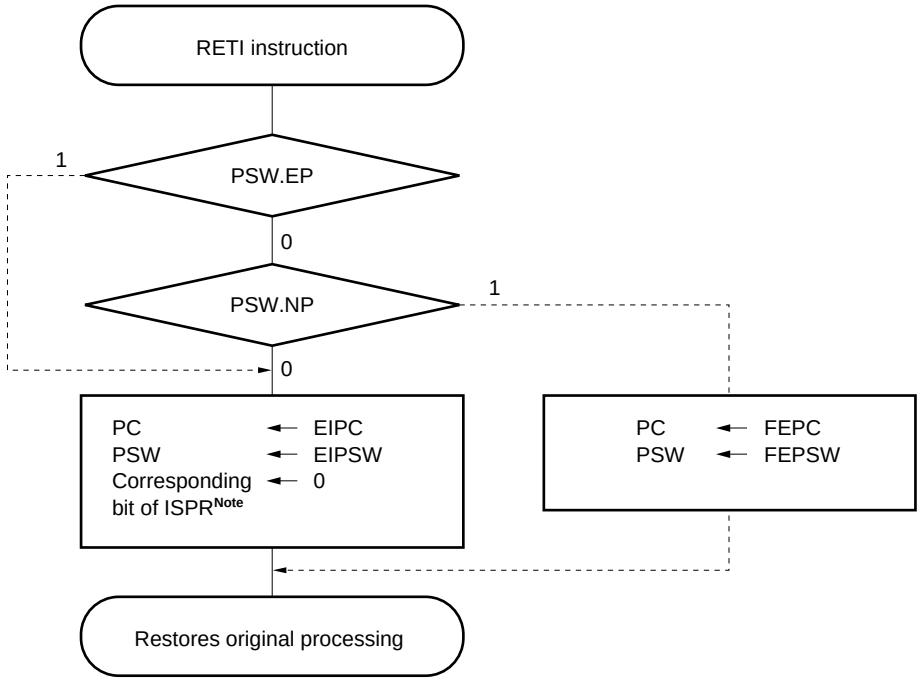
4.3.2 Restore

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 4-6 illustrates the processing of the RETI instruction.

Figure 4-6: RETI Instruction Processing



Note: For the ISPR register, see Chapter 4.3.5 “In-service priority register (ISPR)” on page 128.

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.

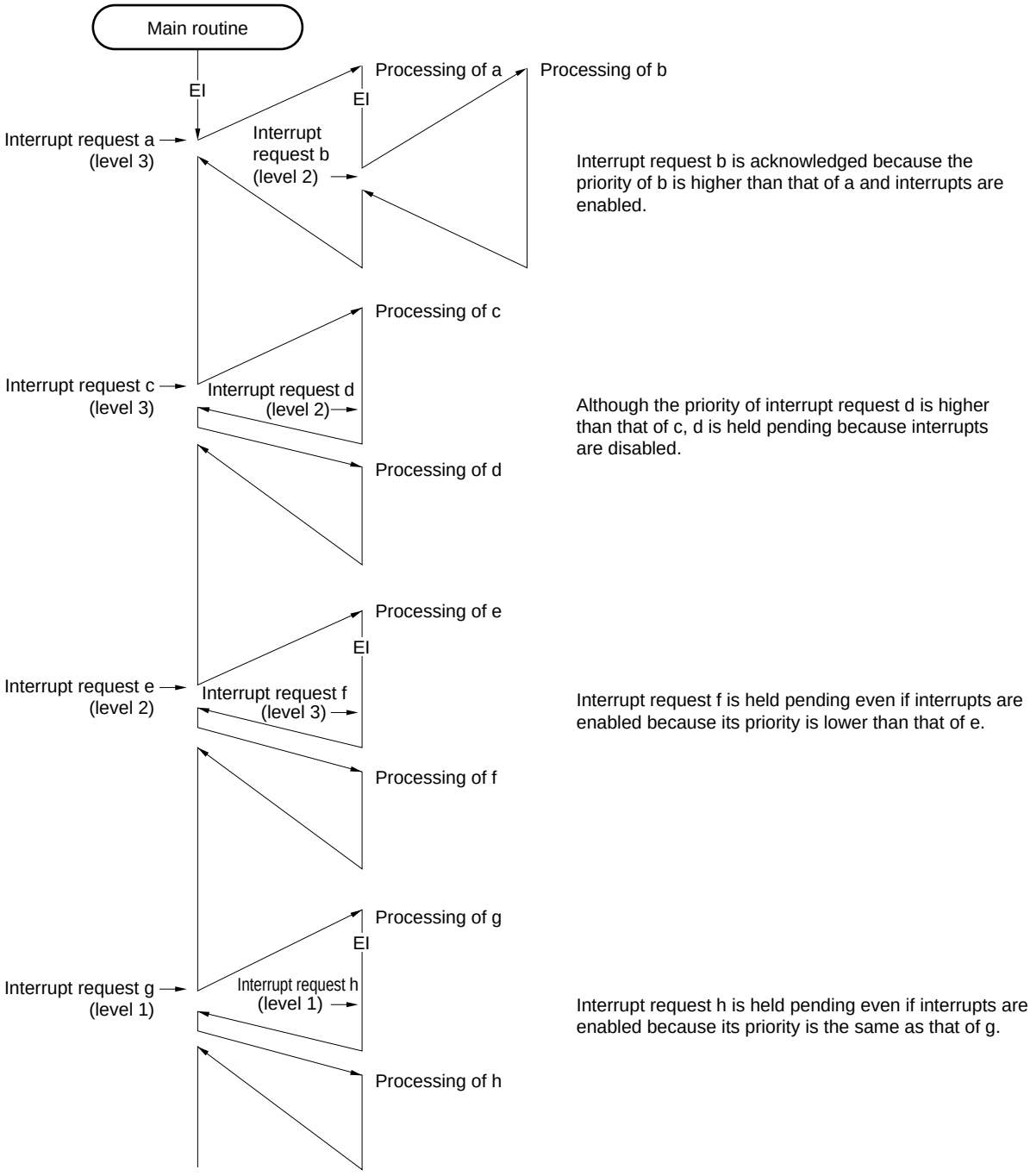
4.3.3 Priorities of maskable interrupts

The V850/DB1 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 (xxPRn) of the interrupt control register (xxICn).

When two or more interrupts having the same priority level specified by the xxPRn 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 4-1, “Interrupt/ Exception Sources,” on page 110. 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.

Figure 4-7: Example of Processing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Processed (1/2)

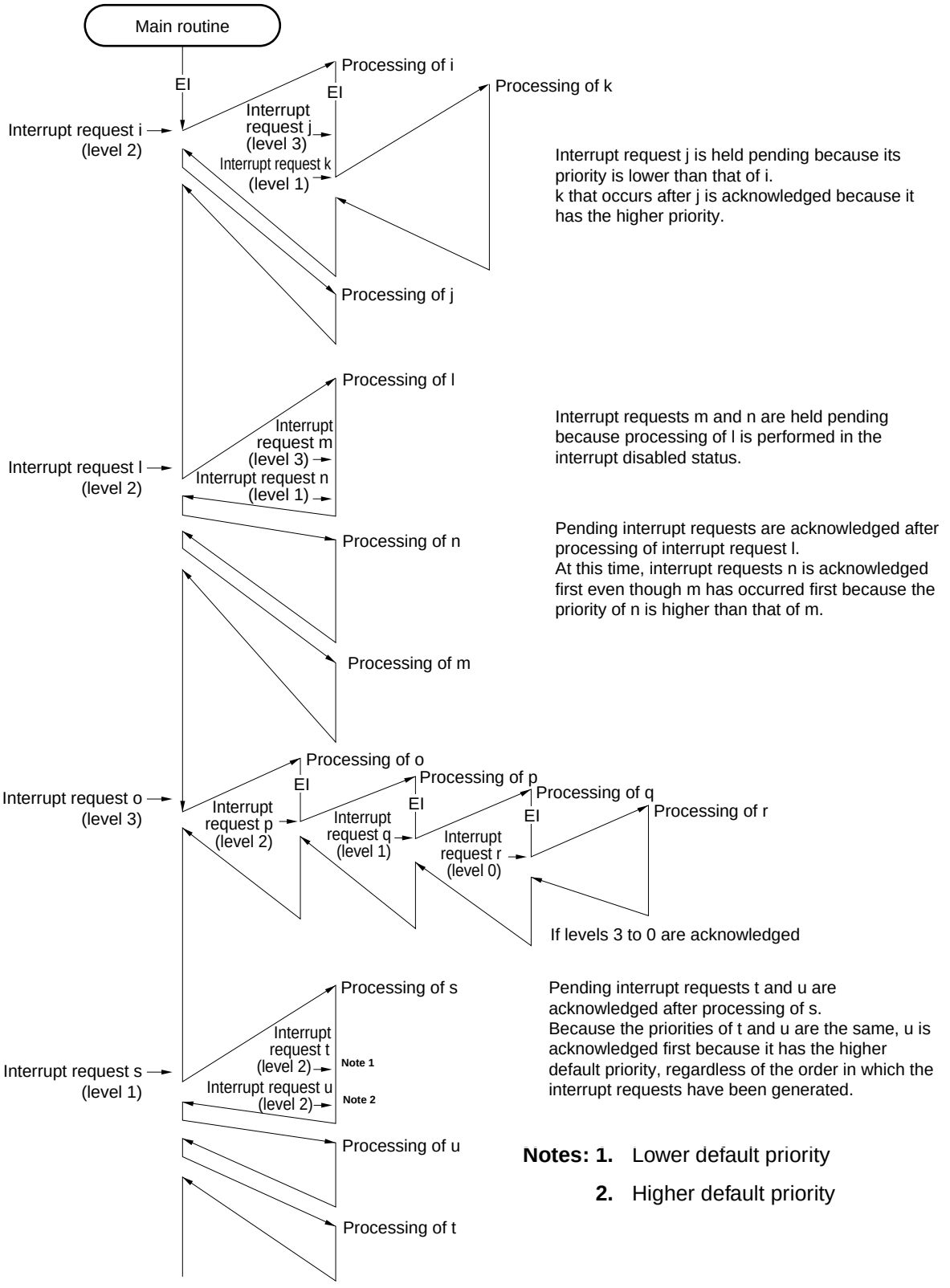


Caution: The values of the EIPC and EIPSW registers must be saved before executing multiple interrupts. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks: 1. a to u in the figure are the temporary names of interrupt requests shown for the sake of explanation.

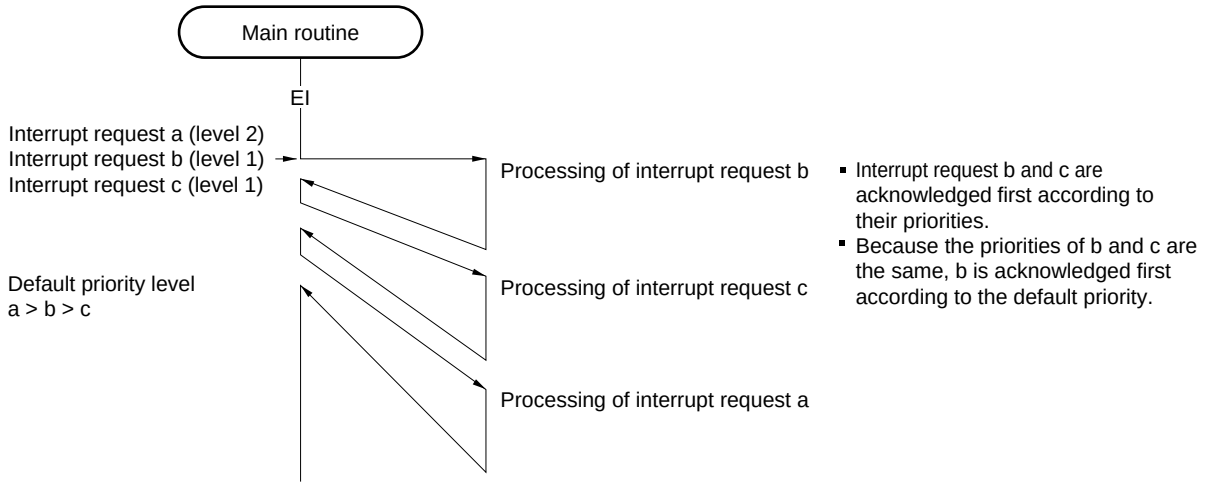
2. The default priority in the figure indicates the relative priority between two interrupt requests.

Figure 4-7: Example of Processing in Which Another Interrupt Request Is Issued While an Interrupt Is Being Processed (2/2)



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.

Figure 4-8: 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.

4.3.4 Interrupt control register (xxICn)

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 4-9: Interrupt Control Register (xxICn)

	7	6	5	4	3	2	1	0	Address	Initial value
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0	FFFF F100H to FFFF F156H	47H

Bit Position	Bit Name	Function
7	xxIFn	This is an interrupt request flag. 0: Interrupt request not issued 1: Interrupt request issued The flag xxIFn is reset automatically by the hardware if an interrupt request is acknowledged.
6	xxMKn	This is an interrupt mask flag. 0: Enables interrupt processing 1: Disables interrupt processing (pending)
2 to 0	xxPRn2 to xxPRn0	8 levels of priority order are specified for each interrupt.

xxPRn2	xxPRn1	xxPRn0	Interrupt Priority Specification Bit
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)

Remarks: 1. xx: Identification name of each peripheral unit (WDT, AD, GOV, GCC, P, CE, CR, CTI, CS, SRE, SR, ST, COV, CCC, TMI, WTI, BRG)

2. n:Peripheral unit number (Refer to **Table 4-2**).

The address and bit of each interrupt control register are shown in the following Table 4-2.

Table 4-2: Address and bit of each Interrupt Control Register (1/2)

Address [FFFF...]	Handler Address	Register	Bit							
			7	6	5	4	3	2	1	0
F100H	00000080H	WDTIC	WDTIF	WDTMK	0	0	0	WDTPR2	WDTPR1	WDTPR0
F102H	00000090H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
F104H	000000A0H	GOVIC0	GOVIF0	GOVMK0	0	0	0	GOVPR02	GOVPR01	GOVPR00
F106H	000000B0H	GOVIC1	GOVIF1	GOVMK1	0	0	0	GOVPR12	GOVPR11	GOVPR10
F108H	000000C0H	GCCIC0	GCCIF0	GCCMK0	0	0	0	GCCPR02	GCCPR01	GCCPR00
F10AH	000000D0H	GCCIC1	GCCIF1	GCCMK1	0	0	0	GCCPR12	GCCPR11	GCCPR10
F10CH	000000E0H	GCCIC2	GCCIF2	GCCMK2	0	0	0	GCCPR22	GCCPR21	GCCPR20
F10EH	000000F0H	GCCIC3	GCCIF3	GCCMK3	0	0	0	GCCPR32	GCCPR31	GCCPR30
F110H	00000100H	GCCIC4	GCCIF4	GCCMK4	0	0	0	GCCPR42	GCCPR41	GCCPR40
F112H	00000110H	GCCIC5	GCCIF5	GCCMK5	0	0	0	GCCPR52	GCCPR51	GCCPR50
F114H	00000120H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
F116H	00000130H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
F118H	00000140H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
F11AH	00000150H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
F11CH	00000160H	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
F11EH	00000170H	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
F120H	00000180H	CEIC0	CEIF0	CEMK0	0	0	0	CEPR02	CEPR01	CEPR00
F122H	00000190H	CRIC0	CRIF0	CRMK0	0	0	0	CRPR02	CRPR01	CRPR00
F124H	000001A0H	CTIC00	CTIF00	CTMK00	0	0	0	CTPR002	CTPR001	CTPR000
F126H	000001B0H	CTIC10	CTIF10	CTMK10	0	0	0	CTPR102	CTPR101	CTPR100
F128H	000001C0H	CEIC1	CEIF1	CEMK1	0	0	0	CEPR12	CEPR11	CEPR10
F12AH	000001D0H	CRIC1	CRIF1	CRMK1	0	0	0	CRPR12	CRPR11	CRPR10
F12CH	000001E0H	CTIC01	CTIF01	CTMK01	0	0	0	CTPR012	CTPR011	CTPR010
F12EH	000001F0H	CTIC11	CTIF11	CTMK11	0	0	0	CTPR112	CTPR111	CTPR110
F130H	00000200H	CSIC00	CSIF00	CSMK00	0	0	0	CSPR002	CSPR001	CSPR000
F132H	00000210H	CSIC01	CSIF01	CSMK01	0	0	0	CSPR012	CSPR011	CSPR010
F134H	00000220H	CSIC02	CSIF02	CSMK02	0	0	0	CSPR022	CSPR021	CSPR020
F136H	00000230H	SREIC50	SREIF50	SREMK50	0	0	0	SREPR502	SREPR501	SREPR500
F138H	00000240H	SRIC50	SRIF50	SRMK50	0	0	0	SRPR502	SRPR501	SRPR500
F13AH	00000250H	STIC50	STIF50	STMK50	0	0	0	STPR502	STPR501	STPR500
F13CH	00000260H	SREIC51	SREIF51	SREMK51	0	0	0	SREPR512	SREPR511	SREPR510
F13EH	00000270H	SRIC51	SRIF51	SRMK51	0	0	0	SRPR512	SRPR511	SRPR510
F140H	00000280H	STIC51	STIF51	STMK51	0	0	0	STPR512	STPR511	STPR510
F142H	00000290H	COVIC0	COVIF0	COVMK0	0	0	0	COVPR02	COVPR01	COVPR00
F144H	000002A0H	CCCIC00	CCCIF00	CCCMK00	0	0	0	CCCPR002	CCCPR001	CCCPR000
F146H	000002B0H	CCCIC10	CCCIF10	CCCMK10	0	0	0	CCCPR102	CCCPR101	CCCPR100
F148H	000002C0H	COVIC1	COVIF1	COVMK1	0	0	0	COVPR12	COVPR11	COVPR10

Table 4-2: Address and bit of each Interrupt Control Register (2/2)

Address [FFFF...]	Handler Address	Register	Bit							
			7	6	5	4	3	2	1	0
F14AH	000002D0H	CCCIC01	CCCIF01	CCCMK01	0	0	0	CCCPR012	CCCPR011	CCCPR010
F14CH	000002E0H	CCCIC11	CCCIF11	CCCMK11	0	0	0	CCCPR112	CCCPR111	CCCPR110
F14EH	000002F0H	TMIC50	TMIF50	TMMK50	0	0	0	TMPR502	TMPR501	TMPR500
F150H	00000300H	TMIC51	TMIF51	TMMK51	0	0	0	TMPR512	TMPR511	TMPR510
F152H	00000310H	WTIIC0	WTIIF0	WTIMK0	0	0	0	WTIPR02	WTIPR01	WTIPR00
F154H	00000320H	WTIC0	WTIF0	WTMK0	0	0	0	WTPR02	WTPR01	WTPR00
F156H	00000330H	BRGIC	BRGIF	BRGMK	0	0	0	BRGPR02	BRGPR01	BRGPR00

4.3.5 In-service priority register (ISPR)

This 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.

This register is read-only in 8-bit or 1-bit units.

Figure 4-10: In-Service Priority Register (ISPR)

	7	6	5	4	3	2	1	0	Address	Initial value
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0	FFFFF166H	00H

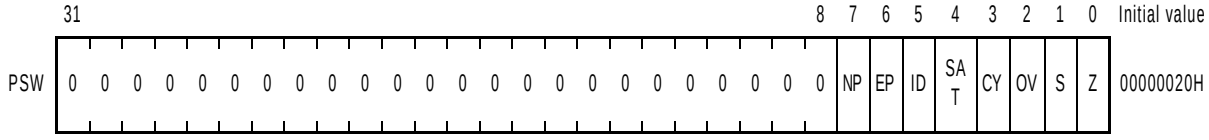
Bit Position	Bit Name	Function
7 to 0	ISPR7 to ISPR0	Indicates priority of interrupt currently acknowledged 0: Interrupt request with priority n not acknowledged 1: Interrupt request with priority n acknowledged

Remark: n = 0 to 7 (priority level)

4.3.6 Maskable interrupt status flag (ID)

The ID flag is bit 5 of the PSW and this controls the maskable interrupt's operating state, and stores control information regarding enabling or disabling of interrupt requests.

Figure 4-11: Maskable Interrupt Status Flag (ID)



Bit Position	Bit Name	Function
5	ID	Indicates whether maskable interrupt processing is enabled or disabled. 0: Maskable interrupt request acknowledgement enabled 1: Maskable interrupt request acknowledgement disabled (pending) This bit 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 to PSW. Non-maskable interrupt requests 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) is acknowledged when the xxIFn bit of xxICn register is set to 1, and the ID flag is reset to 0.

4.4 Noise Elimination Circuit

V850/DB1 is provided with filter/edge detection circuits for the external interrupt pins.

4.4.1 Analog Filter (INTPn pins)

The analogue filter consists of a comparator stage, which compares the input pin level against a delayed input pin level. The filter output follows the filter input, if this compare operation matches.
The delay stage is set to a fixed delay of 500 ns.

4.4.2 Interrupt trigger mode selection with edge detection

The valid edge of the NMI and INTPn pins (n= 0 to 5) can be selected by program. 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
- None detection

The rising edge detect mode register (EGP0) and the falling edge detect mode register (EGN0) is a register that specifies if the rising/falling edge of a signal at the input pins are valid for interrupt output.

(1) Rising Edge Detect Mode Register

This register specifies the valid edge of an external interrupt.
The NMI rising edge can be specified by the EGP00 bit.

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

Figure 4-12: Rising Edge Specification Register (EGP0) Format

	7	6	5	4	3	2	1	0	Address	Initial value
EGP0	0	EGP06	EGP05	EGP04	EGP03	EGP02	EGP01	EGP00	FFFF F0C0H	00H

Bit Position	Bit Name	Function
6 to 1	EGP06 to EGP01	Specifies the valid edge for external maskable interrupt

EGP0n	edge specification register
0	No interrupt request signal occurs at the rising edge
1	Interrupt request signal occurs at the rising edge

Remarks: 1. n = 0: NMI control

2. n = 1 to 6: INTP(n-1) pins control (INTP0 to INTP5)

(2) Falling Edge Detect Mode Register

This register specifies the valid edge of an external interrupt.
The NMI falling edge can be specified by the EGP00 bit.

This register is read-only in 8-bit or 1-bit units.

Figure 4-13: Falling Edge Specification Register (EGN0) Format

	7	6	5	4	3	2	1	0	Address	Initial value
EGN0	0	EGN06	EGN05	EGN04	EGN03	EGN02	EGN01	EGN00	FFFFF0C2H	00H

Bit Position	Bit Name	Function
6 to 1	EGN06 to EGN01	Specifies the valid edge for external maskable interrupt

EGN0n	edge specification register
0	No interrupt request signal occurs at the falling edge
1	Interrupt request signal occurs at the falling edge

Remarks: 1. n = 0: NMI control

2. n = 1 to 6: INTP(n-1) pins control (INTP0 to INTP5)

4.5 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can be always acknowledged.

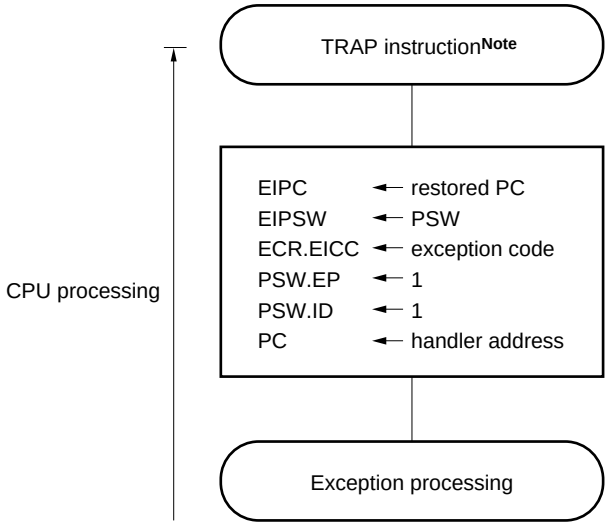
4.5.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine:

(1) Saves the restored 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 the PSW.
(5) Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

Figure 4-14 illustrates the processing of a software exception.

Figure 4-14: 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.

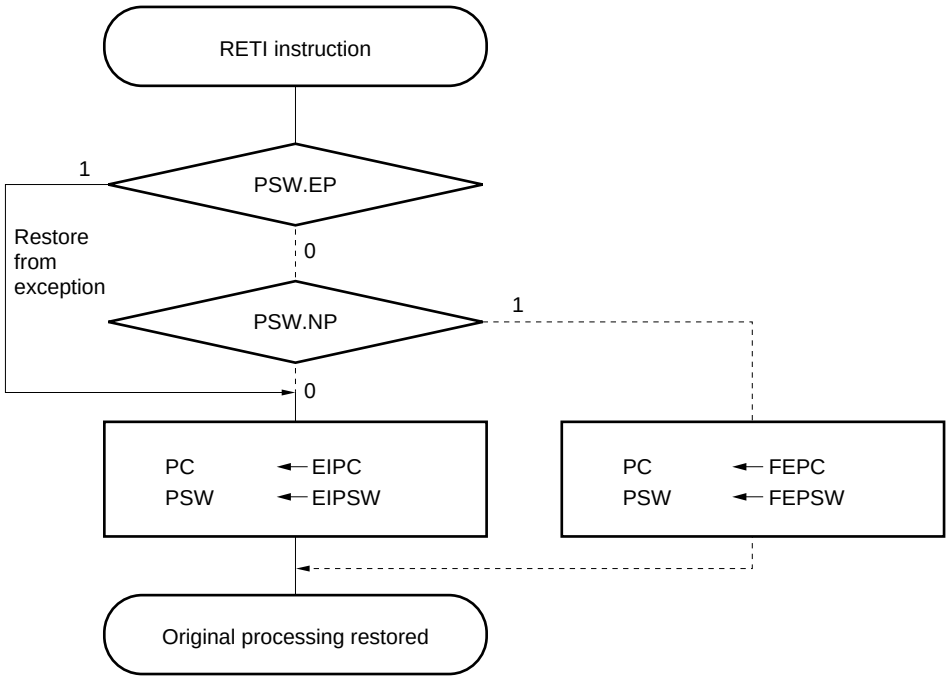
4.5.2 Restore

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 EP bit of the PSW is 1.
(2) Transfers control to the address of the restored PC and PSW.

Figure 4-15 illustrates the processing of the RETI instruction.

Figure 4-15: RETI Instruction Processing



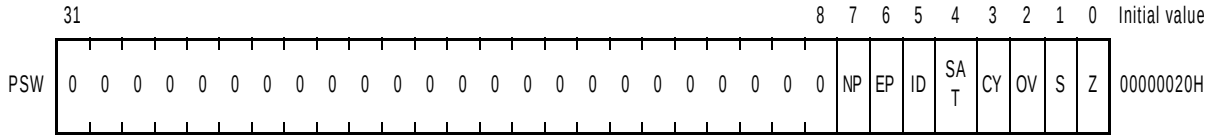
Caution: When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during the software exception processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark: The solid lines show the CPU processing flow.

4.5.3 Exception status flag (EP)

The EP flag is bit 6 of PSW, and is a status flag used to indicate that exception processing is in progress. It is set when an exception occurs.

Figure 4-16: Exception Status Flag (EP)



Bit Position	Bit Name	Function
6	EP	Shows that exception processing is in progress. 0: Exception processing not in progress. 1: Exception processing not in progress.

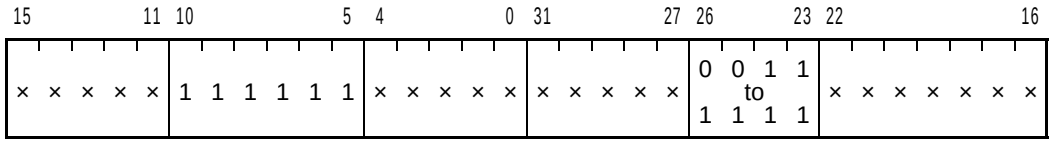
4.6 Exception Trap

An exception trap is an interrupt that is requested when an illegal execution of an instruction takes place. In the V850/DB1, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

4.6.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B and a sub-opcode (bits 26 to 23) of 0011B to 1111B. An exception trap is generated when an instruction applicable to this illegal instruction is executed.

Figure 4-17: Illegal opcode definition



Remark: ×: Arbitrary

Caution: Since it is possible that such an instruction becomes a legal opcode for future devices, it is not recommended to use it for marking of no-opcode.

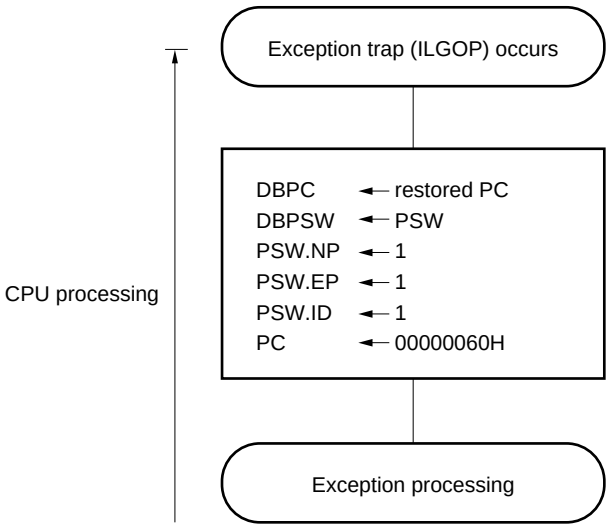
(1) Operation

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 NP, EP, and ID bits of the PSW.
(4) Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

Figure 4-18 illustrates the processing of the exception trap.

Figure 4-18: Exception Trap Processing



4.7 Multiple Interrupt Processing Control

4.7.1 Priorities of interrupts and exceptions

Table 4-3: Priorities of Interrupts and Exceptions

	RESET	NMI	INT	TRAP	ILGOP
RESET		*	*	*	*
NMI	×		←	←	←
INT	×	↑		←	←
TRAP	×	↑	↑		←
ILGOP	×	↑	↑	↑	

RESET: Reset
NMI: Non-maskable interrupt
INT: Maskable interrupt
TRAP: Software exception
ILGOP: Illegal opcode exception
*: The item on the left ignores the item above.
×: The item on the left is ignored by the item above.
↑: The item above is higher than the item on the left in priority.
←: The item on the left is higher than the item above in priority.

4.7.2 Multiple interrupt processing

Multiple interrupt processing control is a process by which an interrupt request that is currently being processed can be interrupted during processing if there is an interrupt request with a higher priority level, and the higher priority interrupt request is received and processed first.
If there is an interrupt request with a lower priority level than the interrupt request currently being processed, that interrupt request is held pending.
Maskable interrupt multiple processing control is executed when an interrupt has an enable status (ID = 0). Thus, if multiple interrupts are executed, it is necessary to have an interrupt enable status (ID = 0) even for an interrupt processing routine.
If a maskable interrupt enable or a software exception is generated in a maskable interrupt or software exception service program, it is necessary to save EIPC and EIPSW.
This is accomplished by the following procedure.

(1) Acknowledgment of maskable interrupts in service program

Service program of maskable interrupt or exception

```
...
...
• EIPC saved to memory or register
• EIPSW saved to memory or register
• EI instruction (interrupt acknowledgment enabled)
...
...                                   ← Maskable interrupt acknowledgment
...
...
• DI instruction (interrupt acknowledgment disabled)
• Saved value restored to EIPSW
• Saved value restored to EIPC
• RETI instruction
```

(2) Generation of exception in service program

Service program of maskable interrupt or exception

```
...
...
• EIPC saved to memory or register
• EIPSW saved to memory or register
...
• TRAP instruction                    ← Exception such as TRAP instruction acknowledged.
...
• Saved value restored to EIPSW
• Saved value restored to EIPC
• RETI instruction
```

The priority order for multiple interrupt processing control has 8 levels, from 0 to 7 for each maskable interrupt request (0 is the highest priority), but it can be set as desired via software. Setting of the priority order level is done using the xxPRn0 to xxPRn2 bits of the interrupt control request register (xxICn), which is provided for each maskable interrupt request. After system reset, an interrupt request is masked by the xxMKn bit and the priority order is set to level 7 by the xxPRn0 to xxPRn2 bits.

The priority order of maskable interrupts is as follows.

(High) Level 0 > Level 1 > Level 2 > Level 3 > Level 4 > Level 5 > Level 6 > Level 7 (Low)

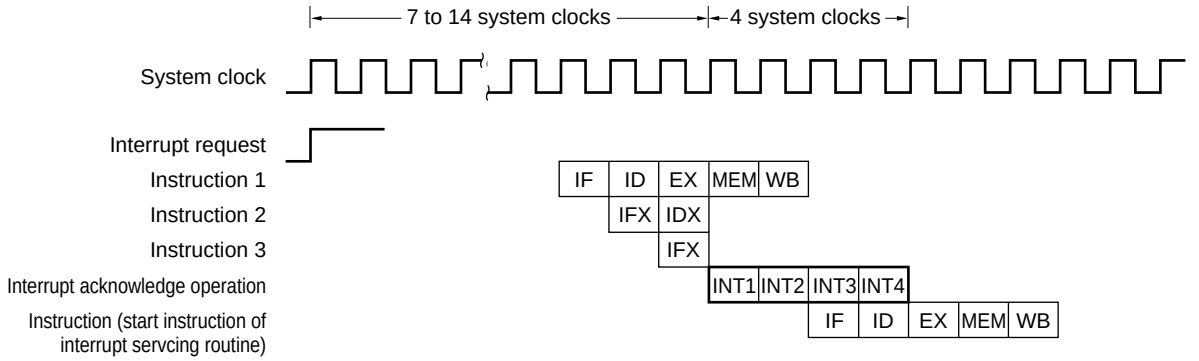
Interrupt processing that has been suspended as a result of multiple processing control is resumed after the processing of the higher priority interrupt has been completed and the RETI instruction has been executed.
A pending interrupt request is acknowledged after the current interrupt processing has been completed and the RETI instruction has been executed.

Caution: In a non-maskable interrupt processing routine (time until the RETI instruction is executed), maskable interrupts are suspended and not acknowledged.

4.8 Interrupt Response Time

The following table describes the V850/DB1 interrupt response time (from interrupt generation to start of interrupt processing).

Figure 4-19: Pipeline Operation at Interrupt Request Acknowledgment (Outline)



Remark: INT1 to INT4: Interrupt acknowledgment processing
IFX: Invalid instruction fetch
IDX: Invalid instruction decode

Table 4-4: Interrupt Response Time

	Interrupt Response Time (Internal System Clocks)		Condition
	Internal Interrupt	External interrupt INTP0 to INTP5, NMI	
Minimum	11	13 + analog delay time	The following cases are exceptions: • In WATCH or STOP mode • Two or more interrupt request non-sample instructions are executed • Access to interrupt control register
Maximum	18	20 + analog delay time	

4.9 Periods in Which Interrupts Are Not Acknowledged

An interrupt is acknowledged while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt non-sample instruction and the next instruction.
The interrupt request non-sampling instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- The store instruction for the interrupt control register (xxICn), in-service priority register (ISPR), and command register (PRCMD).

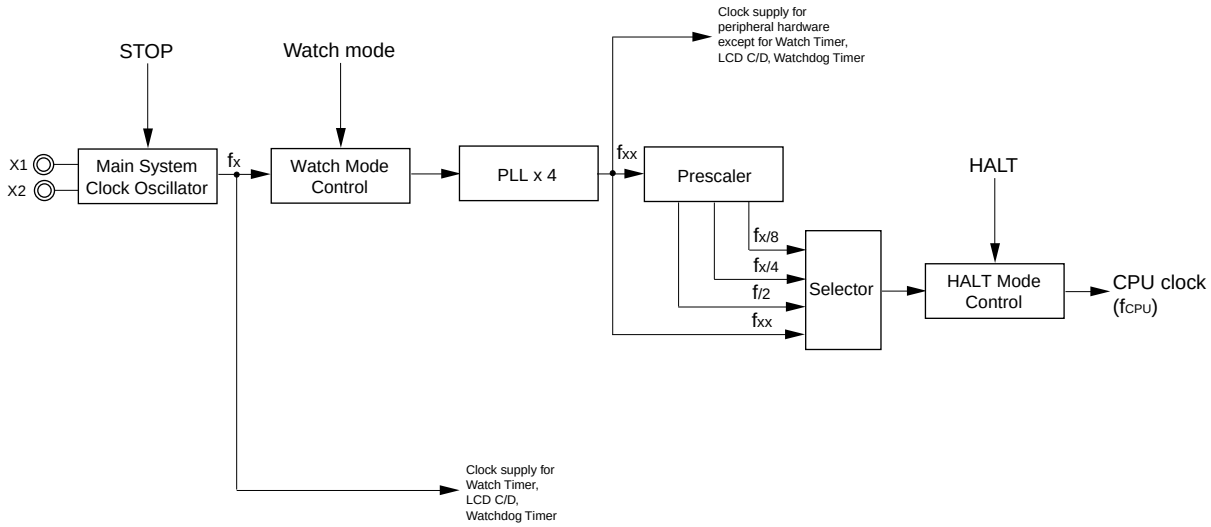
Chapter 5 Clock Generator

5.1 Features

- Multiplication function by PLL synthesizer: 4 x multiplication
- Clock sources
  - Oscillation through oscillator connection
- Power save modes
  - HALT mode
  - WATCH mode
  - STOP mode

5.2 Configuration

Figure 5-1: Block Diagram of the Clock Generator

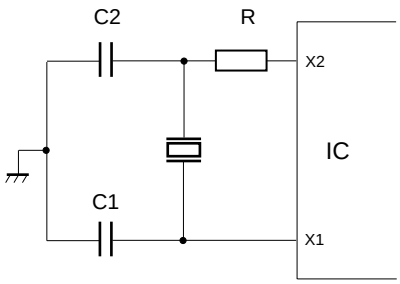


This block diagram does not necessarily show the exact wiring in hardware but the functional structure. For example the CLKSEL pin is not connected to the CV_{DD} of the PLL block but the function is as if.

5.3 Main System Clock Oscillator

The main system clock oscillator oscillates with a crystal resonator or a ceramic resonator connected to the X1 and X2 pins.

Figure 5-2: Main System Cock Oscillator



5.4 Control Registers

- PCC
- PSC
- CLOM

5.4.1 Processor Clock Control Register (PCC)

This register makes configuration of the prescaler of the CPU.
This is a specific register. It can be written to only when a specified combination of sequences is used. For details, see Chapter 3.4.9 “Specific registers” on page 106.

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

Figure 5-3: Format of Processor Clock Control Register (PCC)

	7	6	5	4	3	2	1	0	Address	R/W	After reset
PCC	0	0	0	0	0	0	CK1	CK0	FFFF F074H	R/W	03H

CK1	CK0	CPU clock
0	0	f_{XX}
0	1	$f_{XX}/2$
1	0	$f_{XX}/4$
1	1	$f_{XX}/8$

5.4.2 Power Save Control Register (PSC)

This is a specific register. It can be written to only when a specified combination of sequences is used. For details, see Chapter 5.4.3 “Writing a special Register” on page 145.

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

Figure 5-4: Format of Power Save Control Register (PSC)

	7	6	5	4	3	2	1	0	Address	R/W	After reset
PSC	0	0	0	0	0	WATCH	STP	0	FFFF F070H	R/W	00H

WATCH	WATCH Mode Setting
0	No WATCH mode
1	WATCH mode**Note 1**

STP	STOP Mode Setting
0	No STOP mode
1	STOP mode**Note 2**

Notes: 1. When WATCH mode is canceled, this bit is automatically reset to 0.

2. When STOP mode is canceled, this bit is automatically reset to 0.

5.4.3 Writing a special Register

This is an example how to write to a special Register. Here the PSC register is used as a demonstration.

Data is set in the power save control register (PSC) according to the following sequence.

<1> Prepare data in any one of the general-purpose registers to set to the specific register.
<2> Write arbitrary data to the command register (PRCMD).
<3> Set the power save control register (PSC) (with the following instructions).

- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

<4> Assert the NOP instructions (5 instructions (<5> to <9>).

Sample coding

```
<1> MOV 0x04, r10
<2> ST.Br10, PRCMD [r0]; Write PRCMD register
<3> ST.Br10, PSC [r0]  ; Set PSC register
<4> NOP                ; Dummy instruction
<5> NOP                ; Dummy instruction
<6> NOP                ; Dummy instruction
<7> NOP                ; Dummy instruction
<8> NOP                ; Dummy instruction
(next instruction)     ; Execution routine after software STOP mode and
                       ; IDLE mode release
```

No special sequence is required to read the specific register.

Cautions: 1. A store instruction for the command register (PRCMD) does not accept interrupts. This coding is made on assumption that <2> and <3> above are executed by the program with consecutive store instructions. If another instruction is set between <2> and <3>, the above sequence may become ineffective when the interrupt is accepted by that instruction, and a malfunction of the program may result.

2. Although the data written to the PRCMD register is dummy data, use the same register as the general register used in specific register setting <3> for writing to the PRCMD register (<2>). The same method should be applied when using a general register for addressing.

3. At least 5 NOP instructions must be inserted after executing a store instruction to the PSC register to set software STOP or WATCH mode.

5.4.4 Processor Clock Output Mode Register (CLOM)

The V850/DB1 supports clock output at the PCL-pin. The Output can be enabled or disabled and the frequency is selected according to the CLOM register.

Caution: To change the clock output, CLE must be set to 0 before changing the FS1 and FS0 bits.

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

Figure 5-5: Format of Clock Output Mode Register (CLOM)

	7	6	5	4	3	2	1	0	Address	R/W	After reset
CLOM	0	0	0	CLE	0	0	FS1	FS0	FFFF F38AH	R/W	00H

CLE	PCL operation control
0	PCL output disable (“0” output)
1	PCL output enable

FS1	FS0	PCL output
0	0	$f_{XX}/2$
0	1	$f_{XX}/4$
1	0	$f_{XX}/8$
1	1	$f_{XX}/16$

5.4.5 Oscillation stabilization time selection register (OSTS)

For the release from WATCH or STOP mode the V850/DB1 needs some time to switched on all of its peripherals. With the OSTS register a waiting time can be specified to hold the function of the CPU pending. Therefore the CPU starts with instruction fetch when all of the on-chip peripherals works well again after the release of the WATCH or STOP mode.

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

Figure 5-6: Format of Oscillation Stabilization Time Selection Register (OSTS)

	7	6	5	4	3	2	1	0	Address	R/W	After reset
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFF F380H	R/W	04H

OSTS2	OSTS1	OSTS0	Selection of Oscillation Stabilization Time	
			Clock	f_{XX} =16 MHz
0	0	0	$f_{XX}/2^{10}$	0.256 ms
0	0	1	$f_{XX}/2^{12}$	1.02 ms
0	1	0	$f_{XX}/2^{13}$	2.05 ms
0	1	1	$f_{XX}/2^{14}$	4.10 ms
1	0	0	$f_{XX}/2^{15}$	8.19 ms
Other than above			Setting prohibited	

Cautions: 1. From HALT or WATCH mode waken select $f_{xx}/2^{12}$ to $f_{xx}/2^{15}$

2. From STOP mode waken select $f_{xx}/2^{15}$

3. The wait time at the release of the STOP mode does not include the count time of the timebase clock to wait for the correct oscillator oscillation (see (1)“Securing the time using an on-chip time base counter” on page 157).

5.5 Power Saving Functions

5.5.1 General

The device provides the following power saving functions. These modes can be switched to suit the target application, which enables effective implementation of low-power systems.

Table 5-1: Power Saving Modes Overview

Mode	Clock Supply to		
	peripherals	CPU	watch
Normal	×	×	×
HALT	×	–	×
WATCH[Note]	–	–	×
STOP	–	–	–

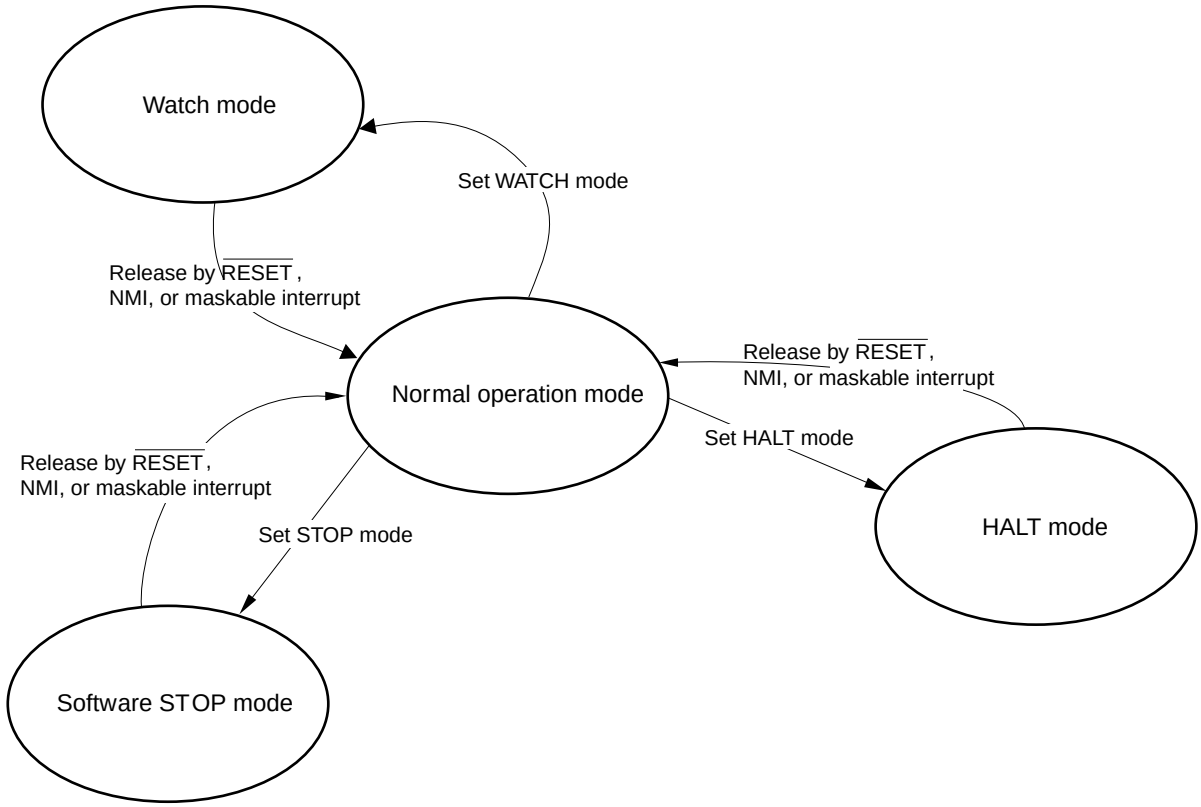
Remarks: 1. × Operates

2. – Stopped

Note: In WATCH mode the power supply to the flash memory is switched off.

Figure 5-7 shows the operation of the clock generator in normal operation mode, HALT mode, WATCH mode, and software STOP mode.
An effective low power consumption system can be realized by combining these modes and switching modes according to the required use.

Figure 5-7: Power Save Mode State Transition Diagram



5.5.2 Power Save Modes Outline

V850/DB1 is provided with the following standby modes: HALT, WATCH, and software STOP.
Application systems, which are designed so that these modes are switched appropriately according to operation purposes, reduce power consumption efficiently.

(1) HALT mode:

In this mode supply of the operating clock to the CPU is stopped whereby other on-chip peripheral functions continue to operate. Combining this mode with the normal operating mode to provide intermittent operations enables the overall system power consumption to be reduced.
This mode is entered by executing the dedicated instruction (HALT).

(2) WATCH mode:

In this mode, the clock generator stop to supply the clock excluding Watch Timer, Watchdog Timer and LCD unit. The entire system stops. This mode provides ultra-low power consumption, where the power consumed is only from OSC, Watch Timer, Watchdog Timer and LCD circuit.
This mode is entered by setting registers with software.

(3) Software STOP mode:

In this mode, the clock generator is stopped and the entire system stops. This mode provides ultra-low power consumption, where the power consumed is only leakage current.
This mode is entered by setting registers with software.

5.5.3 Operating States in Power Save Functions Mode

Table 5-2: Operating States in Power Save Functions Mode

Macro Name	HALT	WATCH	STOP
CPU	--	--	--
Prescaler (peripherals except WD, WT, LCD)	×	--	--
Prescaler (WD, WT, LCD)	×	×	--
Port	×	×	--
Interrupt controller	×	×	×
Watchdog Timer	×	×	--
Timer 5 (2 channels)	×	--	--
Timer C (2 channels)	×	--	--
Timer G	×	--	--
Watch Timer	×	×	--
Baud Rate Generator	×	×	--
DCAN (2 channel[Note])	×	--	--
DCAN RAM	×	--	--
UART5 (2 channels)	×	--	--
CSI0 (3 channels)	×	--	--
A/D	×	--	--
LCD Controller/Driver	×	×	--
Meter Controller/Driver (6 channels)	×	--	--
Processor Clock output	×	--	--
ROM	--	--	--
RAM	--	--	--
Voltage Regulator	×	×	×
Oscillator	×	×	--
PLL	×	--	--
RESET circuit	×	×	×

Notes: 1. ×: Can operate
--: Operation stopped

2. Two channels only for Flash product.
One channel for Mask product.

5.5.4 HALT mode

In this mode, the CPU clock is stopped, though the clock generators (oscillator and PLL synthesizer) continue to operate for supplying clock signals to other peripheral function circuits.
Setting the HALT mode when the CPU is idle reduces the total system power consumption. The state of the various hardware units in the HALT mode is tabulated below.

Table 5-3: Operating States in HALT mode

Items	Operation
Clock generator	Operating
Internal system clock	Operating
WT, WDT, LCD clock	Operating
CPU	Stopped
I/O line	Unchanged
Peripheral function	Operating
Internal data	Retains all internal data before entering HALT mode, such as CPU registers, state, data, and on-chip RAM.
CLKOUT (PCL pin)	Clock output (when not inhibited by port setting)

Remark: In the HALT mode, program execution is stopped but the contents of all registers and internal RAM prior are retained as is.
On-chip peripheral hardware irrelevant to the CPU instruction execution also continues to operate. Even after the HALT instruction is executed, instruction fetch operations continue until the internal instruction prefetch queue is full. After the queue becomes full, the CPU stops with the items set as tabulated above.

HALT mode release:

The HALT mode can be released by a non-maskable interrupt request, an unmasked maskable interrupt request, or $\overline{\text{RESET}}$ signal input.

(1) Release by interrupt request

The HALT mode is released unconditionally by an unmasked maskable interrupt request regardless of its priority level. However, if the HALT mode is entered during execution of an interrupt handler, the operation differs on interrupt priority levels as follows:

(a) If an interrupt request less prioritized than the currently serviced interrupt request is generated, the HALT mode is released but the interrupt is not acknowledged. The interrupt request itself is retained.

(b)If an interrupt request (including a non-maskable one) prioritized than the currently serviced interrupt request is generated, the interrupt request is acknowledged along with the HALT mode release.

Table 5-4: Operation after HALT mode release by interrupt request

Release cause	EI state	DI state
NMI request	Branches to handler address.	
Maskable interrupt request	Branches to handler address, or executes the next instruction.	Executes the next instruction.

Remark: If HALT mode is entered during execution of a particular interrupt handler and an unmasked interrupt request with a higher priority than the previous one is subsequently generated, the program branches to the vector address for the latter interrupt.

(2) Release by $\overline{\text{RESET}}$ pin input

This operation is the same as normal reset operation.

5.5.5 WATCH mode

In this mode f_{CPU} clock is stopped while the oscillator continue to operate to achieve low power, though only oscillator, Watch timer, Watchdog timer and LCD continue to operate.
This mode compensates the HALT modes concerning the oscillator stabilization time and power consumption. Only for the Flash memory an additional stabilization time is required.
This mode is entered by configuration of the PSC registers.
In the WATCH mode, program execution is stopped but the contents of all registers and internal RAM prior to entering this mode are retained. On-chip peripheral hardware operation is also stopped (except WT, WD and LCD). The state of the various hardware units in the WATCH mode is tabulated below.

Table 5-5: Operating States in WATCH Mode

Items	Operation
Clock generator	Operating
Internal system clock	Stopped
WT, WDT, LCD	Operating
CPU	Stopped
I/O line	Unchanged
Peripheral function	Stops excluding WT, WD, LCD
Internal data	Retains all internal data before entering WATCH mode, such as CPU registers, state, data, and on-chip RAM.
CLKOUT (PCL pin)	Unchanged**Note**

Note: If it is necessary that the PCL pin is “0”, the CLE bit of CLOM register have to set to “0”.

Watch mode release:

The WATCH mode can be released by a non-maskable interrupt request, an unmasked maskable interrupt request, or $\overline{\text{RESET}}$ signal input.

(1) Release by interrupt request:

The WATCH mode is released unconditionally by an unmasked maskable interrupt request regardless of its priority level. After 1 ms has passed, CPU starts operation.
However, if the WATCH mode is entered during execution of an interrupt handler, the operation differs on interrupt priority levels as follows:

(a) If an interrupt request less priorities than the currently serviced interrupt request is generated, the WATCH mode is release but the interrupt is not acknowledged. The interrupt request itself is retained.

(b) If an interrupt request (including a non-maskable one) priorities than the currently serviced interrupt request is generated, the interrupt request is acknowledged along with the WATCH mode release.

Table 5-6: Operation after WATCH mode release by interrupt request

Release cause	EI state	DI state
NMI request	Branches to handler address.	
Maskable interrupt request	Branches to handler address, or executes the next instruction.	Executes the next instruction.

Remark: If WATCH mode is entered during execution of a particular interrupt handler and an unmasked interrupt request with a higher priority than the previous one is subsequently generated, the program branches to the vector address for the later interrupt.

(2) When released by $\overline{\text{RESET}}$ input

This operation is the same as normal reset operation. PLL stabilization time (> 1ms) must be ensured by external reset input.

(3) When released by Watchdog Timer $\overline{\text{RESET}}$ input

After 1 ms has passed, CPU starts operation.

5.5.6 Software STOP mode

In this mode, the CPU clock is stopped including the clock generators (oscillator and PLL synthesizer), resulting in stop of the entire system for ultra-low power consumption (the only consumed is device leakage current). When this mode is released, the oscillation stabilization time for the oscillator should be secured until the system clock is stabilized. However, when the external clock operates this product, securing the oscillation stabilization time for the oscillator until the system clock is stabilized is unnecessary. In the direct mode as well, the lock-up time does not have to be secured.
This mode is entered by setting the PSC register.
In this mode, the program execution stops, but the contents of all registers and internal RAM prior to entering this mode are retained. none peripheral operations are also stopped.
The state of the various hardware units in the software STOP mode is tabulated below.

Table 5-7: Operating States in STOP Mode

Items	Operation
Clock generator	Stopped
Internal system clock	Stopped
WT, WDT, LCD clock	Stopped
CPU	Stopped
I/O line **Note 1**	Unchanged
Peripheral function	Stopped
Internal data **Note 1**	Retains all previous internal data, such as CPU registers, state, data, and on-chip RAM.
CLKOUT (PCL pin)	Unchanged**Note 2**

Notes: 1. When the V_{DD} value is within the operating range. However, even if V_{DD} falls below the lowest operating voltage, the internal RAM content is retained as long as the data retention voltage V_{DDDR} is maintained.

2. If it is necessary that the PCL pin is “0”, the CLE bit of CLOM register have to set to “0”.

STOP mode release:

The STOP mode can be released by a non-maskable interrupt request, an unmasked maskable interrupt request or $\overline{\text{RESET}}$ signal input.

5.6 Securing Oscillation Stabilization Time

5.6.1 Oscillation stabilization time security specification

Two methods can be used to secure the required stabilization times when WATCH mode or software STOP mode is released.

(1) Securing the time using an on-chip time base counter

WATCH mode and software STOP mode are released when a valid edge is input to the NMI pin or a maskable interrupt request is input (INTPn). Valid edge input to the pin causes a time base counter to start counting, and the time until the clock output from the oscillator stabilizes is secured during that counting time.

After a fixed time, internal system clock output begins, and processing branches to the NMI interrupt or maskable interrupt (INTPn) handler address.

Figure 5-8: WATCH mode release by NMI or INTPn

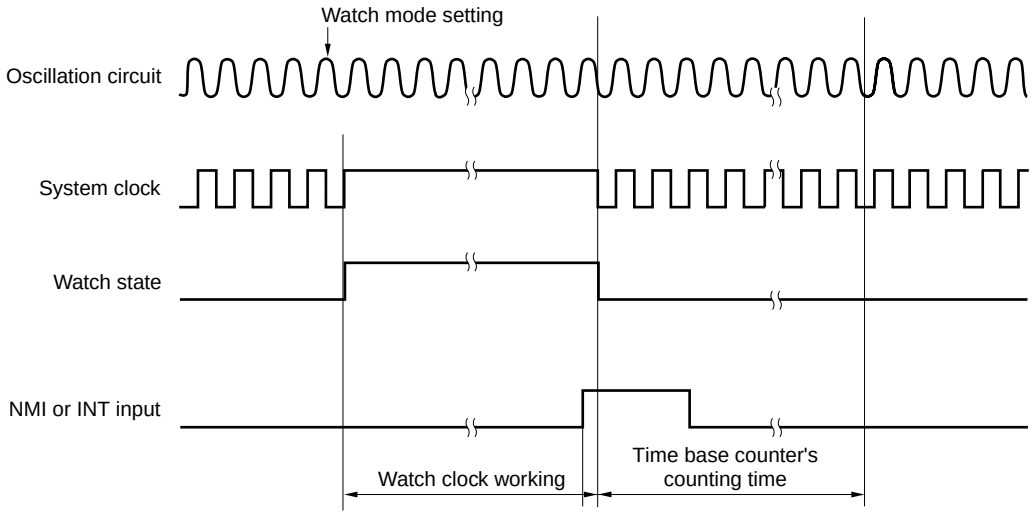
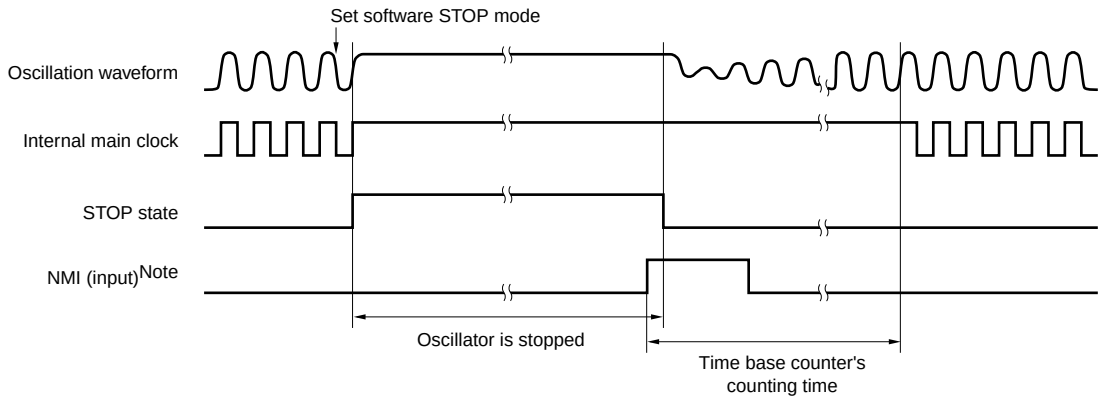


Figure 5-9: STOP mode release by NMI or INTPn



Note: Valid edge: When specified as the rising edge.

The NMI pin should usually be set to an inactive level (for example, high level when the valid edge is specified as the falling edge) in advance.
Watch mode and software STOP mode are immediately released if an operation is performed according to NMI valid edge input or maskable interrupt request input (INTPn) timing in which STOP mode is set until the CPU acknowledges the interrupt.

(2) Securing the time according to the signal level width ($\overline{\text{RESET}}$ pin input)

WATCH mode and software STOP mode are released due to falling edge input to the $\overline{\text{RESET}}$ pin. The time until the clock output from the oscillator stabilizes is secured according to the low level width of the signal that is input to the pin.

The supply of internal system clocks begins after a rising edge is input to the $\overline{\text{RESET}}$ pin, and processing branches to the handler address used for a system reset.

Figure 5-10: WATCH mode release by $\overline{\text{RESET}}$-pin input or watchdog timer

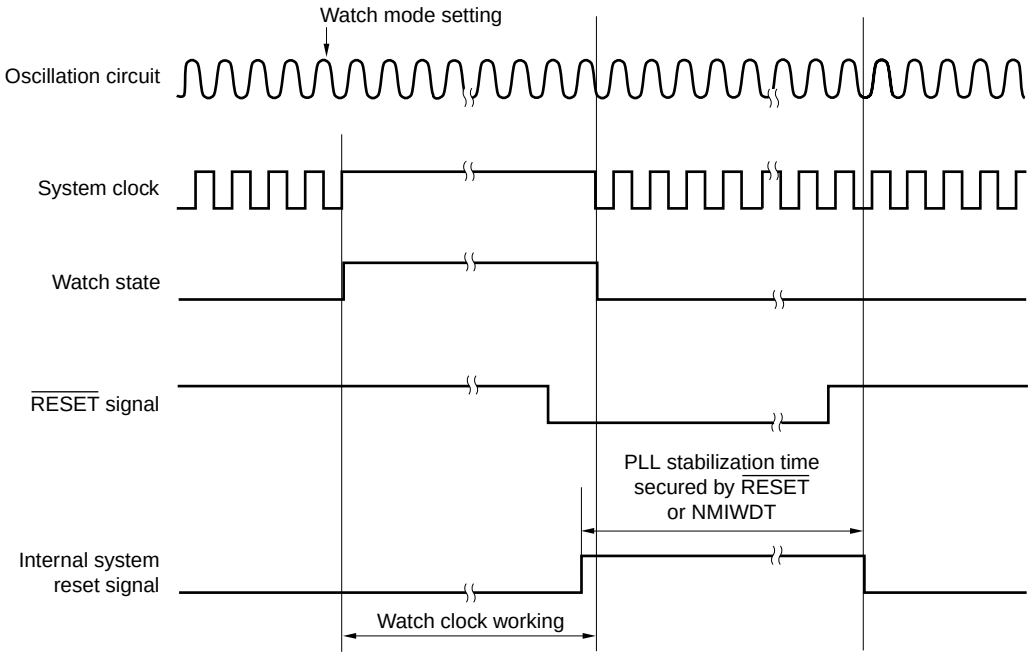
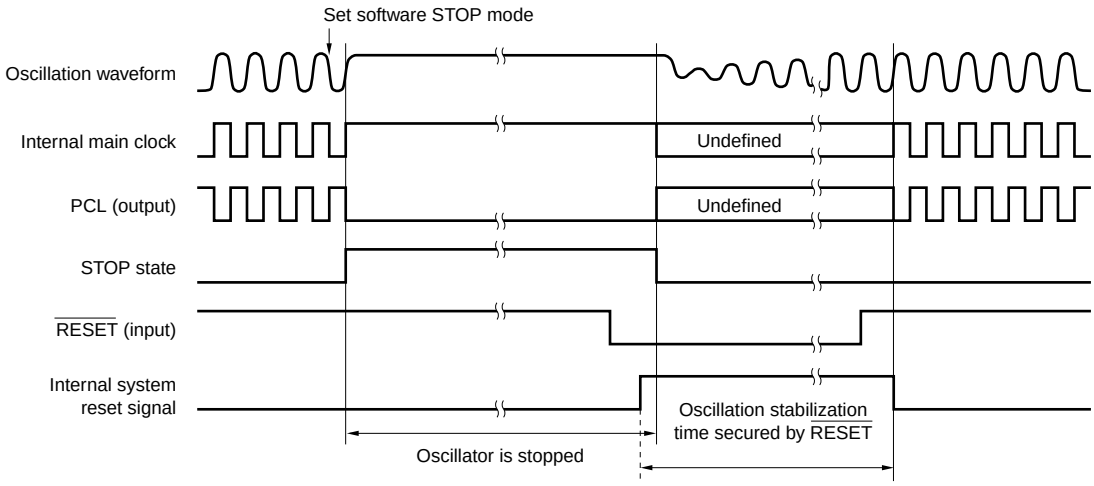


Figure 5-11: STOP mode release by $\overline{\text{RESET}}$ pin input



[MEMO]

Chapter 6 Timer

6.1 Timer G

6.1.1 Features (Timer G)

The 6-channel 16-bit multi purpose Timer G operates as

- Pulse interval and frequency measurement counter
- event counter
- Interval timer
- Programmable pulse output
- PWM output timer

Remark: In this Timer G chapter following indexes were consequently used

- m = 1 to 4 (for the free assignable Input/Output-channels)
- n = 0 to 5 (for all CC-channels)
- x = 0, 1 (for bit-index, i.e. 2 counter for each Timer Gn)

6.1.2 Function overview (Timer G)

- 16-bit timer/counter (TMG0, TMG1): 2 channels

- Bit length
  - Timer G registers (TMG0, TMG1): 16 bits

- Capture/compare register: 6
  - 16-bit
  - 2 registers are assigned fix to the corresponding one of the 2 counters
  - 4 free assignable registers to one of the 2 counters

- Count clock division selectable by prescaler (frequency of system clock: f_{XX}=16 MHz)
  - In 8 steps from $f_{XX}/2$ to $f_{XX}/256$

- Interrupt request sources
  - Edge detection circuit with noise elimination.
  - Compare-match interrupt requests: 6 types
    Perform comparison of capture/compare register with one of the 2 counters (TMG0, TMG1) and generate the INTGCC0 to INTGCC5 interrupt upon compare match.
  - Timer counter overflow interrupt requests: 2 types
    In free run mode the INTGOVF0 (INTGOVF1) interrupt is generated when the count value of TMG0 (TMG1) toggles from FFFFH to 0000H.
    In match and clear mode the INTGOVF0 (INTGOVF1) interrupt is generated when the count value of TMG0 (TMG1) matches the GCC0 (GCC1) value.

- PWM output function
  - Control of the outputs of pins TOG1 through TOG4 in the compare mode and PWM output can be performed using the compare match timing of the GCC1 to GCC4 register and the corresponding timebase (TMG0, TMG1).

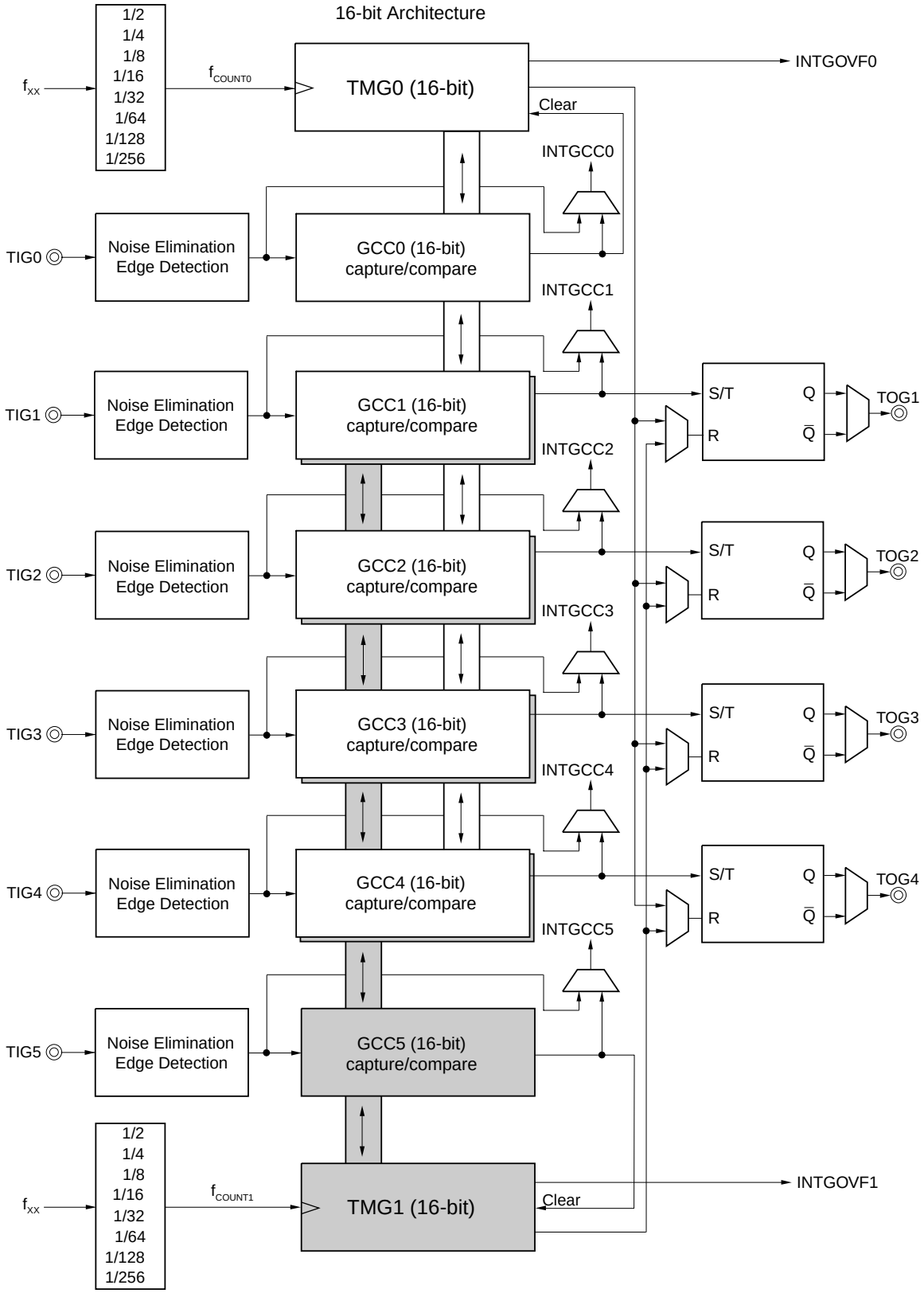
- Output delay operation
  - A clock-synchronized output delay can be added to the output signal of pins TOG1 through TOG4.
  - This is effective as an EMI counter measure.

- Edge detection and noise elimination filter
  - External signals shorter than 2 count clock (f_{COUNT}, not f_{XX}) are eliminated as noise.

Note: The TIG1 to TIG4 and TOG1 to TOG4 are each alternate function pins.

Figure 6-1, “Block Diagram of Timer G,” on page 163 shows the block diagram of Timer G.

Figure 6-1: Block Diagram of Timer G



Remark: f_{XX}: Internal system clock (16 MHz)

Note: TMG0/TMG1 are cleared by GCC0/5 register compare match.

6.1.3 Basic configuration

The basic configuration is shown below.

Table 6-1: Timer G Configuration List

Timer	Count Clock	Register	R/W	Generated Interrupt Signal	Capture Trigger	Timer Output PWM
Timer G	f_{XX} /2, f_{XX} /4, f_{XX} /8, f_{XX} /16, f_{XX} /32, f_{XX} /64, f_{XX} /128, f_{XX} /256	TMG0	R	INTGOVF0	-	-
		TMG1	R	INTGOVF1	-	-
		GCC0	R/W	INTGCC0	TIG0	-
		GCC1	R/W	INTGCC1	TIG1	TOG1
		GCC2	R/W	INTGCC2	TIG2	TOG2
		GCC3	R/W	INTGCC3	TIG3	TOG3
		GCC4	R/W	INTGCC4	TIG4	TOG4
		GCC5	R/W	INTGCC5	TIG5	-

Remark: f_{XX}: Internal system clock (16 MHz)

(1) Timer G 16-bit counter register (TMG0, TMG1)

The features of the 2 counters TMG0 and TMG1 are listed below:

- Free-running counter that enables counter clearing by compare match of register GCC0/GCC5.
- Counter clear can be set by software.
- Counter stop can be set by software.

These registers can be read in 16 bit units.

Figure 6-2: Timer G Counter Value Registers TMG0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
TMG0																	FFFF F210H	0000H

Figure 6-3: Timer G Counter Value Registers TMG1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
TMG1																	FFFF F214H	0000H

(2) Timer G capture/compare registers of the 2 counters (GCC0, GCC5)

The GCC0, GCC5 register is a 16-bit capture/compare register of Timer G. This registers are fixed assigned to the counter registers (TMG0 and TMG1).

In the **capture register mode**, GCC0 and GCC5 captures the TMG0 (TMG1) count value if an edge is detected at Pin TIG0 (TIG5).

In the **compare register mode**, GCC0 and GCC5 detects match with TMG0 (TMG1) and clears the assigned Timebase. So this “match and clear mode” is used to reduce the number of valid bits of the counter TMG0 (TMG1).

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

Caution: If in Compare Mode write to this registers before POWER and ENFGx bit (x = 0, 1) are “1“at the same time

Figure 6-4: Timer G counter TMG0 assigned Capture/Compare Register (GCC0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
GCC0																	FFFF F218H	0000H

Remark: This register is assigned fix to timebase TMG0.

Figure 6-5: Timer G counter TMG1 assigned Capture/Compare Register (GCC5)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
GCC5																	FFFF F22CH	0000H

Remark: This register is assigned fix to timebase TMG1.

(3) Timer G capture/compare registers with external PWW-output function (GCC1 to GCC4)

The GCC1 to GCC4 register is a 16-bit capture/compare register of Timer G. It can be assigned to one of the 2 counters either TMG0 or TMG1.

In the **capture register mode**, this register captures the value of TMG0 when the TBGm bit (m = 1 to 4) of the TMGCMH register = 0. When the TBGm bit = 1, this register holds the value of TMG1.

In **compare mode**, this register represents the actual compare value and the TOGm-Output (m = 1 to 4) can generates a PWW if it is activated.

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

Figure 6-6: Timer G free assignable Capture/Compare Registers (GCCm) (m = 1 to 4)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
GCC1																	FFFF F21CH	0000H
GCC2																	FFFF F220H	0000H
GCC3																	FFFF F224H	0000H
GCC4																	FFFF F228H	0000H

Remarks: 1. In capture mode only reading is possible

2. In compare mode read/write is possible.

6.1.4 Control Registers

(1) Timer G Mode Register High (TMGMH)

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

Remark: Some bits and registers can't be rewrite if the POWER bit is set. However the OLDE bit, CSEx2 to CSEx0 bits of the TMGMH register and the CCSG5, CCSG0 bit of the TMGML register can be set with the POWER bit simultaneously.

Figure 6-7: Timer G Mode Register High (TMGMH)

	7	6	5	4	3	2	1	0	Address	Initial value
TMGMH	POWER	OLDE	CSE12	CSE11	CSE10	CSE02	CSE01	CSE00	FFFF F202H	00H

Bit Position	Bit Name	Function
7	POWER	Timer G Operation control. 0: Operation Stop the capture registers and TMGST register are cleared the TOGm pins (m = 1 to 4) are inactive all the time 1: Operation enable **Remark:** At least 14 system-clocks (f_{XX}) are need to start the timer function
6	OLDE	Set Output Delay Operation. 0: Don't perform output delay operation 1: Set output delay to n count-clocks **Caution: When the POWER bit is set, the rewriting of this Bit is prohibited! Simultaneously writing with the POWER bit is allowed.** **Remark:** The delay operation is used for EMI counter measures.
5 to 3, 2 to 0	CSEx2, CSEx1, CSEx0	Selects internal count clock of TMG **Caution: When the POWER bit is set, the rewriting of this Bits are prohibited! Simultaneously writing with the POWER bit is allowed.** **Remarks: 1.** x = 0, 1 **2.** f_{XX}: system clock

CSEx2	CSEx1	CSEx0	Count Clock
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	$f_{XX}/128$
1	1	1	$f_{XX}/256$

(2) Timer G Mode Register Low (TMGML)

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

Figure 6-8: Timer G Mode Register* Low *(TMGML)

	7	6	5	4	3	2	1	0	Address	Initial value
TMGML	CCSG5	CCSG0	0	0	CLRG1	TMG1E	CLRG0	TMG0E	FFFF F200H	00H

Bit Position	Bit Name	Function
7, 6	CCSG5, CCSG0	Specifies the mode of the TMG0 (TMG1)(CCSG5 for TMG1, CCSG0 for TMG0): 0: Free-run mode for TMG1 (TMG0), GCC5 (GCC0) in capture mode (an detected edge at Pin TIG5 (TIG0) stores the value of TMG1 (TMG0) in GCC5 (GCC0) and an interrupt INTGCC5 (INTGCC0) is output) 1: Match and Clear mode of the TMG1 (TMG0), GCC5 (GCC0) in compare mode (when the data of GCC5 (GCC0) match the count value of the TMG1 (TMG0), the counter is cleared and the interrupt INTGCC5 (INTGCC0) occurs) **Caution: When the POWER bit is set, the rewriting of this Bits are prohibited! Simultaneously writing with the POWER bit is allowed.**
3, 1	CLRGx	Specifies software clear for TMGx (x = 0, 1): 0: Continue TMGx operation 1: Clears (0) the count value of TMGx, the corresponding TOGx is deactivated **Remark:** TMGx starts 2 system-clocks after this bit is set this bit is not readable (always read 0)
2, 0	TMGxE	Specifies TMGx (x = 0, 1) count operation enable/disable 0: Stop count operation the counter holds the immediate preceding value the corresponding TOGx is deactivated 1: Enable count operation **Remarks: 1.** the counter needs at least 2 system-clocks (f_{XX}) to stop **2.** the counter needs at least 8 system-clocks (f_{XX}) to start

(3) Timer G Channel Mode Register High and Low (TMGCMH, TMGCML)

This registers specifies the assigned counter (TMG0 or TMG1) for the GCCm register (m = 1 to 4). Furthermore it specifies the edge detection for the TIGn-input-pins (n = 0 to 5).

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

Caution: When the POWER bit is set, the rewriting of this registers is prohibited!

Figure 6-9: Timer G Channel Mode Register (TMGCMH, TMGCML)

	7	6	5	4	3	2	1	0	Address	Initial value
TMGCMH	TBG4	TBG3	TBG2	TBG1	IEG51	IEG50	IEG41	IEG40	FFFF F206H	00H

	7	6	5	4	3	2	1	0	Address	Initial value
TMGCML	IEG31	IEG30	IEG21	IEG20	IEG11	IEG10	IEG01	IEG00	FFFF F204H	00H

Bit Position	Bit Name	Function
TMGCMH 7 to 4	TBGm	Assigns Capture/Compare registers GCC1 to GCC4 to one of the 2 counters TMG0 or TMG1: 0: SetTMG0 as the corresponding counter to GCCm register and TIGm/ TOGm-pin 1: SetTMG1 as the corresponding counter to GCCm register and TIGm/ TOGm-pin
TMGCMH 3 to 0 and TMGCML 7 to 0	IEGn1, IEGn0	Specifies the valid edge of external capture signal input pin (TIGm) for the capture register performing capture-match with the assigned counter TMG0 or TMG1:

IEGn1	IEGn0	Valid Edge
0	0	Falling edge
0	1	Rising edge
1	0	No edge detection performed
1	1	Both rising and falling edges

Remarks: 1. n = 0 to 5

2. m = 1 to 4

(4) Timer G output control register (OCTLGH, OCTLGL)

This registers controls the timer output from the TOGm pin (m = 1 to 4) and the capture or compare mode for the GCCm register.

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

Cautions: 1. When the POWER bit is set, the rewriting of CCSGm is prohibited

2. When the POWER and TMG0E bit (TMG1E bit) are set at the same time, the rewriting of the ALVGm bits are prohibited.

Figure 6-10: Timer G Output Control Register (OCTLGH, OCTLGL)

	7	6	5	4	3	2	1	0	Address	Initial value
OCTLGH	SWFG4	ALVG4	CCSG4	0	SWFG3	ALVG3	CCSG3	0	FFFF F20AH	44H

	7	6	5	4	3	2	1	0	Address	Initial value
OCTLGL	SWFG2	ALVG2	CCSG2	0	SWFG1	ALVG1	CCSG1	0	FFFF F208H	44H

Bit Position	Bit Name	Function
7, 3	SWFGm	Fixes the TOGm pin output level according to the setting of ALVGm bit. 0: disable TOGm to inactive level 1: enable TOGm
6, 2	ALVGm	Specifies the active level of the TGOm pin output. 0: Active level is 0 1: Active level is 1 **Caution: Don't write this bit, before ENFG0 or ENFG1 of TMGST is 0, so first clear TMG0E or TMG1E bit of the TMGM register and check ENFG0 or ENFG1 bit before writing.**
5, 1	CCSGm	Specifies Capture/Compare mode selection: 0: Capture mode: if external edge is detected the INTGCCm interrupt occurs, the corresponding counter value is written to GCCm 1: Compare mode: if GCCm matches with corresponding timebase the INTGCCm interrupt occurs, if SWFGm is set the PWM output mode is set **Caution: Don't write this bit, before POWER bit of TMGMH is 0.**

Remark: m = 1 to 4

(5) Timer G status register (TMGST)

The TMGST register indicates the status of TMG0 and TMG1.

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

Figure 6-11: Timer G Status Register (TMGST)

	7	6	5	4	3	2	1	0	Address	Initial value
TMGST	ENFG1	ENFG0	CCFG5	CCFG4	CCFG3	CCFG2	CCFG1	CCFG0	FFFF F20CH	00H

Bit Position	Bit Name	Function
5 to 0	CCFGn	Indicates TMG0 or TMG1 overflow status. 0: No overflow 1: Overflow **Caution: The CCFGn bit is set if a TMG0 (TMG1) overflow occurs. This flag is only updated if the corresponding GCCn register was read, so first read the GCCn register and then read this flag if necessary**
7 to 6	ENFG1, ENFG0	Indicates TMG1 (TMG0) operation. 0: indicates operation stopped 1: indicates operation

Remark: n = 0 to 5

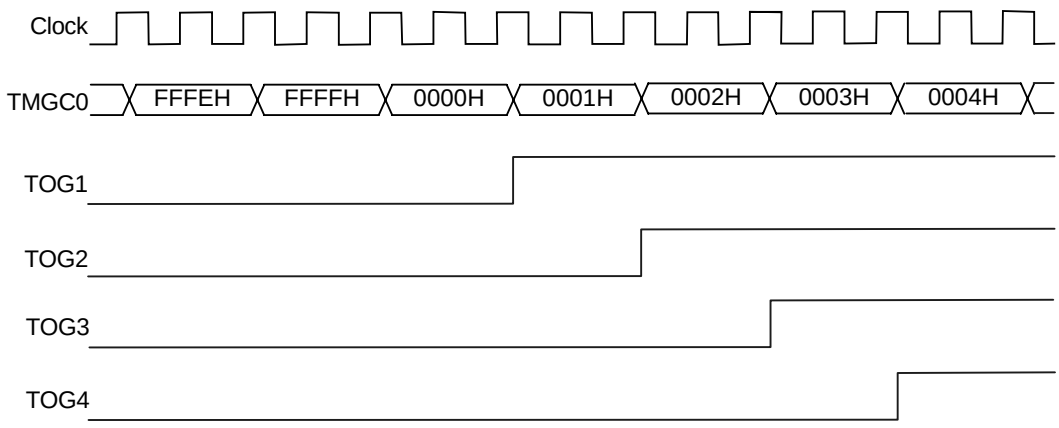
6.1.5 Output delay operation

When the OLDE bit is set, different delays of count clock period are added to the TOGm-pins:

Output-pin	delay $1/f_{COUNT}$
TOG1	0
TOG2	1
TOG3	2
TOG4	3

The figure below shows the timing for the case where the count clock is set to $f_{XX}/2$. However, 0FFFH is set in GCC0.
Similar delays are added also when a transition is made from the active to inactive level. So, a relative pulse width is guaranteed.

Figure 6-12: Timing of Output delay operation



In this case the count clock is set to $f_{XX}/2$.

6.1.6 Explanation of Basic Operation

(1) Overview of the mode settings

The Timer G includes 2 channels of 16-bit counters (TMG0/TMG1), which can operate as independent timebases. TMG0 (TMG1) can be set by CCSG0 (CCSG5) in the

- free-run mode
- match and clear mode.

When a timer output (TOGm) or INTGCCm interrupt is used, one of the two counters can be selected by setting the TBGm bit (m = 1 to 4) of the TMGCM register
The tables below indicate the interrupt output and timer output states dependent on the register setting values.

Register setting value				State of each output pin			
CCSG0	TBGm	SWFGm	CCSGm	INTGOVF0	INTGCC0	INTGCCm	TOGm
0 Free-run mode	0	0	0	Overflow interrupt	TI0 edge detection	TIm edge detection	Tied to inactive level
			1			CMPGm match	
		1	0			TIm edge detection	
			1			CMPGm match	PWM (free run)
1 Match and clear mode		0	0	Overflow interrupt[Note 1]	CMPG0 match[Note 2]	TIm edge detection	Tied to inactive level
			1			CMPGm match	
		1	0			TIm edge detection	
			1			CMPGm match	PWM (match and clear)

Notes: 1. An interrupt is generated only when the value of the GCC0 register is FFFFH.

2. An interrupt is generated only when the value of the GCC0 register is not FFFFH.

Remark: The setting of the CCSGm bit in combination with the SWFGm bit sets the mode for the timing of the actualization of new compare values.

- In compare mode the new compare value will be immediately active.
- In PWM mode the new compare value will be active first after the next overflow or match & clear of the assigned counter (TMG0, TMG1).

Register setting value				State of each output pin			
CCSG5	TBGm	SWFGm	CCSGm	INTGOVF1	INTGCC5	INTGCCm	TOGm
0 Free-run mode	1	0	0	Overflow interrupt	TI5 edge detection	TIm edge detection	Tied to inactive level
			1			CMPGm match	
		1	0			TIm edge detection	
			1			CMPGm match	PWM (free run)
1 Match and clear mode		0	0	Overflow interrupt[Note 1]	CMPG5 match[Note 2]	TIm edge detection	Tied to inactive level
			1			CMPGm match	
		1	0			TIm edge detection	
			1			CMPGm match	PWM (match and clear)

Notes: 1. An interrupt is generated only when the value of the GCC5 register is FFFFH.

2. An interrupt is generated only when the value of the GCC5 register is not FFFFH.

6.1.7 Operation in Free-run mode

This operation mode is the standard mode for Timer G operations. In this mode the 2 counter TMG0 and TMG1 are counting up from 0x0000 to 0xFFFF, generates an overflow and start again. In the match and clear mode, which is described in 6.1.8 on page 185 the fixed assigned register GCC0 (GCC5) is used to reduce the bit-size of the counter TMG0 (TMG1).

(1) Capture operation (free run)

Basic settings (m = 1 to 4):

Bit	Value	Remark
CCSG0	0	free run mode
CCSG5	0	
SWFGm	0	disable TOGm
TBGm	X	assign counter for GCCm 0: TMG0 1: TMG1

(a) Example: Pulse width or period measurement of the TIGn input signal (free run)

Capture setting method:

(1) When using one of the TIG1 to TIG4 pins, select the corresponding counter with the TBGm bit. When TIG0 is used, the corresponding counter is TMG0. When TIG5 is used, the corresponding counter is TMG1.
(2) Select a count clock cycle with the CSE12 to CSE10 bits (TMG1) or CSE02 to CSE02 bits (TMG0).
(3) Select a valid TIGn edge with the IEGn1 and IEGn0 bit. A rising edge, falling edge, or both edges can be selected.
(4) Start timer operation by setting POWER bit and TMG0E bit for TMG0 or TMG1E bit for TMG1.

Capture Operation

(1) When a specified edge is detected, the value of the counter is stored in GCCn, and an edge detection interrupt (INTGCCn) is output.
(2) When the counter overflows, an overflow interrupt (INTGOVF0 or INTGOVF1) is generated.
(3) If an overflow has occurred between capture operations, the CCFGn flag is set when GCCn is read. Correct capture data by checking the value of CCFGn.

Using CCFGn:

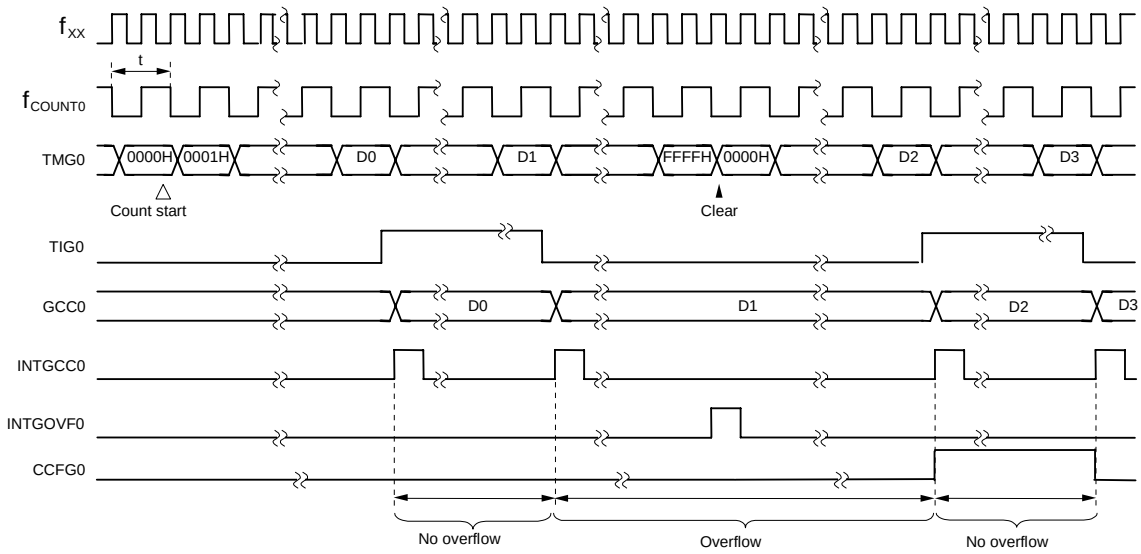
When using GCCn as a capture register, use the procedure below.

<1> After INTGCCn (edge detection interrupt) generation, read the corresponding GCCn register.
<2> Check if the corresponding CCFGn bit of the TMGST register is set.
<3> If the CCFGn bit is set, the counter was cleared from the previous captured value.

CCFGn is set when GCCn is read. So, after GCCn is read, the value of CCFGn should be read. Using the procedure above, the value of CCFGn corresponding to GCCn can be read normally.

Caution: If two or more overflows occur between captures, a software-based measure needs to be taken to count overflow interrupts (INTGOVF0, INTGOVF1).

Figure 6-13: Timing when both edges of TIG0 are valid (free run)



Remark: The figure above shows an image. In actual circuitry, 3 to 4 periods of the count-up signal are required from the input of a waveform to TIG0 until a capture interrupt is output. See (b)“Timing of capture trigger edge detection” on page 177.

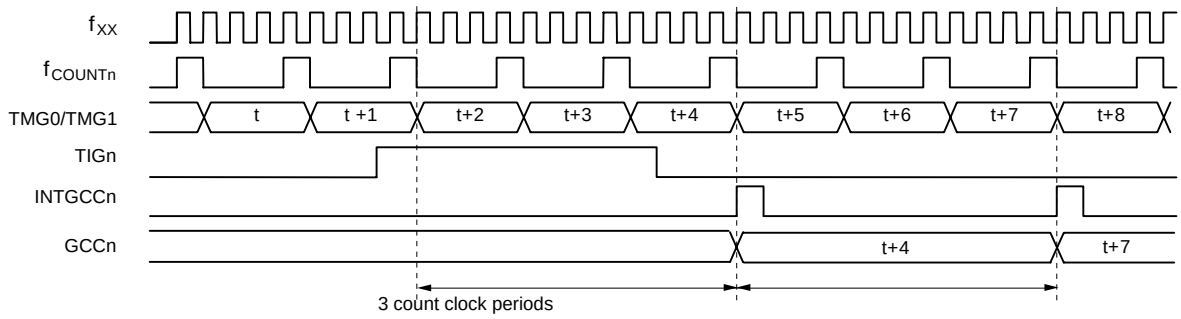
(b) Timing of capture trigger edge detection

The Tin inputs are fitted with an edge-detection and noise-elimination circuit.
Because of this circuit, 3 periods to less than 4 periods of the count clock are required from edge input until an interrupt signal is output and capture operation is performed. The timing chart is shown below.

Basic settings (n = 0 to 5):

Bit	Value	Remark
CSEn2	0	Count clock = $f_{XX}/4$
CSEn1	0	
CSEn0	1	
IEGn1	1	detection of both edges
IEGn0	1	

Figure 6-14: Timing of capture trigger edge detection (free run)



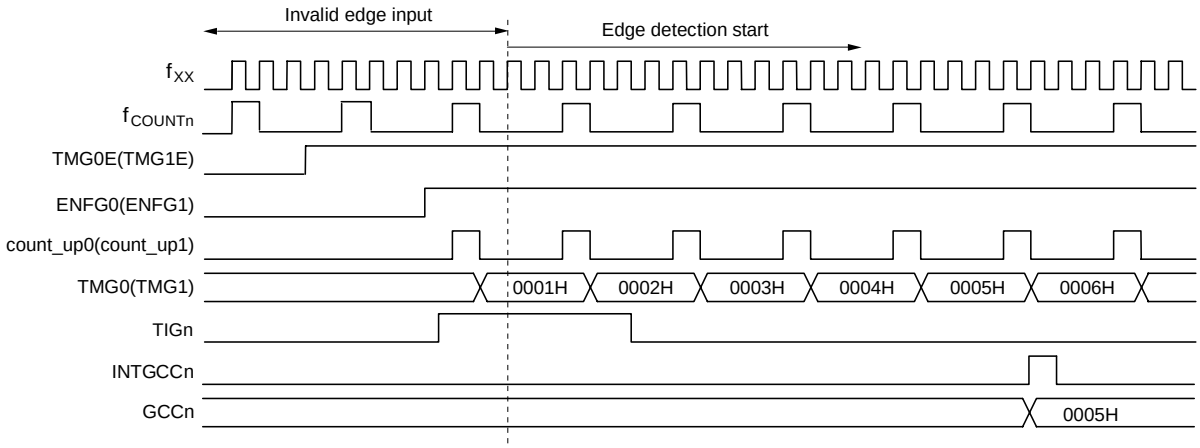
(c) Timing of starting capture trigger edge detection

A capture trigger input signal (TIGn) is synchronized in the noise eliminator for internal use. Edge detection starts when 1 count clock period (f_{COUNT}) has been input after timer count operation starts. (This is because masking is performed to prevent the initial TIGn level from being recognized as an edge by mistake.). The timing chart for starting edge detection is shown below.

Basic settings (x = 0, 1 and n = 0 to 5):

Bit	Value	Remark
CSEx2	0	Count clock = $f_{XX}/4$
CSEx1	0	
CSEx0	1	
IEGn1	1	detection of both edges
IEGn0	1	

Figure 6-15: Timing of starting capture trigger edge detection



(2) Compare operation (free run)

Basic settings (m = 1 to 4):

Bit	Value	Remark
CCSG0	0	free run mode
CCSG5	0	
SWFGm	0	disable TOGm
CCSGm	1	Compare mode for GCCm
TBGm	X	assign counter for GCCm 0: TMG0 1: TMG1

(a) Example: Interval timer (free run)

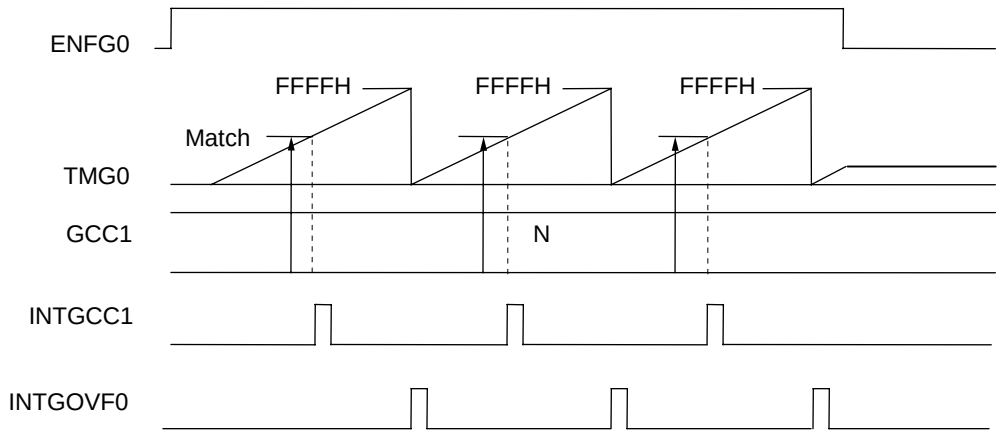
Setting method interval timer:

(1) An usable compare register is one of GCC1 to GCC4, and the corresponding counter (TMG0 or TMG1) must be selected with the TBGm bit.
(2) Select a count clock cycle with the CSE12 to CSE10 (TMG1) bits or CSE02 to CSE00 (TMG0) bits.
(3) Write data to GCCm.
(4) Start timer operation by setting POWER and TMG0E (or TMG1E).

Compare Operation:

(1) When the value of the counter matches the value of GCCm (m = 0 to 4), a match interrupt (INTGCCm) is output.
(2) When the counter overflows, an overflow interrupt (INTGOVF0/INTGOVF1) is generated.

Figure 6-16: Timing of compare mode (free run)



Data N is set in GCC1, and the counter TMG0 is selected

(b) When the value 0000H is set in GCCm

INTGCCm is activated when the value of the counter becomes 0001H.
INTGOVF0/INTGOVF1 is activated when the value of the counter changes from FFFFH to 0000H.
Note, however, that even if no data is set in GCCm, INTGCCm is activated immediately after the counter starts.

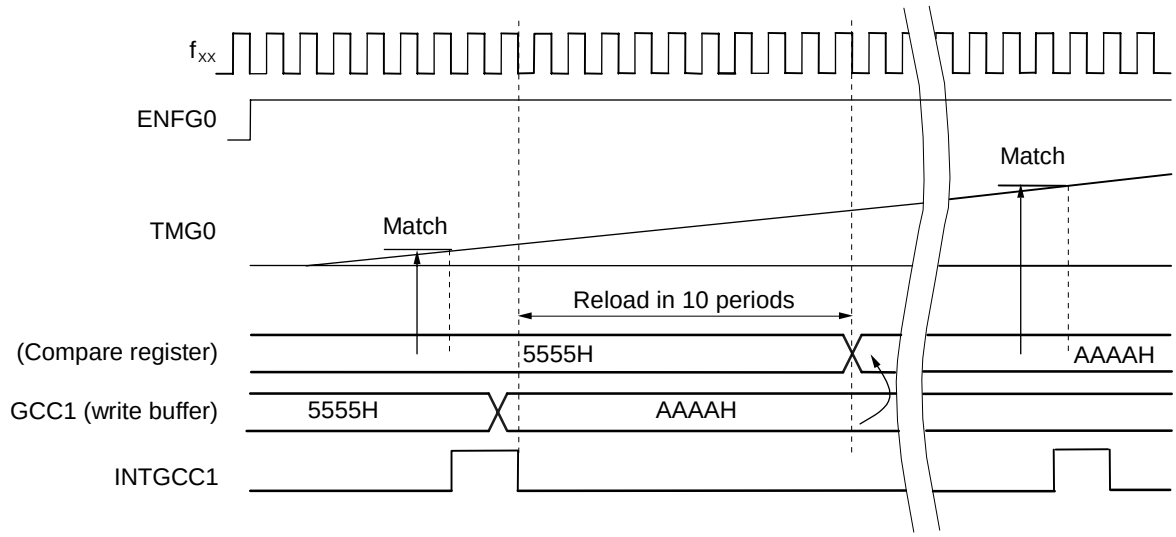
(c) When the value FFFFH is set in GCCm

INTGCCm and INTGOVF0/INTGOVF1 are activated when the value of the counter changes from FFFFH to 0000H.

(d) When GCCm is rewritten during operation

When GCC1 is rewritten from 5555H to AAAAH. TMG0 is selected as the counter.
The following operation is performed:

Figure 6-17: Timing when GCC1 is rewritten during operation (free run)



Caution: To perform successive write access during operation, for rewriting the GCCn register (n = 1 to 4), you have to wait for minimum 14 system clocks periods (f_{XX}).

(3) PWM output (free run)

Basic settings (m = 1 to 4):

Bit	Value	Remark
CCSG0	0	free run mode
CCSG5	0	
SWFGm	1	enable TOGm
CCSGm	1	Compare mode for GCCm
TBGm	X	assign counter for GCCm 0: TMG0 1: TMG1

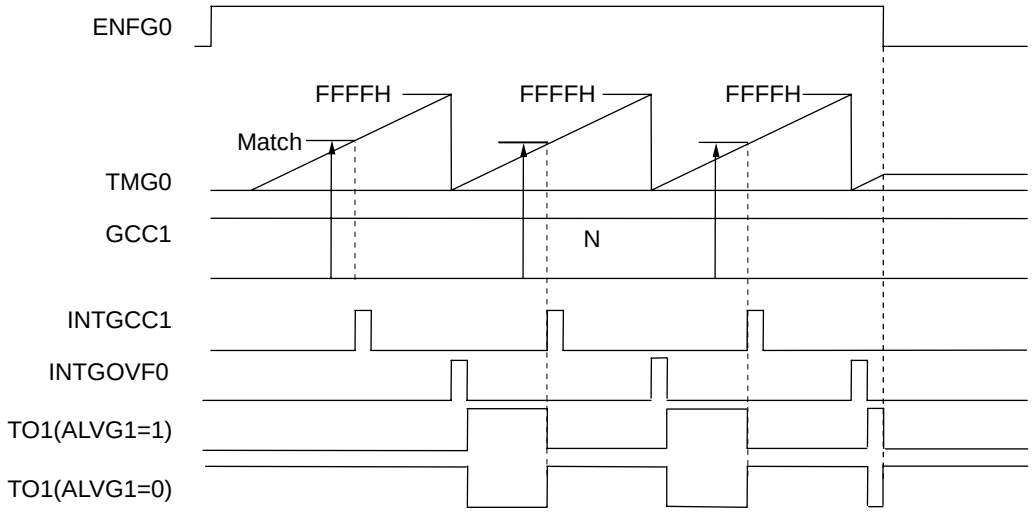
PWM setting method:

(1) An usable compare register is one of GCC1 to GCC4, and the corresponding counter must be selected with the TBGm bit.
(2) Select a count clock cycle with the CSE12 to CSE10 (TMG1) bits or CSE02 to CSE00 (TMG0) bits.
(3) Specify the active level of a timer output (TOGm) with the ALVGm bit.
(4) When using multiple timer outputs, the user can prevent TOGm from becoming active simultaneously by setting the OLDE bit of TMGMH register to provide step-by-step delays for TOGm. (This capability is useful for reducing noise and current.)
(5) Write data to GCCm.
(6) Start timer operation by setting POWER bit and TMG0E bit (or TMG1E bit).

PWM operation:

(1) When the value of the counter matches the value of GCCm, a match interrupt (INTGCCm) is output.
(2) When the counter overflows, an overflow interrupt (INTGOVF0 or INTGOVF1) is generated.
(3) TOGm does not make a transition until the first overflow occurs. (Even if the counter is cleared by software, TOGm does not make a transition until the next overflow occurs. After the first overflow occurs, TOGm is activated.
(4) When the value of the counter matches the value of GCCm, TOGm is deactivated, and a match interrupt (INTGCCm) is output. The counter is not cleared, but continues count-up operation.
(5) The counter overflows, and INTGOVF0 or INTGOVF1 is output to activate TOGm. The counter resumes count-up operation starting with 0000H.

Figure 6-18: Timing of PWM operation (free run)

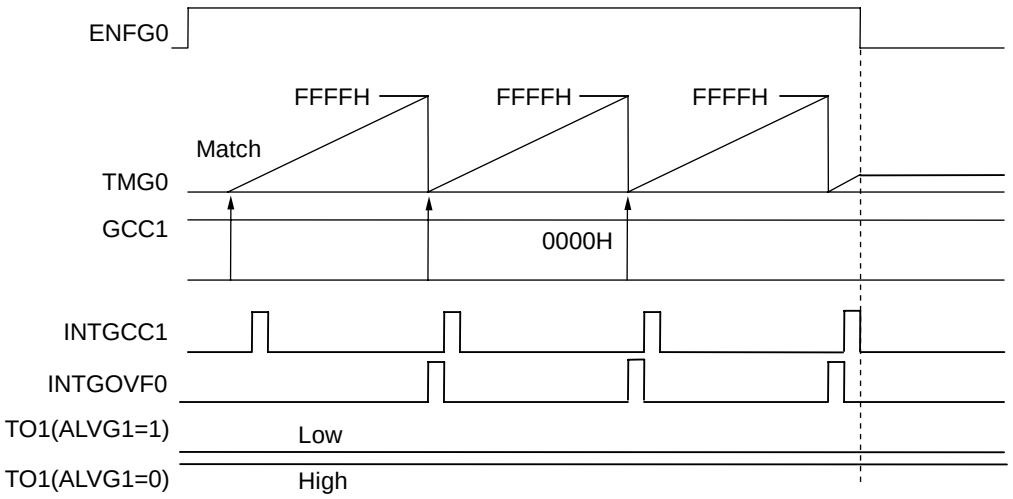


Data N is set in GCC1, counter TMG0 is selected

(a) When 0000H is set in GCCm (m = 0 to 4)

When 0000H is set in GCCm, TOGm is tied to the inactive level.
The figure below shows the state of TOG1 when 0000H is set in GCC1, and TMG0 is selected.

Figure 6-19: Timing when 0000H is set in GCCm (free run)



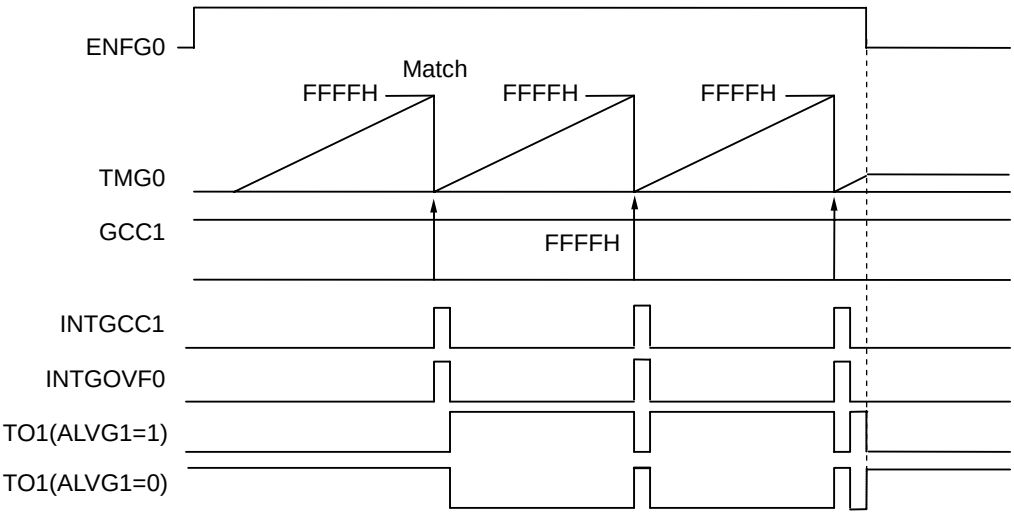
GCC1 and TMG0 are selected.

(b) When FFFFH is set in GCCm (m = 1 to 4)

When FFFFH is set in GCCm, TOGm outputs the inactive level for one clock period immediately after each counter overflow (except the first overflow).

The figure shows the state of TOG1 when FFFFH is set in GCC1, and TMG0 is selected.

Figure 6-20: Timing when FFFFH is set in GCCm (free run)



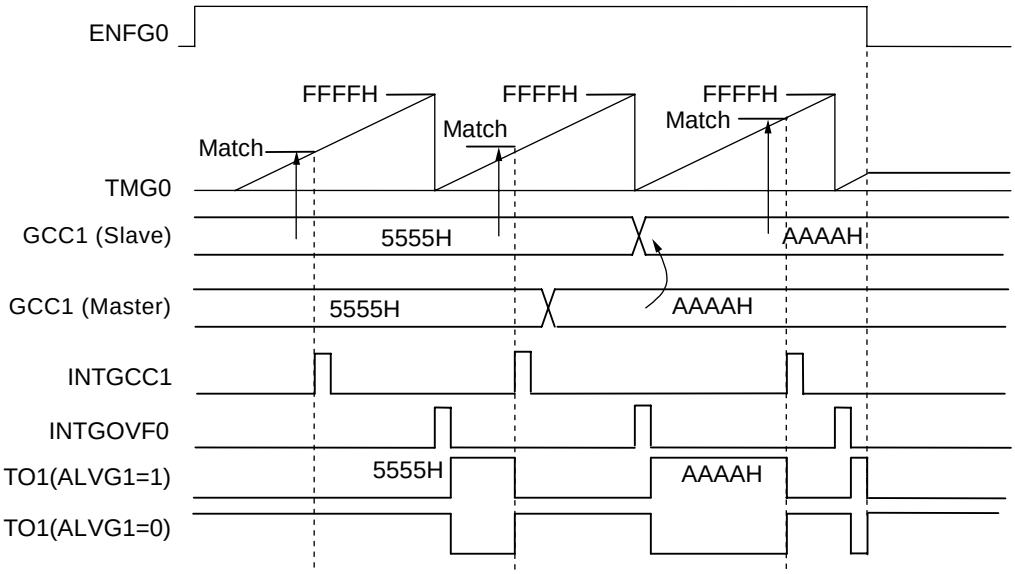
GCC1 and TMG0 are selected

(c) When GCCm is rewritten during operation (m = 1 to 4)

When GCC1 is rewritten from 5555H to AAAAH, the operation shown below is performed.

The figure below shows a case where TMG0 is selected for GCC1.

Figure 6-21: Timing when GCCm is rewritten during operation (free run)



GCC1 and TMG0 are selected.

If GCC1 is rewritten to AAAAH after the second INTGCC1 is generated as shown in the figure above, AAAAH is reloaded from the GCC1 master register to the GCC1 slave register when the next overflow occurs.
The next match interrupt (INTGCC1) is generated when the value of the counter is AAAAH. The pulse width also matches accordingly.

6.1.8 Match and clear mode

The match and clear mode is mainly used reduce the number of valid bits of the counters (TMG0, TMG1).
Therefore the fixed assigned register GCC0 (GCC1) is used to compare its value with the counter TMG0 (TMG1). If the values match, than an interrupt is generated and the counter is cleared. Than the counter starts up counting again.

(1) Capture operation (match ad clear)

Basic settings (m = 1 to 4):

Bit	Value	Remark
CCSG0	1	match and clear mode
CCSG5	1	
SWFGm	0	disable TOGm
CCSGm	0	Capture mode for GCCm
TBGm	X	assign counter for GCCm 0: TMG0 1: TMG1

(a) Example: Pulse width measurement or period measurement of the TIGm input signal

Setting method:

(1) When using one of TIG1 to TIG4-pin, select the corresponding counter with the TBGm bit. When CCSG0=1, TI0 cannot be used. When CCSG5=1, TIG5 cannot be used.
(2) Select a count clock cycle with the CSE12 to CSE10 (TMG1) bits or CSE02 to CSE00 (TMG0) bits.
(3) Select a valid TIGm edge with the IEGm1 and IEGm0 bit. A rising edge, falling edge, or both edges can be selected.
(4) Set an upper limit on the value of the counter in GCC0 or GCC5.
(5) Start timer operation by setting POWER bit and TMG0E bit (or TMG1E bit).

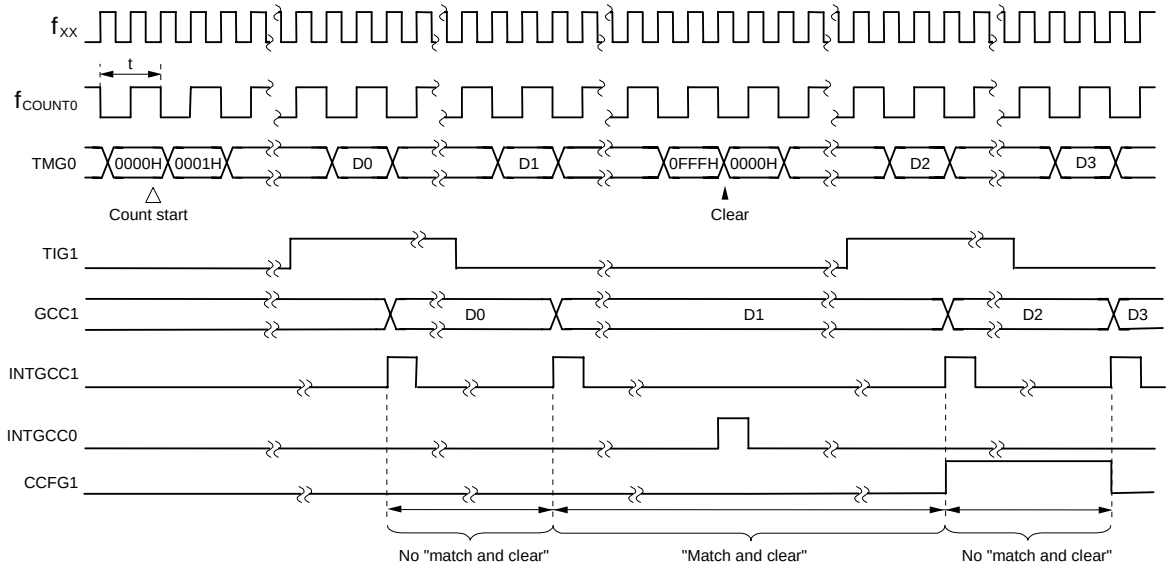
Operation:

(1) When a specified edge is detected, the value of the counter is stored in GCCm, and an edge detection interrupt (INTGCCm) is output.
(2) When the value of GCC0 or GCC5 matches the value of the counter, INTGCC0 (INTGCC5) is output, and the counter is cleared. This operation is referred to as “match and clear”.
(3) If a match and clear event has occurred between capture operations, the CCFGn flag is set when GCCn is read. Correct capture data by checking the value of CCFGn.

(b) Example: Capture where both edges of TIGm are valid (match and clear)

For the timing chart TMG0 is selected as the counter corresponding to TIG1, and 0FFFH is set in GCC0.

Figure 6-22: Timing when both edges of TIGm are valid (match and clear)



Remark: The figure above shows an image. In actual circuitry, 3 to 4 periods of the count-up signal (f_{COUNT}) are required from the input of a waveform to TIG1 until a capture interrupt is output. (See Figure 6-14, “Timing of capture trigger edge detection (free run),” on page 177.)

Caution: If two or more match and clear events occur between captures, a software-based measure needs to be taken to count INTGCC0 or INTGCC5.

(c) When 0000H is set in GCC0 or GCC5 (match and clear)

When 0000H is set in GCC0 (GCC5), the value of the counter is fixed at 0000H, and does not operate. Moreover, INTGCC0 (INTGCC5) continues to be active.

(d) When FFFFH is set in GCC0 or GCC5 (match and clear)

When FFFFH is set in GCC0 (GCC5), operation equivalent to the free-run mode is performed. When an overflow occurs, INTGOVF0 (INTGOVF1) is generated, but INTGCC0 (INTGCC5) is not generated.

(2) Compare operation (match and clear)

Basic settings (m = 1 to 4):

Bit	Value	Remark
CCSG0	1	match and clear mode
CCSG5	1	
SWFGm	0	disable TOGm
CCSGm	1	Compare mode for GCCm
TBGm	X	assign counter for GCCm 0: TMG0 1: TMG1

(a) Example: Interval timer (match and clear)

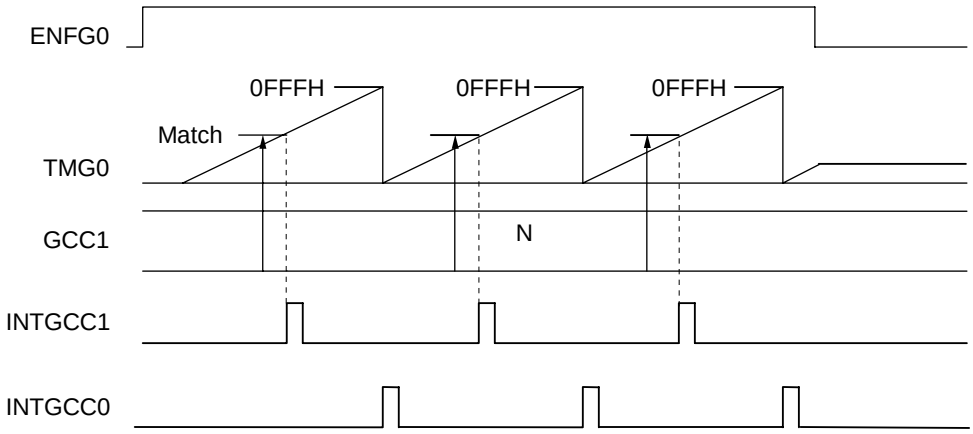
Setting Method

(1) An usable compare register is one of GCC1 to GCC4, and the corresponding counter must be selected with the TBGm bit.
(2) Select a count clock cycle with the CSE12 to CSE10 bits (TMG1) or CSE02 to CSE00 bits (TMG0).
(3) Set an upper limit on the value of the counter in GCC0 or GCC5.
(4) Write data to GCCm.
(5) Start timer operation by setting the POWER and TMGnE bit (n = 0, 1).

Operation:

(1) When the value of the counter matches the value of GCCm, a match interrupt (INTGCCm) is output.
(2) When the value of GCC0 or GCC5 matches the value of the counter, INTGCC0 (or INTGCC5) is output, and the counter is cleared. This operation is referred to as “match and clear”.
(3) The counter resumes count-up operation starting with 0000H.

Figure 6-23: Timing of compare operation (match and clear)



In this example, the data N is set in GCC1, and TMG0 is selected.
0FFFH is set in GCC0. Here, N < 0FFFH.

(b) When 0000H is set in GCC0 or GCC5 (match and clear)

When 0000H is set in GCC0 or GCC5, the value of the counter is fixed at 0000H, and does not operate. Moreover, INTGCC0 (or INTGCC5) continues to be active.

(c) When FFFFH is set in GCC0 or GCC5 (match and clear)

When FFFFH is set in GCC0 or GCC5, operation equivalent to the free-run mode is performed. When an overflow occurs, INTGOVF0 (or INTGOVF1) is generated, but INTGCC0 (or INTGCC5) is not generated.

(d) When 0000H is set in GCCm (m = 1 to 4) (match and clear)

INTGCCm is activated when the value of the counter becomes 0001H.
Note, however, that even if no data is set in GCCm, INTGCCm is activated immediately after the counter starts.

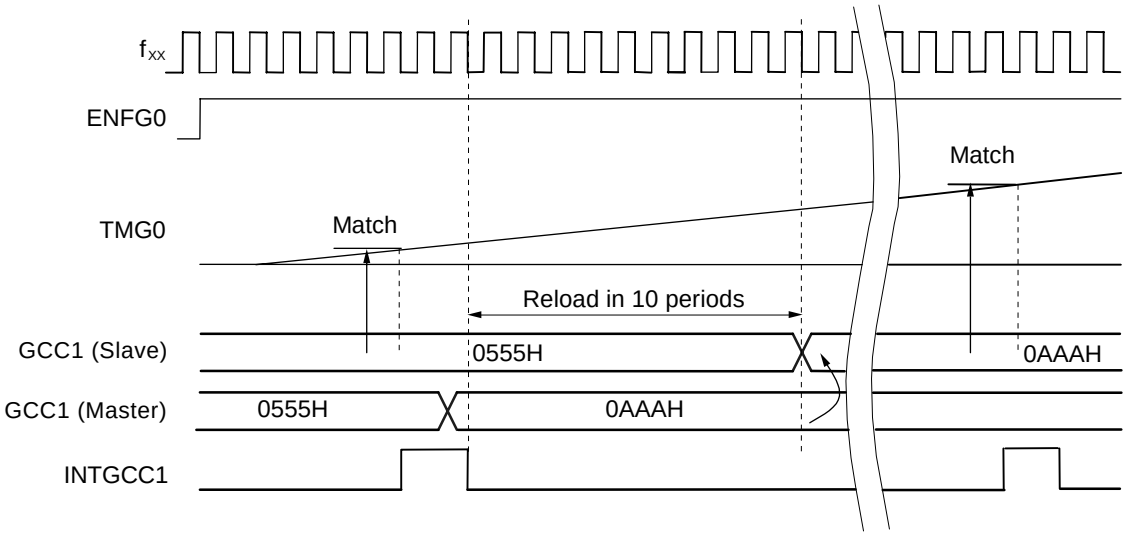
(e) When a value exceeding the value of GCC0 or GCC5 is set in GCCm (m = 1 to 4) (match and clear)

INTGCCm is not generated.

(f) When GCCm (m = 1 to 4) is rewritten during operation (match and clear)

When the value of GCC1 is changed from 0555H to 0AAAH, the operation described below is performed.
TMG0 is selected as the counter, and 0FFFH is set in GCC0.

Figure 6-24: Timing when GCCm is rewritten during operation (match and clear)



The GCC1 compare value, that is written in the GCC1 master register is transferred to the GCC1 slave register 5 clocks later

Caution: To perform successive write access during operation, for rewriting the GCCn register (n = 1 to 4), you have to wait for minimum 14 system clocks periods (f_{XX}).

(3) PMW output (match and clear)

Basic settings (m = 1 to 4):

Bit	Value	Remark
CCSG0	1	match and clear mode
CCSG5	1	
SWFGm	1	enable TOGm
CCSGm	1	Compare mode for GCCm
TBGm	X	assign counter for GCCm 0: TMG0 1: TMG1

Setting Method:

(1) An usable compare register is one of GCC1 to GCC4, and the corresponding counters TMG0 or TMG1 must be selected with the TBGm bit (m = 1 to 4).
(2) Select a count clock cycle with the CSE12 to CSE10 (TMG1) bits or CSE02 to CSE00 (TMG0) bits.
(3) Specify the active level of a timer output (TOGm) with the ALVGm bit.
(4) When using multiple timer outputs, the user can prevent TOGm from making transitions simultaneously by setting the OLDE bit of TMGMH register. (This capability is useful for reducing noise and current.)
(5) Set an upper limit on the value of the counter in GCC0 or GCC5. (0x0000 is forbidden)
(6) Write data to GCCm.
(7) Start count operation by setting POWER bit and TMG0E bit (or TMG1E bit).

Operation of PWM (match and clear):

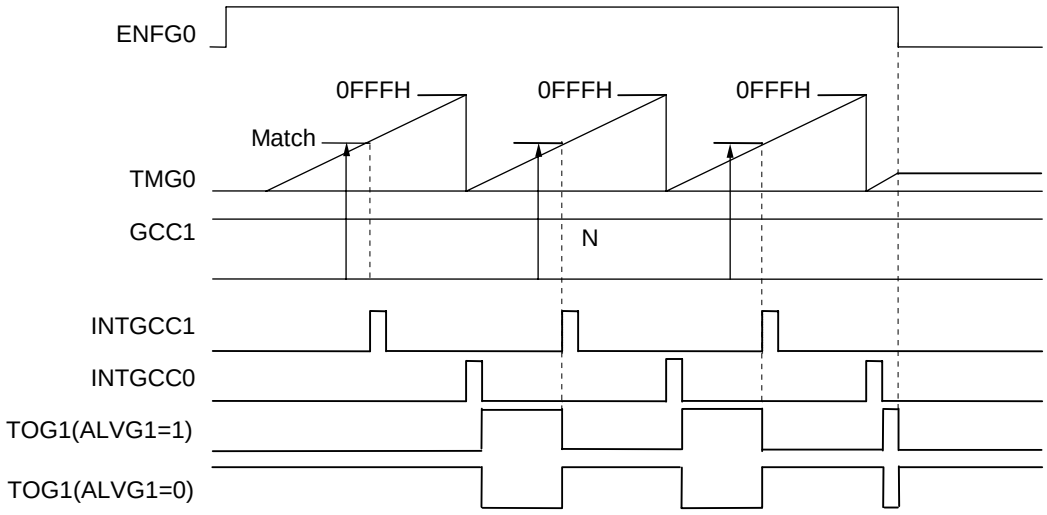
(1) When the value of the counter matches the value of GCCm, a match interrupt (INTGCCm) is output.

Caution: Do not set 0000H in GCC0 or GCC5 in match and clear mode.

(2) When the value of GCC0 (GCC5) matches the value of the counter, INTGCC0 (INTGCC5) is output, and the counter is cleared. This operation is referred to as “match and clear”.
(3) TOGm does not make a transition until the first match and clear event.
(4) TOGm makes a transition to the active level after the first match and clear event.
(5) When the value of the counter matches the value of GCCm, TOGm makes a transition to the inactive level, and a match interrupt (INTGCCm) is output.
(6) When the next match and clear event occurs, INTGCC0 (INTGCC5) is output, and the counter is cleared. The counter resumes count-up operation starting with 0000H.

Example where the data N is set, and the counter TMG0 is selected.
0FFFH is set in GCC0 and N < 0FFFH.

Figure 6-25: Timing of PWM operation (match and clear)



When 0000H is set in GCC0 (GCC5), the value of the counter is fixed at 0000H, and the counter does not operate. The waveform of INTGCC0 (INTGCC5) varies, depending on whether the count clock is the reference clock or the sampling clock.

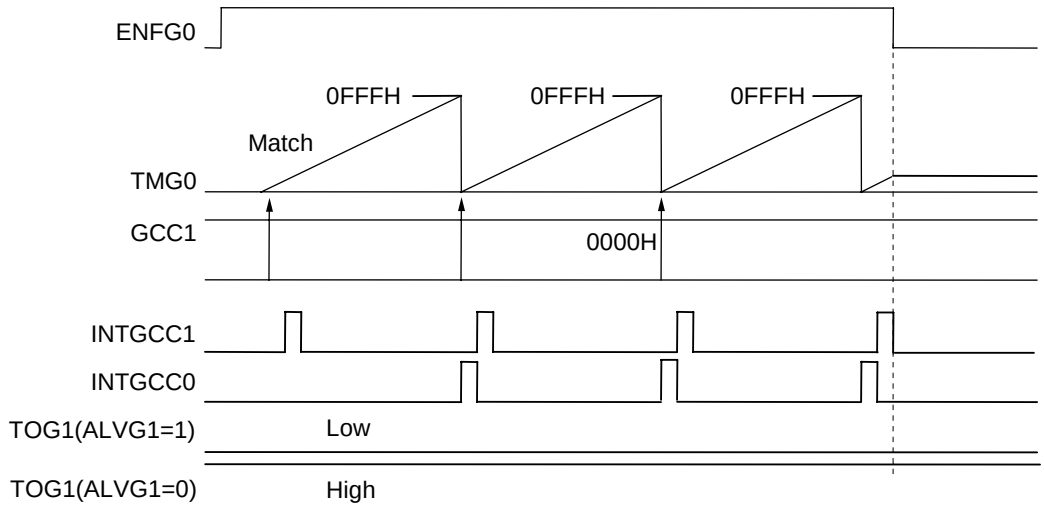
(a) When FFFFH is set in GCC0 or GCC5 (match and clear)

When FFFFH is set in GCC0 (GCC5), operation equivalent to the free-run mode is performed. When an overflow occurs, INTGOVF0 (INTGOVF1) is generated, but INTGCC0 (INTGCC5) is not generated.

(b) When 0000H is set in GCCm (match and clear)

When 0000H is set in GCCm, TOGm is tied to the inactive level.
The figure below shows the state of TOG1 when 0000H is set in GCC1, and TMG0 is selected. Note, however, that 0FFFH is set in GCC0.

Figure 6-26: Timing when 0000H is set in GCCm (match and clear)

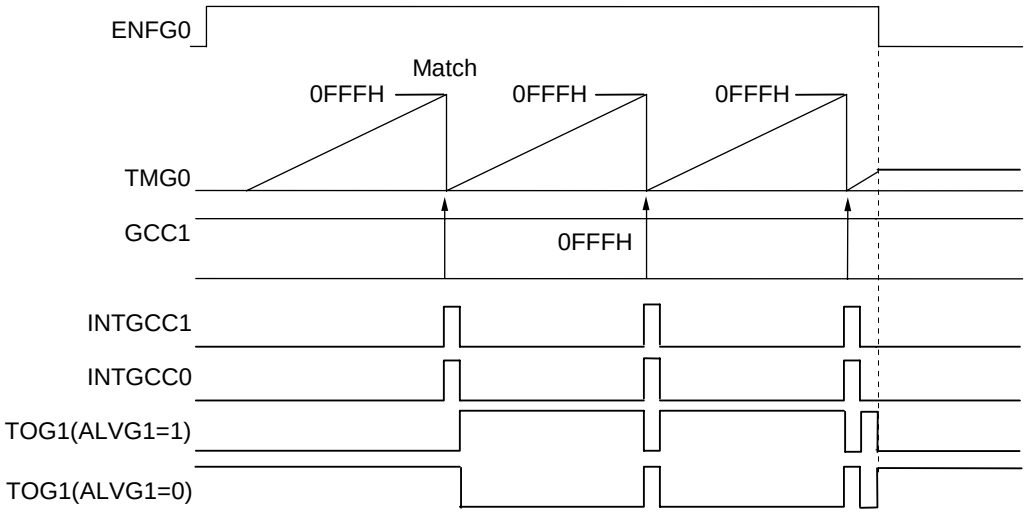


(c) When the same value as set in GCC0 or GCC5 is set in GCCm (match and clear)

When the same value as set in GCC0 (GCC5) is set in GCCm, TOGm outputs the inactive level for only one clock period immediately after each match and clear event (excluding the first match and clear event).

The figure below shows the state of TOG1 when 0FFFH is set in GCC0 and GCC1, and TMG0 is selected.

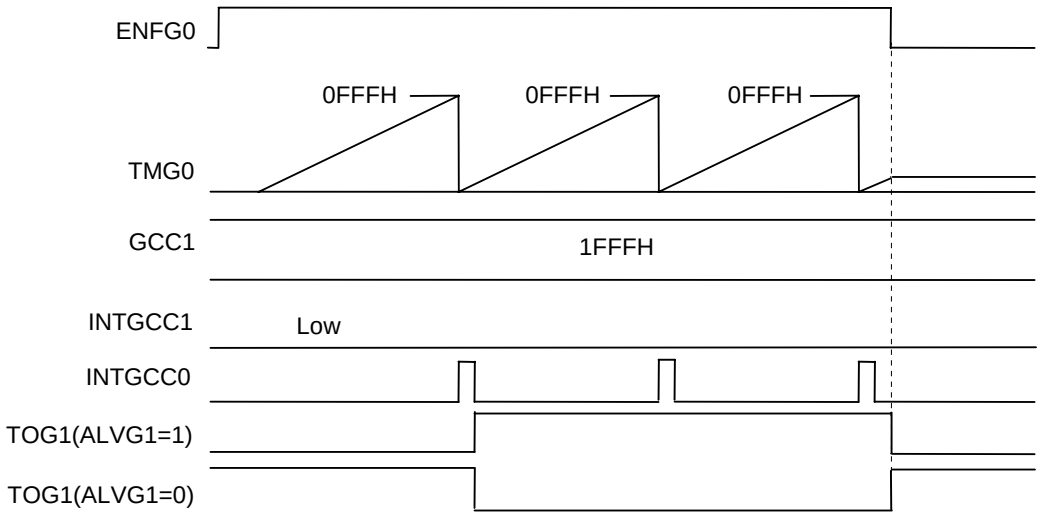
Figure 6-27: Timing when the same value as set in GCC0/GCC5 is set in GCCm (match and clear)



(d) When a value exceeding the value set in GCC0 or GCC5 is set in GCCm (match and clear)

When a value exceeding the value set in GCC0 (GCC5) is set in GCCm, TOGm starts and continues outputting the active level immediately after the first match and clear event (until count operation stops.)
The figure shows the state of TOG1 when 0FFFH is set in GCC0, 1FFFH is set in GCC1, and TMG0 is selected.

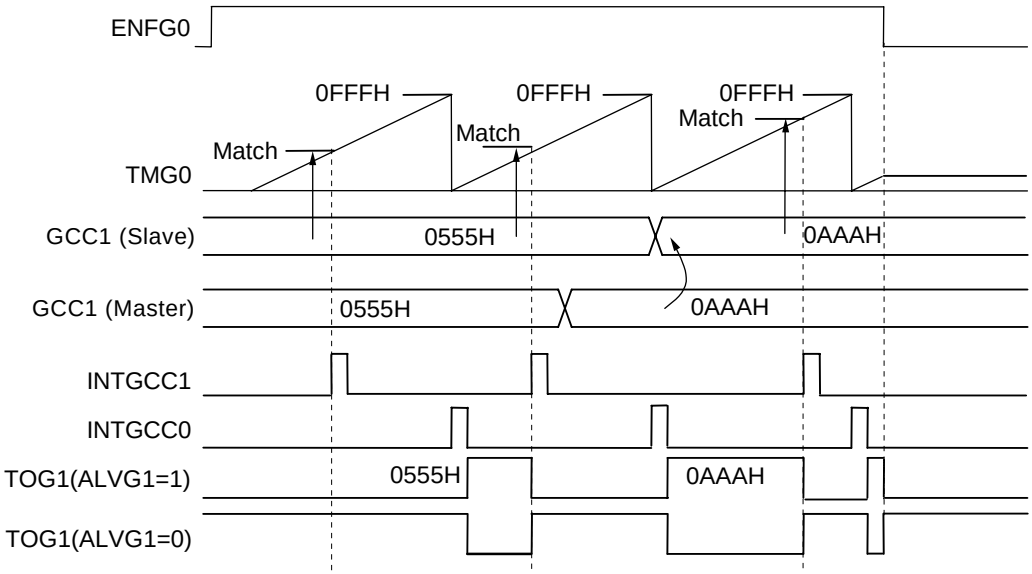
Figure 6-28: Timing when the value of GCCm exceeding GCC0 or GCC5 (match and clear)



(e) When GCCm is rewritten during operation (match and clear)

When GCC1 is rewritten from 0555H to 0AAAH, the operation shown below is performed.
The figure below shows a case where 0FFFH is set in GCC0, and TMG0 is selected for GCC1.

Figure 6-29: Timing when GCCm is rewritten during operation (match and clear)



If GCC1 is rewritten to 0AAAH after the second INTGCC1 is generated as shown in the figure above, 0AAAH is transferred from the GCC1 master to the GCC1 slave register when the next match and clear occurs.
The next match interrupt (INTGCC1) is generated when the value of the counter is 0AAAH. The pulse width also matches accordingly.

6.1.9 Edge detection noise elimination

The edge detection circuit has a noise elimination function. This function regards:

- a pulse **not wider than one count clock** period as a **noise**, and does not detect it as an edge.
- a pulse **not shorter than two count clock** periods is detected normally as an **edge**.
- a pulse **wider than one count clock period but shorter than two count clock** periods may be **detected as an edge or may be eliminated as noise**, depending on the timing.

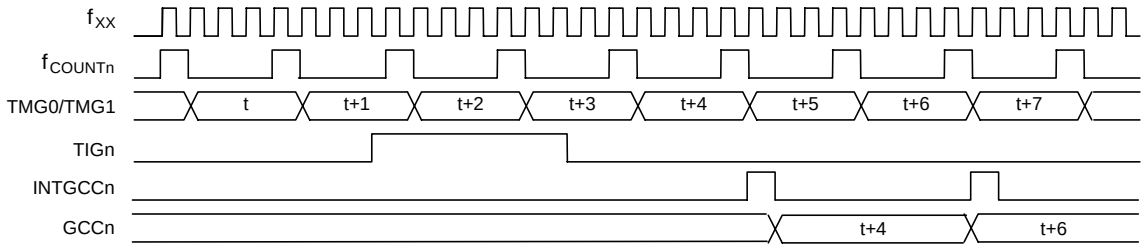
(This is because the count-up signal of the counter is used for sampling timing.) The upper figure below shows the timing chart for performing edge detection. The lower figure below shows the timing chart for not performing edge detection.

Basic settings (m = 1 to 4):

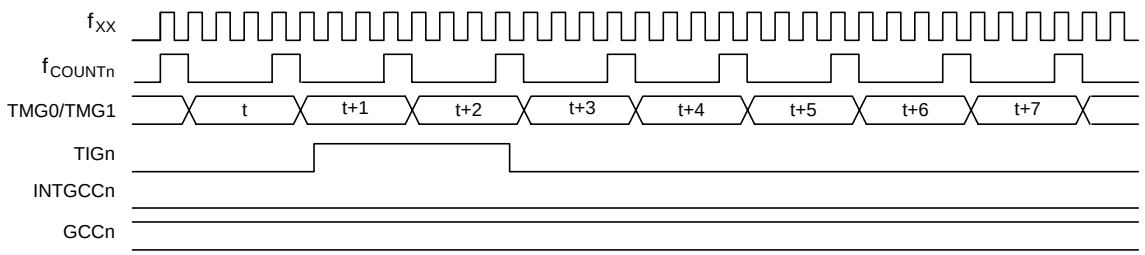
Bit	Value	Remark
CSEn2	0	Count clock = $f_{XX}/4$
CSEn1	0	
CSEn0	1	
IEGn1	1	detection of both edges
IEGn0	1	

Figure 6-30: Timing of Edge detection noise elimination

<Timing chart for performing edge detection>



<Timing chart for noise elimination>



6.1.10 Precautions Timer G

(1) When POWER bit of TMGMH register is set

The rewriting of the CSEn2 to CSEn0 bits (n = 0, 1) of TMGMH register is prohibited.
This bits sets the prescaler for the Timer G counter.

The rewriting of the CCSGn bits (n = 0 to 5) are prohibited.
This bits (OCTLGL and OCTLGH register) set the capture or the compare mode to the GCCn register. For the GCC0 and the GCC5 register this bits (TMGML register) set the “free run” or “match and clear” mode of the TMG0 and TMG1 counter.

The rewriting of the TMGCML and the TMGCMH register is prohibited.
This registers configured the counter (TMG0 or TMG1) for the GCCm register (m = 1 to 4) and defines the edge detection for the TIGm input pins (falling, rising, both).

Even when POWER bit is set, TOGm output is switched by switching the ALVGm bit of OCTLGL and OCTLGH register.
These bits configured the active level of the TOGm pins (m = 1 to 4).

(2) When POWER and TMGnE bit is set (n = 0, 1)

The rewriting of ALVGm is prohibited (m = 1 to 4)
These bits configured the active level of the TOGm pins (m = 1 to 4).

When in compare-mode the rewriting of the GCC0 or GCC5 register is prohibited.
In compare mode this registers set the value for the “match and clear” mode of the TMG0 and TMG1 counter.

(3) Functionality

When the POWER bit is set to “0”, regardless of the SWFGm bit (OCTLGL and OCTLGH registers), the TOGm pins are tied to the inactive level.
The SWFGm bit enables or disables the output of the TOGm pins. This bit can be rewritten during timer operation.

The CLRGn bit (n = 0, 1) is a flag. If this bit is read, a 0 is read at all times.
This bit clears the corresponding counter (TMG0 or TMG1)

When GCCm register (m = 1 to 4) are used in capture operation:
If two or more overflows of TMG0 or TMG1 occur between captures, a software-based measure needs to be taken to count overflow interrupts (INTGOVF0 or INTGOVF1).
If only one overflow is necessary, the CCFGn bit (n = 0 to 5) can be used for overflow detection

Only the overflow of the TMG0 or TMG1-counter clears the CCFGn bit (TMGST). The software-based clearing via CLRG0 or CLRG1 bit (TMGML register) doesn’t affect this bits.
The CCFGn bit is set if a TMG0 (TMG1) overflow occurs. This flag is only updated if the corresponding GCCn register was read, so first read the GCCn register and then read this flag if necessary.

(4) Timing

The delay of each timer output TOGm (m = 1 to 4) varies according to the setting of the count clock with the CSEx2 to CSEx0 bits (x= 0, 1).

In capture operation 3 to 4 periods of the count-clock (f_{COUNT}) signal are required from the TIGn-pin (n = 0 to 5) until a capture interrupt is output.

when TMGxE (x = 0, 1) is set earlier or simultaneously with POWER bit, than the Timer G needs 14 system clocks periods (f_{XX}) to start counting.
when TMGxE (x = 0, 1) is set later than POWER bit, than the Timer G needs 8 system clocks periods (f_{XX}) to start counting.

When a capture register (GCCn) is read, the capturing is disable during read operation. This is intended to prevent undefined data during reading. So, if a contention occurs between an external trigger signal and the read operation, capture operation may be cancelled, and old data may be read.

GCCm register (m = 1 to 4) in Compare mode:
After setting the POWER bit you have to wait for 10 system clocks periods (f_{XX}) to perform write access to the GCCm register (m = 1 to 4).
To perform successive write access during operation, for rewriting the GCCm register (n = 1 to 4), you have to wait for minimum 14 system clocks periods (f_{XX}).

6.2 Timer C

6.2.1 Features (Timer C)

Two channels of Timer C are implemented. Timer C (TMCn) is a 16-bit timer/counter that can perform the following operations.

- 2 capture/compare register
- Programmable pulse generator function
- Interval timer function
- PWM output
- External signal cycle measurement
- Support for CAN Time Stamp Function (Timer C0)

6.2.2 Function overview (Timer C)

- 16-bit timer/counter (TMCn): 1 channel
- Capture/compare registers: 2
- Count clock division selectable by prescaler (maximum frequency of count clock: 8 MHz)
- Prescaler divide ratio from $f_{XX}/2$ to $f_{XX}/256$
- Interrupt request sources
  - Capture/compare match interrupt requests: 2 sources

    In case of capture register:
    - INTCCC0n generated by TIC0n input
    - INTCCC1n generated by TIC1n input

    In case of compare register:
    - INTCCC0n generated by CCC0n match signal
    - INTCCC1n generated by CCC1n match signal
- Overflow interrupt request: 1 source

  INTCOVFn generated upon overflow of TMCn register
- Timer/counter count clock sources: 1 type

  (internal system clock cycle)
- One of two operation modes when the timer/counter overflows can be selected: free-running mode or overflow-stop mode
- The timer/counter can be cleared by match of timer/counter and compare register
- External pulse output (TOCn): 1

Remark: n = 0, 1

Figure 6-31, “Block Diagram of Timer C0,” on page 201 shows the block diagram of Timer C0.

Figure 6-32, “Block Diagram of Timer C1,” on page 201 shows the block diagram of Timer C1.

Figure 6-31: Block Diagram of Timer C0

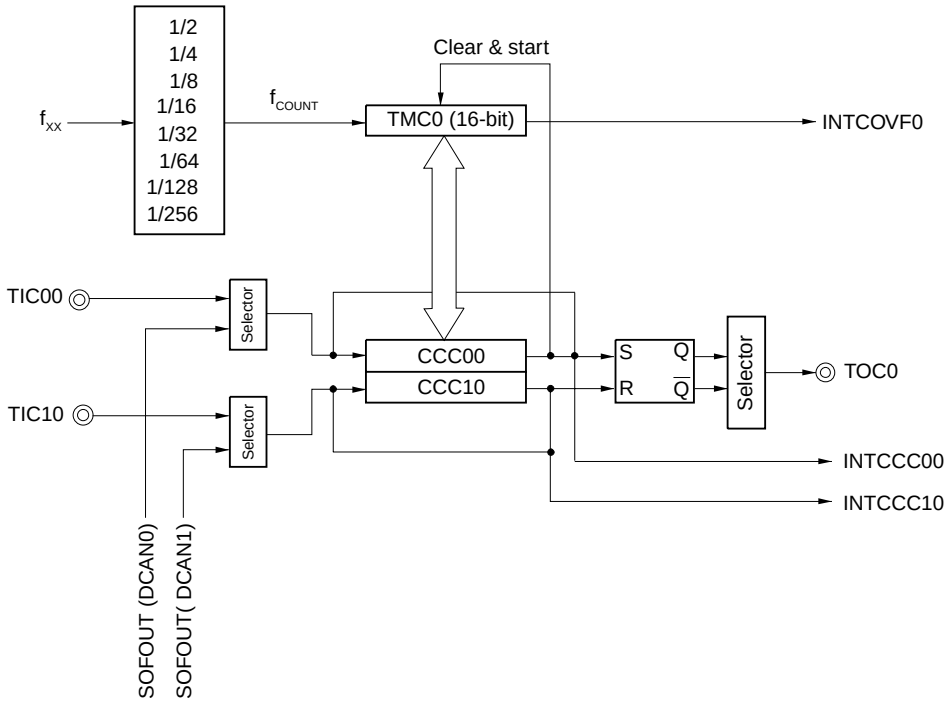
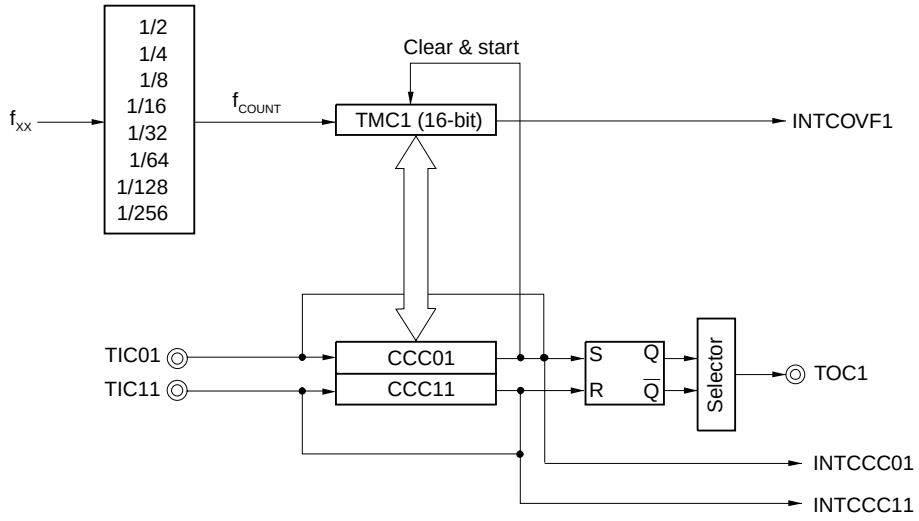


Figure 6-32: Block Diagram of Timer C1



Remark: f_{CLK}: Base clock (16 MHz)

6.2.3 Basic configuration

Table 6-2: Timer C Configuration List

Timer	Count Clock	Register	Read/ Write	Generated Interrupt Signal	Capture Trigger	Timer Output S/R
Timer C	$f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$,$f_{XX}/16$ $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$	TMCn	Read	INTCOVFn	-	-
		CCC0n	Read/write	INTCCC0n	INTCCC0n	TOCn (S)
		CCC1n	Read/write	INTCCC1n	INTCCC1n	TOCn (R)

Remarks: 1. f_{XX}: Internal system clock

2. S/R: Set/Reset

3. n = 0,1

(1) 16-bit counter (TMCn)

TMCn functions as a 16-bit free-running timer or as an event counter for an external signal. Besides being mainly used for cycle measurement, Timer C can be used as pulse output. Furthermore the instance TMC0 can be used for the “Time Stamp Function” of the DCAN-Interface.

TMCn is a 16-bit units read-only register.

Caution: Continuous reading of TMCn is prohibited. If TMCn is continuously read, the second read value may differ from the actual value.

Figure 6-33: Timer C counter (TMC0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
TMC0																	FFFF F260H	0000H
TMC1																	FFFF F280H	0000H

Remarks: 1. The TMC0 (TMC1) register can only be read. If writing is performed to the TMC0 (TMC1) register, the subsequent operation is undefined.

2. If the CAE bit of the TMCC0n register is cleared to “0”, a reset is performed asynchronously.

TMCn performs the count-up operations of an internal count clock or external count clock. Timer starting and stopping are controlled by the CE bit of Timer C control register 0 (TMCC0n).

Selection of the internal count clock

TMCn operates as a free-running timer.
When an internal clock is specified as a count clock by Timer C control register 1 (TMCC1n), TMCn is counted up for each input clock cycle specified by the CS2 to CS0 bits of the TMCC0n register.
A division by the prescaler can be selected for the count clock from among $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$ and $f_{XX}/256$ by the TMCC0n register (f_{XX}: internal system clock).
An overflow interrupt can be generated if the timer overflows.

Caution: The count clock cannot be changed while the timer is operating.

The conditions when the TMCn register becomes 0000H are:

(a) Asynchronous reset

- CAEn bit of TMCC0n register = 0
- Reset input

(b) Synchronous reset

- CE bit of TMCC0n register = 0
- The CCC0n register is used as a compare register, and the TMCn and CCC0n registers match when “clearing the TMCn register” is enabled (CCLR bit of the TMCC1n register = 1)

(2) Capture/compare registers (CCC0n and CCC1n)

These capture/compare registers are 16-bit registers.
They can be used as capture registers or compare registers according to the CMS1 and CMS0 bit specifications of Timer C control register 1 (TMCC1n).

These registers can be read/written in 16-bit units (However, write operations can only be performed in compare mode).

Figure 6-34: Capture/Compare Register 0 of TMCn (CCC0n) (n = 0, 1)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
CCC00																	FFFF F264H	0000H
CCC01																	FFFF F284H	0000H

Figure 6-35: Capture/Compare Register 1 of TMC1 (CCC1n) (n = 0, 1)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
CCC10																	FFFF F268H	0000H
CCC11																	FFFF F288H	0000H

(a) Setting CCCnm registers to capture registers (CMS1 and CMS0 of TMCC1n = 0)

When these registers are set to capture registers, the valid edges of the corresponding external interrupt signals TICnm (n, m = 0, 1) are detected as capture triggers. The timer TMCn is synchronized with the capture trigger, and the value of TMCn is latched in the CCC0n and CCC1n registers (capture operation).

The valid edge of the TIC0n (n = 0,1) pins is specified (rising, falling, or both edges) according to the IES01 and IES00 bits of the SESCn register,
The valid edge of the TIC1n (n = 0,1) pin is specified according to the IES11n and IES10n bits of the SESCn register.

The capture operation is performed asynchronously relative to the count clock. The latched value is held in the capture register until the next time the capture operation is performed.
When the CAE bit of Timer C control register 0 (TMCC0n) is 0, 0000H is read.

If the CCC0n (CCC1n) registers are specified as capture registers, an interrupt is generated (INTCCC0n and INTCCC1n) by detecting the valid edge of signals.

Caution: If the capture operation and the TMCn register count prohibit setting (CE bit of TMCC0n register = 0) timings conflict, the captured data becomes undefined, and no INTCCC0n interrupt is generated (n = 0, 1).

(b) Setting CCCnm registers to compare registers (CMS1 and CMS0 of TMCC1n = 1)

When these registers are set to compare registers, the TMCn and register values are compared for each timer count clock, and an interrupt is generated by a match.
If the CCLR bit of Timer C control register 1 (TMCC1n) is set (1), the TMCn value is cleared (0) at the same time as a match with the CCC0n register (it is not cleared (0) by a match with the CCC1n register).

A compare register is equipped with a set/reset output function. The corresponding timer output (TOCn) is set or reset, synchronized with the generation of a match signal.

The interrupt selection source differs according to the function of the selected register.

Cautions: 1. To write to capture/compare registers 0 and 1 (CCC0n, CCC1n), always set the CAEn bit to 1 first. When the CAEn bit is 0, even if writing to registers CCC0n and CCC1n, the data that is written will be invalid because the reset is asynchronous.

2. Perform a write operation to capture/compare registers 0 and 1 after setting them to compare registers according to the TMCC1n register setting. If they are set to capture registers (CMS1 and CMS0 bits of TMCC1n register = 0), no data is written even if a write operation is performed to CCC0n and CCC1n.

3. When these registers are set to compare registers, TIC0n and TIC1n cannot be used as external interrupt input pins.

6.2.4 Control registers

(1) Timer C control register 0 (TMCC00, TMCC01)

The TMCC0n register controls the operation of TMCn.

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

Caution: The CAE bit and CE bit cannot be set at the same time. Be sure to set the CAE bit prior to setting the CE bit. To use an external pin related to the timer function when using Timer C, be sure to set the CAE bit to "1" after setting the external pin to the control mode.

Figure 6-36: Timer C control Register 0 (TMCC0n) (n = 0, 1) (1/2)

	7	6	5	4	3	2	1	0	Address	Initial value
TMCC00	OVF	CS2	CS1	CS0	0	0	CE	CAE	FFFF F26CH	00H
TMCC01	OVF	CS2	CS1	CS0	0	0	CE	CAE	FFFF F28CH	00H

Bit Position	Bit Name	Function
7	OVF	Flag that indicates TMCn overflow. 0: No overflow 1: Overflow The OVF bit becomes "1" when TMCn changes from FFFFH to 0000H. An overflow interrupt request (INTCOVFn) is generated at the same time. However, if CCC0n is set to the compare mode (CMS0 bit of the TMCC1n register = 1) and match clear during comparison of TMCn and CCC0n is enabled (CCLR bit of TMCC1n register = 1), and TMCn is cleared to 0000H following match at FFFFH, TMCn is considered to have been cleared and the OVF bit does not become"1", nor is the INTCOVFn interrupt generated. The OVF bit holds a "1" until "0" is written to it or an asynchronous reset is applied while the CAEn bit = 0. Interrupts by overflow and the OVF bit are independent, and even if the OVF bit is manipulated, this does not affect the interrupt request flag for INTCOVFn (TMCIF0). If an overflow occurs while the OVF bit is being read, the value of the flag changes and the value is returned at the next read.

Figure 6-36: Timer C control Register 0 (TMCC0n) (n = 0, 1) (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>6 to 4</td><td>CS2 to CS0</td><td>Selects the internal count clock for TMCn.
<table>
<tr><th>CS2</th><th>CS1</th><th>CS0</th><th>Count Clock</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>$f_{XX}/2$</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>$f_{XX}/4$</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>$f_{XX}/8$</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>$f_{XX}/16$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>$f_{XX}/32$</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>$f_{XX}/64$</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>$f_{XX}/128$</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>$f_{XX}/256$</td></tr>
</table>
Caution: Do not change the CS2 to CS0 bits during timer operation. If they are to be changed, they must be changed after setting the CE bit to “0”. If the CS2 to CS0 bits are overwritten during timer operation, the operation is not guaranteed.
Remark: f_{XX}: system-clock (16 MHz)</td></tr>
<tr><td>1</td><td>CE</td><td>Controls the operation of TMCn.
0: Disable count (timer stopped at 0000H and does not operate)
1: Perform count operation.
Caution: If CE = 0, the external pulse output (TOCn) becomes inactive level (The active level of TOCn output is set with the ALV bit of the TMCC1n register).</td></tr>
<tr><td>0</td><td>CAE</td><td>Controls the internal count clock (f_{COUNT}).
0: Asynchronously reset entire TMCn unit. Stop base clock supply to TMCn unit.
1: Supply system-clock (f_{XX}) to TMCn unit.
Cautions: 1. When CAE = 0 is set, the TMCn unit can be reset asynchronously.
2. When CAE = 0, the TMCn unit is in a reset state. To operate TMCn, first set CAE = 1.
3. When the CAE bit is changed from “1” to “0”, all the registers of the TMCn unit are initialized. When again setting CAE = 1, be sure to then again set all the registers of the TMCn unit.</td></tr>
</table>

(2) Timer C control register 1 (TMCC1n) (n = 0, 1)

The TMCC1n register controls the operation of TMCn.

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

Cautions: 1. Do not change the bits of the TMCC1n register during timer operation. If they are to be changed, they must be changed after setting the CE bit of the TMCC0n register to 0. If the TMCC1n register is overwritten during timer operation, the operation is not guaranteed.

2. If the ENTO bit and the ALV bit are changed simultaneously, a glitch (spike-shaped noise) may be generated in the TOCn pin output. Either design the circuit that will not malfunction even if a glitch is generated, or make sure that the ENTO bit and the ALV bit do not change at the same time.

3. TOCn output remains unchanged by external interrupt signals (INTCCC0n, INTCCC1n). When using the TOCn signal, set the capture/compare register to the compare register (CMS1, CMS0 bits of TMCC1n register = 1).

Remark: A reset takes precedence for the flip-flop of the TOCn output.

Figure 6-37: Timer C control Register 1 (TMCC1n) (n = 0, 1) (1/2))

	7	6	5	4	3	2	1	0	Address	Initial value
TMCC10	0	ENTO	ALV	0	CLR	0	CMS1	CMS0	FFFF F270H	20H
TMCC11	0	ENTO	ALV	0	CLR	0	CMS1	CMS0	FFFF F290H	20H

Figure 6-37: Timer C control Register 1 (TMCC1n) (n = 0, 1) (2/2)

Bit Position	Bit name	Function
6	ETO	Enables/disables output of external pulse output (TOCn). 0: Disable external pulse output. Output of inactive level of ALV bit to TOCn pin is fixed. TOCn pin level remains unchanged even if match signal from corresponding compare register is generated. 1: Enable external pulse output. Compare register match causes TOCn output to change. However, in capture mode, TOCn output does not change. An ALV bit inactive level is output from the time when timer output is enabled until a match signal is generated. **Caution: If either CCC0n or CCC1n is specified as a capture register, the ENTO bit must be set to "0".**
5	ALV	Specifies active level of external pulse output (TOCn). 0: Active level is low level. 1: Active level is high level. **Caution: The initial value of the ALV bit is "1".**
3	CLR	Enables/disables TMCn clearing during compare operation. 0: Disable clearing. 1: Enable clearing (TMCn is cleared when CCC0n and TMCn match during compare operation).
1	CMS1	Selects operation mode of capture/compare register (CCC1n). 0: Register operates as capture register. 1: Register operates as compare register.
0	CMS0	Selects operation mode of capture/compare register (CCC0n). 0: Register operates as capture register. 1: Register operates as compare register.

(3) Valid edge selection register (SESCn)

This register specifies the valid edge of external interrupt requests from an external TICmn-pin (n, m = 0 to 1).
The rising edge, the falling edge, or both rising and falling edges can be specified as the valid edge independently for each pin.

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

Caution: Do not change the bits of SESCn register during timer operation. If they have to be changed, they must be changed after setting the CE bit of the TMCC0n register to "0". If the SESCn register is overwritten during timer operation, the operation is not guaranteed.

Figure 6-38: Valid Edge Selection Register (SESCn) (n = 0, 1) (1/2))

	7	6	5	4	3	2	1	0	Address	Initial value
SESC0	0	0	0	0	IES11	IES10	IES01	IES00	FFFF F272H	0H
SESC1	0	0	0	0	IES11	IES10	IES01	IES00	FFFF F292H	0H

Bit Position	Bit Name	Function
3, 2	IES11, IES10	Specifies the valid edge of TICnm pins.
1, 0	IES01, IES00	

IESn1	IESn0	Operation
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

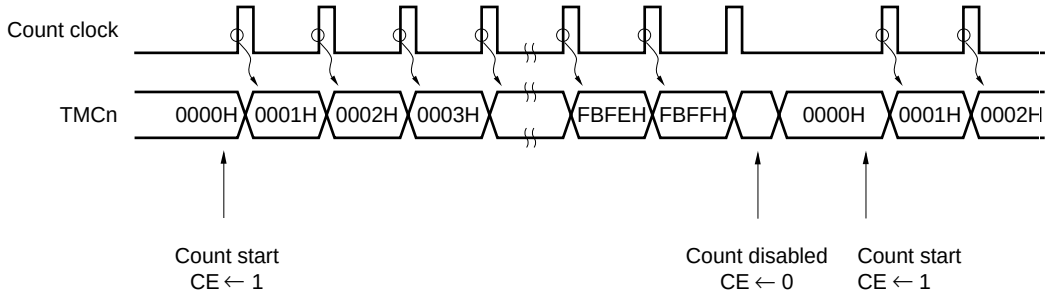
Remark: m, n = 0, 1

6.2.5 Operation

(1) Count operation

Timer C can function as a 16-bit free-running timer.
When it operates as a free-running timer and the CCC0n or CCC1n register and the TMCn count value match, an interrupt signal is generated and the timer output signal (TOCn) can be set or reset.
Also, a capture operation that holds the TMCn count value in the CCC0n or CCC1n register is performed, synchronized with the valid edge that was detected from the external interrupt request input pin as an external trigger. The capture value is held until the next capture trigger is generated.

Figure 6-39: Timing of basic operation of Timer C



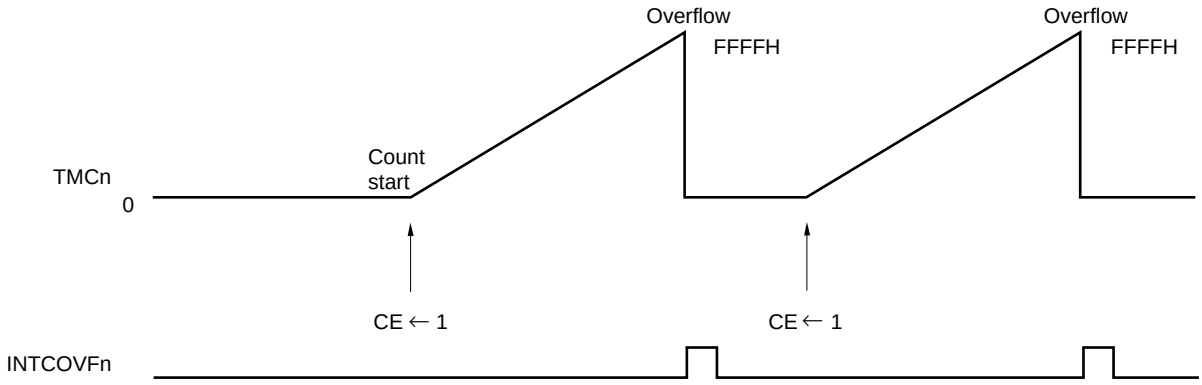
(2) Overflow

When the TMCn register has counted the count clock from FFFFH to 0000H, the OVF bit of the TMCC0n register is set to “1”, and an overflow interrupt (INTCOVFn) is generated at the same time.

However, if the CCC0n register is set to compare mode (CMS0 = 1) and to the value FFFFH, when match clearing is enabled (CCLR = 1) the TMCn counter register is considered to be cleared and the OVF bit is not set to “1” when the TMCn counter register changes from FFFFH to 0000H. Also, the overflow interrupt (INTCOVFn) is not generated.

When the TMCn counter register is changed from FFFFH to 0000H because the CE bit changes from “1” to “0”, the TMCn register is considered to be cleared, but the OVF bit is not set to 1 and no INTCOVFn interrupt is generated.

Figure 6-40: Timing of interrupt operation after overflow



(3) Capture operation

The TMCn register has two capture/compare registers. These are the CCC0n register and the CCC1n register.
A capture operation or a compare operation is performed according to the settings of both the CMS1 and CMS0 bits of the TMCC1n register. If the CMS1 and CMS0 bits of the TMCC1n register are set to “0”, the register operates as a capture register.

A capture operation that captures and holds the TMCn count value asynchronously relative to the count clock is performed synchronized with an external trigger.

An interrupt request (INTCCC0n or INTCCC1n) (n = 0, 1) is generated by TIC0n or TIC1n signal input and is used as an external trigger (capture trigger).
The valid edge of the capture trigger is set by valid edge selection register (SESCn).

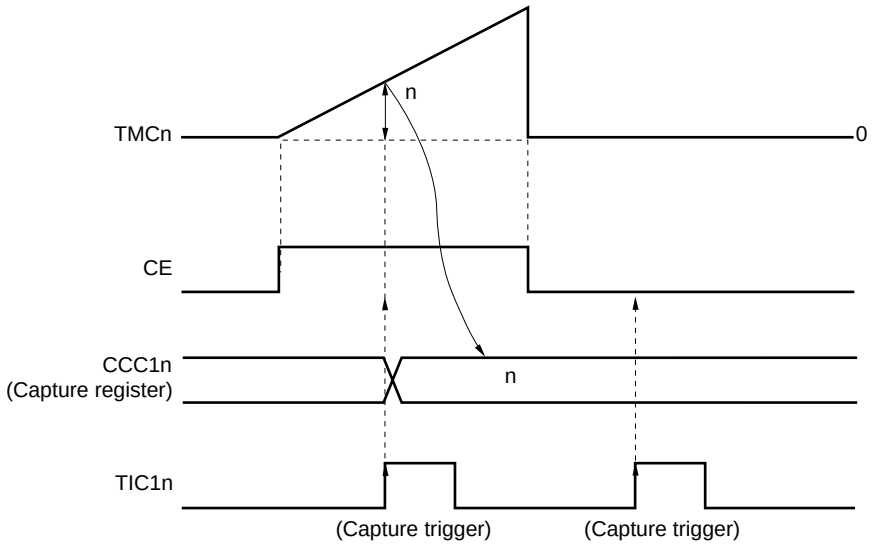
For the DCAN time stamp function TMC0 have got a selector for the external capture trigger. The CANCn register defines if the external pin TICn or the SOFOUT signal of the DCANn is used for capture operation.

The TMCn count value during counting is captured and held in the capture register, synchronized with that capture trigger signal. The capture register value is held until the next capture trigger is generated.

(a) Example: capture for pulse cycle measurement

If one of the edges is set as the capture trigger, the input **pulse cycle** can be measured.

Figure 6-41: Timing of capture for pulse cycle measurement (rising edge)



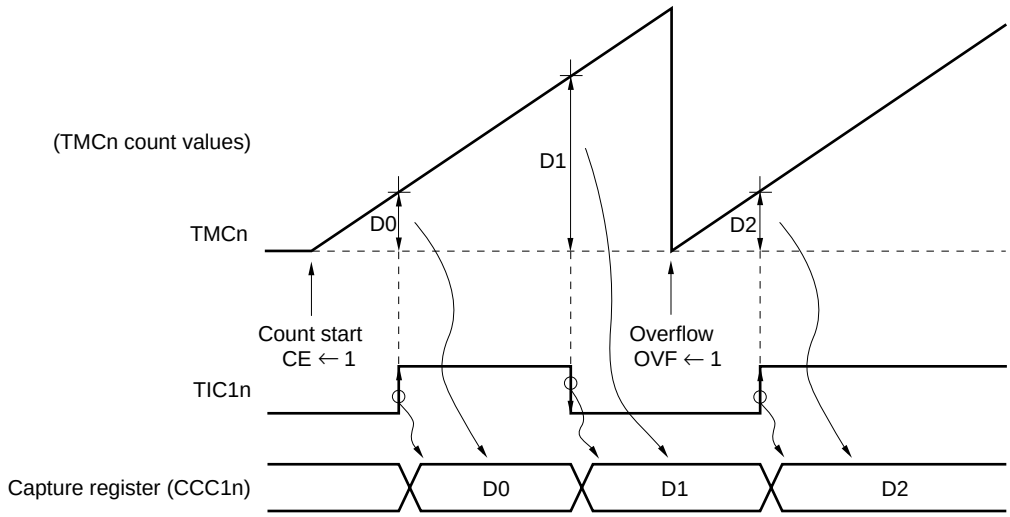
Remarks: 1. When the CE bit is 0, no capture operation is performed even if INTCCC1n is input.

2. Valid edge of TICn1: Rising edge

(b) Example: capture for pulse cycle measurement

If both the rising and falling edges are set as capture triggers, the input **pulse width** from an external source can be measured.

Figure 6-42: Timing of capture for pulse width measurement (both edges)



Remark: D0 to D2: TMCn count values

(c) Example: Cycle measurement

By setting the TMCC0n and TMCC1n registers as described below TMCn can measure the cycle of signals input to the TICn pin.
The valid edge of the TIC0 pin is selected according to the IES01 and IES00 bits of the SESCn register
(Similar the valid edge of the TIC1 pin is selected according to the IES11 and IES10 bits of the SESCn register.)
Either the rising edge, the falling edge, or both edges can be selected as the valid edges of both pins.

Setting method:

(1) set corresponding input port pins to the correct level (depends on control logic: ---, AND, OR)
(2) set CAE bit of TMCC0n register to 1 for activating the TMCn peripheral
(3) set the valid edge of the TICn pin with the IES01 and IES00 bits of the SESCn register (here for rising edge: IES01 = 0, IES00 = 1)
(4) set CMS1 and CMS0 bit of TMCC1n register to 0
(5) set CE bit to enable the counter and start operation

Operation:

(1) the valid edge input of the TIC0n-pin is set as the trigger for capturing the TMCn register value in the CCC0n register.
(2) When this value is captured, an INTCCC0n interrupt is generated.

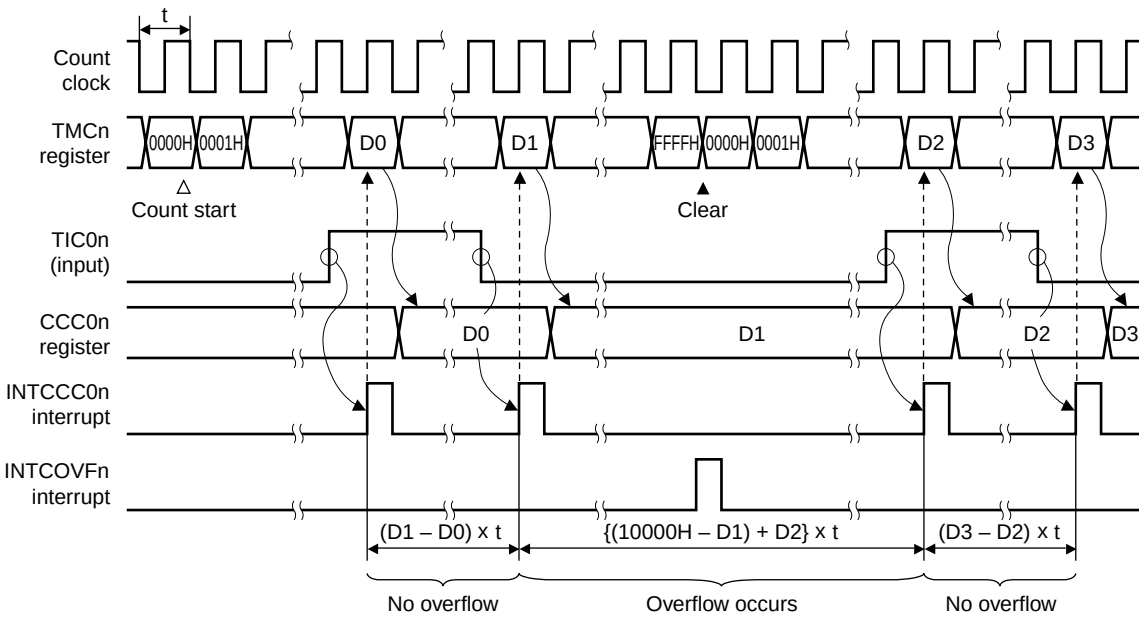
(Similarly, the valid edge input of the TIC0n-pin is set as the trigger for capturing the TMCn register value in the CCC1n register. When this value is captured, an INTCCC1n interrupt is generated.)

Calculation:

The cycle of signals input to the INTCCC0n pin is calculated by obtaining the difference between the TMCn register's count value (Dx) that was captured in the CCC0n register according to the x-th valid edge input of the TIC0n pin and the TMCn register's count value (D(x+1)) that was captured in the CCC0n register according to the (x+1)-th valid edge input of the TIC0n pin and multiplying the value of this difference by the cycle of the clock control signal.

(Similarly the cycle of signals input to the INTCCC1n pin is calculated by obtaining the difference between the TMCn register's count value (Dx) that was captured in the CCC1n register according to the x-th valid edge input of the TIC1n pin and the TMCn register's count value (D(x+1)) that was captured in the CCC1n register according to the (x+1)-th valid edge input of the TIC1n pin and multiplying the value of this difference by the cycle of the clock control signal.)

Figure 6-43: Timing of cycle measurement operation



Caution: An overflow must not be generated more than once between the 1st and 2nd INTCCC0n interrupts.

Remarks: 1. D0 to D3: TMCn register count values

2. t: Count clock cycle

3. In this example, the valid edge of TIC0n input has been set to both edges (rising and falling).

(4) Compare operation

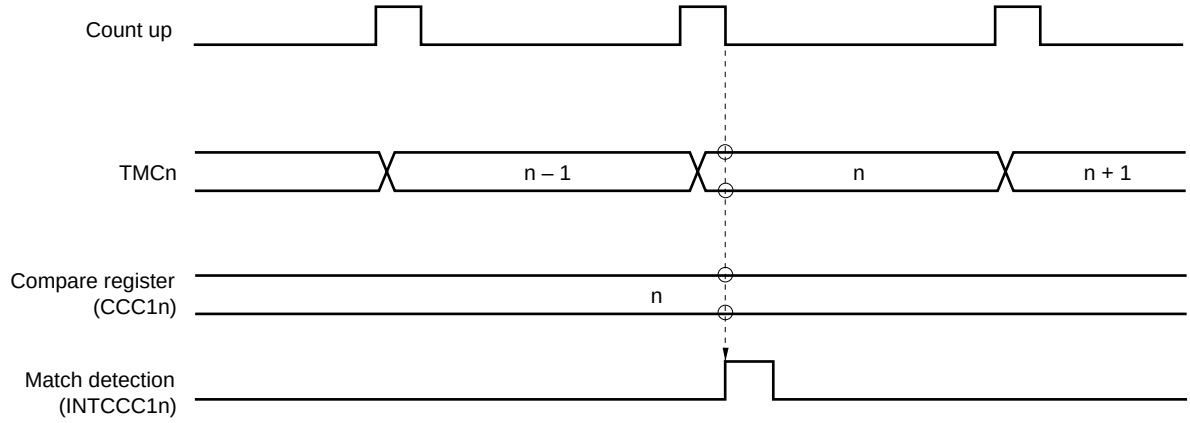
The TMCn register has two capture/compare registers. These are the CCC0n register and the CCC1n register.

A capture operation or a compare operation is performed according to the settings of both the CMS1 and CMS0 bits of the TMCC1n register: If “1” is set in the CMS1 and CMS0 bits of the TMCC1n register, the register operates as a compare register.

A compare operation that compares the value that was set in the compare register and the TMCn count value is performed.

If the TMCn count value matches the value of the compare register, which had been set in advance, a match signal is sent to the output control circuit. The match signal causes the timer output pin (TOCn) to change and an interrupt request signal (INTCCC0n, INTCCC1n) to be generated at the same time.

Figure 6-44: Timing of compare operation



Remark: The match is detected immediately after the count up, and the match detection signal is generated.

(a) When CCC0n register is set to 0000H

If the CCC0n register is set to 0000H, the 0000H after the TMCn register counts up from FFFFH to 0000H is judged as a match. The 0000H when the TMCn register begins counting is not judged as a match.

(b) When match clearing is enabled

If match clearing is enabled (CLR bit = 1) for the CCC0n register, the TMCn register is cleared when a match with the TMCn register occurs during a compare operation.

(c) Example: Interval timer

By setting the TMCC0n and TMCC1n registers as described below TMCn operates as an interval timer that repeatedly generates interrupt requests with the value that was set in advance in the CCC0n register as the interval.

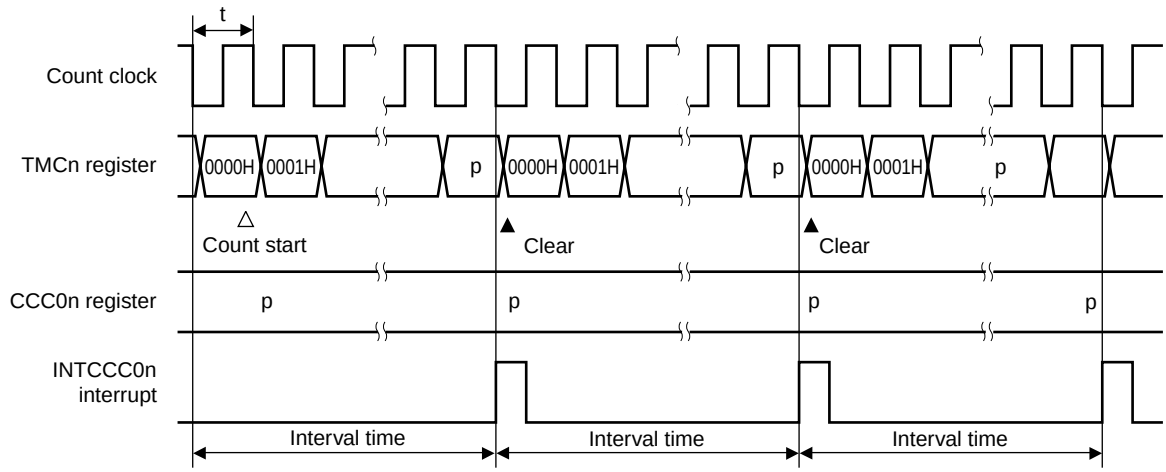
Setting method:

(1) set corresponding input port pins to the correct level (depends on control logic: ---, AND, OR)
(2) set CAE bit to “1” for activate the TMCn peripheral
(3) set CLR and CMS0 bit of TMCC1n register to “1”
(4) set CE bit to “1” to enable the counter and start operation

Operation:

(1) When the counter value of the TMCn register matches the setting value of the CCC0n register, the TMCn register is cleared (0000H)
(2) An interrupt request signal (INTCCC0n) is generated at the same time that the count operation resumes.

Figure 6-45: Timing of interval timer operation



Remarks: 1. p:Setting value of CCC0n register (0000H to FFFFH)

2. t:Count clock cycle

3. Interval time = (p + 1) × t

(5) PWM output

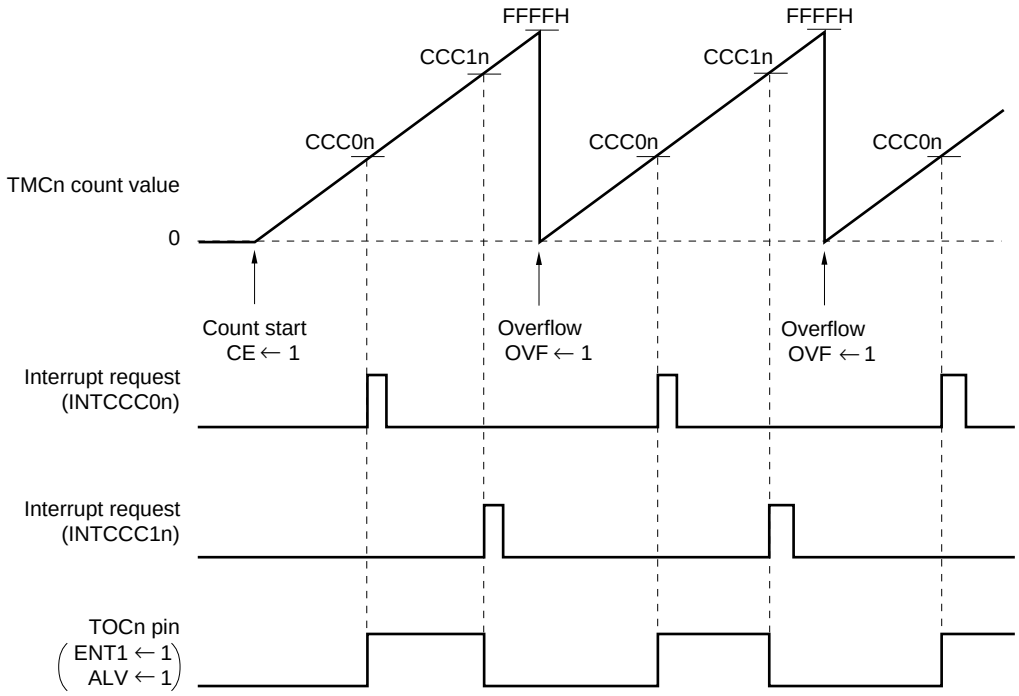
Timer C has one timer output pin (TOCn).
An external pulse output (TOCn) can be generated when a match of the two compare registers (CCC0n and CCC1n) and the TMCn register is detected.

If a match is detected when the TMCn count value and the CCC0n value are compared, the output level of the TOCn pin is set.
Also, if a match is detected when the TMCn count value and the CCC1n value are compared, the output level of the TOCn pin is reset.
The output level (set, reset) depends on the settings of the ALV and ENTO bits of the TMCC1n register.

Table 6-3: TOCn Output Control

ENTO	ALV	TOCn Output	
		External Pulse Output	Output Level
0	0	Disable	High level
0	1	Disable	Low level
1	0	Enable	When the CCC0n register is matched: Low level When the CCC1n register is matched: High level
1	1	Enable	When the CCC0n register is matched: High level When the CCC1n register is matched: Low level

Figure 6-46: Timing of PWM output operation (overview)



(a) Example PWM output

By setting the TMCC0n and TMCC1n registers as described below TMCn can output a PWM of an arbitrary frequency with the values that were set in advance in the CCC0n and CCC1n registers determining the intervals.

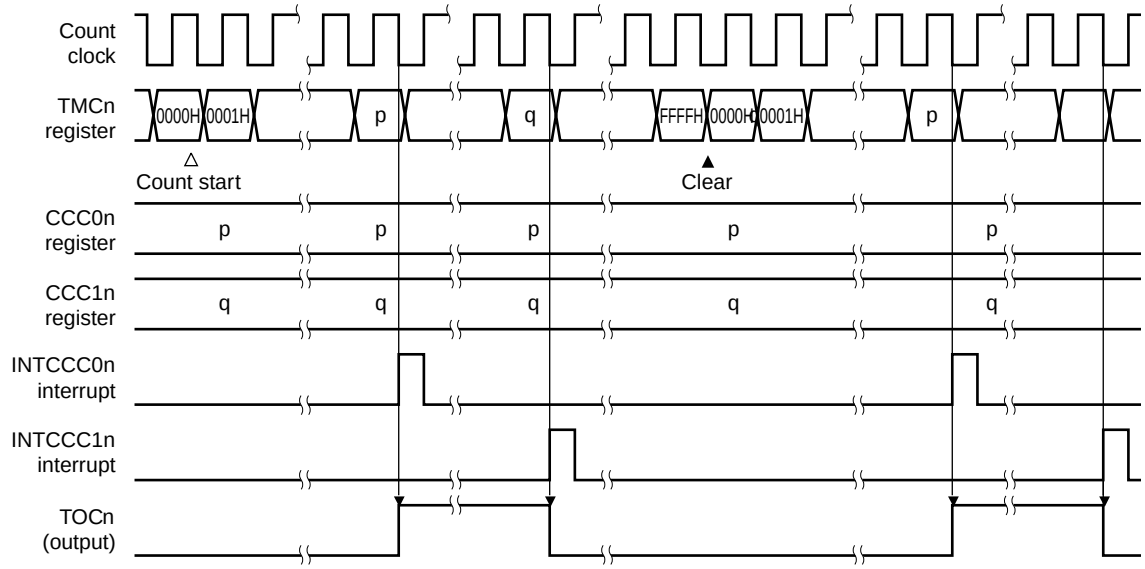
Setting method:

(1) set corresponding output port pins to the correct level (depends on control logic: ---, AND, OR)
(2) set CAE bit of TMCC0n register to “1” for activating the TMCn peripheral
(3) set the active level of TOCn output by the ALV bit of the TMCC1n register (here: ALV = 1)
(4) set ENT1, CMS1 and CMS0 bit of TMCC1n register to “1” (leave CLR bit to 0)
(5) set CE bit to “1” to enable the counter and start operation

Operation:

(1) When the counter value of the TMCn register matches the setting value of the CCC0n register, the TOCn output becomes active.
(2) When the counter value of the TMCn register matches the setting value of the CCC1n register, the TOCn output becomes inactive. This enables a PWM of an arbitrary frequency to be output.

Figure 6-47: Timing of PWM output operation (detail)



Remarks: 1. p: Setting value of CCC0n register (0000H to FFFFH)

2. q: Setting value of CCC1n register (0000H to FFFFH)

3. p: 1/4 q

4. In this example, the active level of TOCn output is set to high level.

(b) When CCC0n = CCC1n

When the setting value of the CCC0n register and the setting value of the CCC1n register are the same, the TOCn output remains inactive and does not change.

6.2.6 Precautions Timer C

Various precautions concerning Timer C are shown below.

(1) The following bits and registers must not be rewritten during operation (TMCC0n register CE = 1).

- CS2 to CS0 bits of TMCC0n register
- TMCC1n register
- SESCn register

(2) The CAE bit of the TMCC0n register is a TMCn counter reset signal. To use TMCn, first set the CAE bit to 1.

(3) The analog noise elimination time + two cycles of the input clock are required to detect a valid edge of the external input (TIC0n or TIC1n). Therefore, edge detection will not be performed normally for changes that are less than the analog noise elimination time + two cycles of the input clock.

(4) The operation of an interrupt output (INTCCC0n or INTCCC1n) is automatically determined according to the operating state of the capture/compare registers (CCC0n, CCC1n). When the capture/compare register is used for a capture mode, the external trigger (TIC0n,TIC1n) is used for valid edge detection. When the capture/compare register is used for a compare mode, the external interrupt output is used for a match interrupt indicating a match with the TMCn register.

(5) If the ENTO and ALV bits of the TMCC1n register are changed at the same time, a glitch (spike shaped noise) may be generated in the TOCn pin output. Either create a circuit configuration that will not malfunction even if a glitch is generated or make sure that the ENTO and ALV bits do not change at the same time.

6.3 Timer 5 (TM50, TM51)

6.3.1 Functions

- 8-bit timer Timer 5n (n = 0, 1) has the following two modes:

  - Mode using timer alone (individual mode)
  - Mode using the cascade connection (16-bit resolution: cascade connection mode)

- 16-bit counters (TM50/51)

- 16-bit compare registers (CR50/51)

These two modes are described next.

(1) Mode using timer alone (individual mode)

The timer operates as an 8-bit timer/event counter.
It can have the following functions:

- Interval timer
- External event counter
- Square wave output
- PWM output

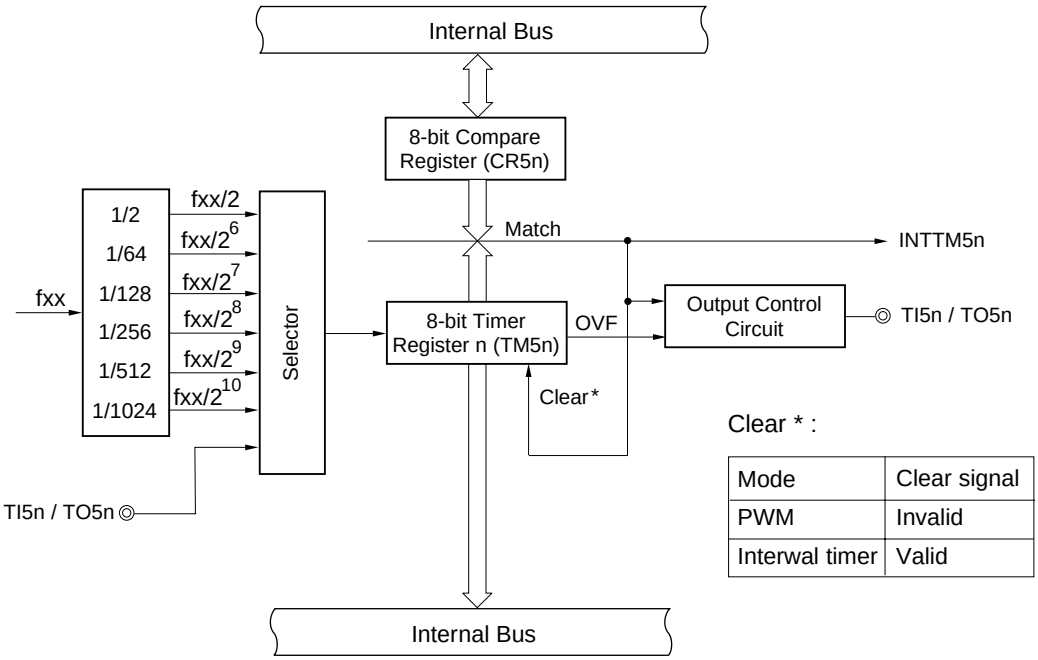
(2) Mode using the cascade connection (16-bit resolution: cascade connection mode)

The timer operates as a 16-bit timer/event counter by connecting TM50 and TM51in cascade.
It can have the following functions:

- Interval timer with 16-bit resolution
- External event counter with 16-bit resolution
- Square wave output with 16-bit resolution

Caution: In cascade mode there is not a 16-bit precision PWM output.

Figure 6-48: Block Diagram of Timer 50, Timer 51



Clear * :

Mode	Clear signal
PWM	Invalid
Interwal timer	Valid

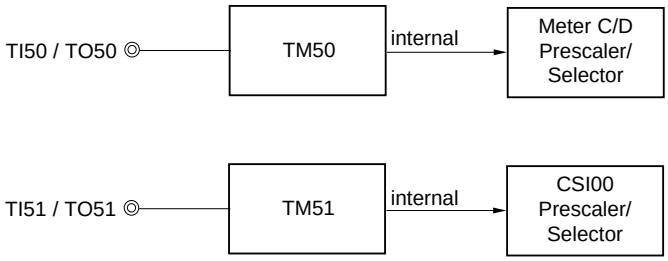
Notes: 1. n = 0, 1

2. Count clock is set by the TCL5n register.

Remark: TI5n/TO5n is a signal that can be directly connected to the port.

Furthermore the Timer 5n can be used as a clock source for the Meter- and the CSI00-peripheral. Therefore internal output connections are used to supply the output to the prescaler of the Meter- or CSI00-prescaler.

Figure 6-49: Timer 50 and Timer 51 as clock source for Meter and CSI00



6.3.2 Configuration

Timer 5n is constructed from the following hardware.

Table 6-4: Timers 5n Configuration

Item	Configuration
Timer registers	8-bit counters (TM50, TM51) 16-bit counters: Only when connecting in cascade
Registers	8-bit compare registers (CR50, CR51) 16-bit compare registers: Only when connecting in cascade
Capture inputs	TI50, TI51 (shared with Timer inputs)
Timer outputs	TO50, TO51 (shared with Capture outputs)
Control registers	Timer clock select registers (TCL50, TCL51) 8-bit timer mode control registers (TMC50, TMC51)

(1) 8-bit counters (TM50, TM51)

The counter increments synchronous to the rising edge of the count clock.

TM5n is an 8-bit read-only register.

Figure 6-50: TM50, TM51 Timer Count Registers (TM50, TM51)

	7	6	5	4	3	2	1	0	Address	After reset
TM50	0	0	0	0	0	0	0	0	FFFF F2A0H	00H
TM51	0	0	0	0	0	0	0	0	FFFF F2A2H	00H

TM50 and TM51 can be connected in cascade and used as 16-bit timers.
When TM50 and TM51 are connected in cascade and used as a 16-bit timer, they can be read by a 16-bit memory manipulation instruction. In cascade mode timer 50 represents the low byte of the time base, while timer 51represents the high byte.
However, since they are connected via the internal 8-bit bus, TM50 and TM51are read separately. Consequently, they should be read twice before comparison to allow for count variation.
When the count is read out during operation, the count clock input temporarily stops and the count is read at that time. In the following cases, the count becomes 00H.

(1) RESET is input.
(2) TCE bit is cleared.
(3) TM5n and CR5n match in the “clear and start mode” that occurs when TM5n and CR5n match.

Caution: When connected in cascade, these registers become 00H even when TCE bit of TMC50 register for the lowest-order timer (TM50) is cleared.

Remark: n = 0, 1

(2) 8-bit compare registers 0, 1 (CR50, CR51)

The value set in CR5n is always compared to the count in the 8-bit counter (TM5n). If the two values match, an interrupt request (INTTM5n) is generated (except in the PWM mode).

This is an 8-bit read/write register.

Figure 6-51: TM50, TM51 Compare Registers (CR50, CR51)

	7	6	5	4	3	2	1	0	Address	After reset
CR50	0	0	0	0	0	0	0	0	FFFF F2A4H	00H
CR51	0	0	0	0	0	0	0	0	FFFF F2A6H	00H

The value of CR5n can be set in the range of 00H to FFH, and can be rewritten during counting.

When TM50 and TM51 are connected in cascade and used as a 16-bit timer, CR50 and CR51 operate as a 16-bit compare register.
This registers generate the interrupt request INTTM50, when the cascade 16-bit counter value and the cascade 16-bit register value are compared and match.

Cautions: 1. Since the INTTM51 interrupt request is also generated at that time, mask the INTTM51 interrupt request when TM50 and TM51 are used connected in cascade.

2. If data is set in a cascade connection, always set after stopping the timer.

6.3.3 Timer 5n control register

The following two registers controls Timer 5n.

- Timer clock select registers (TCL50, TCL51)
- 8-bit timer mode control register (TMC50, TMC51)

(1) Timer 5n clock select registers (TCL50, TCL51)

These registers set the count clock of timer 5n.

This register can be read/written by 1/8-bit memory manipulation.

Figure 6-52: Timer 5 Clock Select Registers (TCL50, TCL51)

	7	6	5	4	3	2	1	0	Address	After reset
TCL50	0	0	0	0	0	TCL2	TCL1	TCL0	FFFF F2A8H	00H
TCL51	0	0	0	0	0	TCL2	TCL1	TCL0	FFFF F2AAH	00H

TCL2	TCL1	TCL0	Count Clock Selection	Resolution (µs)	Max. Counting Time (µs)	Frame Frequency (KHz)
0	0	0	TI5n falling edge	-	-	
0	0	1	TI5n rising edge	-	-	
0	1	0	$f_{XX}/2$	0.125	32	31.25
0	1	1	$f_{XX}/64$	4	1024	0.977
1	0	0	$f_{XX}/128$	8	2048	0.488
1	0	1	$f_{XX}/256$	16	4096	0.244
1	1	0	$f_{XX}/512$	32	8192	0.122
1	1	1	$f_{XX}/1024$	64	16384	0.061

Cautions: 1. When TCL50 and TCL51 are overwritten by different data, write after temporarily stopping the timer.

2. Always set bits 3 to 7 to in TCL5n to “0”.

3. When connected in cascade, the settings of TCL2 to TCL0 of TM51 are invalid.

(2) Timer 5 mode control registers (TMC50, TMC51)

The TMCn register makes the following six settings.

(1) Controls the counting by the 8-bit counter TM5n
(2) Selects the operating mode of the 8-bit counter TM5n
(3) Selects the individual mode or cascade connection mode
(4) Sets the state of the timer output flip-flop
(5) Controls the timer flip-flop or selects the active level in the PWM (free running) mode
(6) Controls timer output

TMC5n is set by an 8-/1-bit memory manipulation instruction.

Remark: Although the state of hardware is initialized to 04H, 00H is read when reading.

Figure 6-53: Timer 5 Mode Control Registers (TMC50, TMC51) (1/2)

	7	6	5	4	3	2	1	0	Address	After reset
TMC50	TCE	TMC56	0	0	LVS5	LVR5	TMC51	TOE	FFFF F2ACH	00H

	7	6	5	4	3	2	1	0	Address	After reset
TMC51	TCE	TMC56	0	TMC54	LVS5	LVR5	TMC51	TOE	FFFF F2AEH	00H

TCE[Note]	TM5n Count Operation Control
0	Counting is disabled after the counter is cleared to 0 (prescaler disabled)
1	Start count operation

Note: In the PWM mode, the PWM output is set to the inactive level by bit TCE = 0.

TMC56	TM5n Operating Mode Selection
0	Clear & Start mode when TM5n and CR5n match
1	PWM (free running) mode

TMC54[Note]	Individual Mode or Cascade Connection Mode Selection
0	Individual mode (fixed to 0 when timer is TM50)
1	Cascade connection mode (connection of TM51 to low-order timer TM50)

Note: Not available for TMC50

Figure 6-53: Timer 5 Mode Control Registers (TM50 to TM51) (2/2)

LVS5[Note]	LVR5[Note]	Setting State of Timer Output Flip-Flop
0	0	Not change
0	1	Reset timer output flip-flop to 0
1	0	Set timer output flip-flop to 1
1	1	Setting prohibited

Note: If bits LVS5 and LVR5 are read after setting data, 0 is read.

TMC51	PWM (free running) Mode (TMC56 = 1)	Other than PWM (free running) Mode (TMC56 = 0)
	Selects active level	Controls timer output TO5n
0	Active high	Disable inversion operation
1	Active low	Enable inversion operation

TOE	Timer Output Control
0	Disable output (port mode)
1	Enable output

Cautions: 1. When using as the timer output pin (TO5n), set the port value to 0 (port mode output). An logic OR connected value of the timer output value is output.

2. Since TO5n and TI5n share the same pin, either function can only be used.

6.3.4 8-Bit Timer Operation

(1) Operating as an interval timer (8-bit operation)

The timer operates as an interval timer that repeatedly generates interrupts at the interval of the preset count in the 8-bit compare register (CR5n).
If the count in the 8-bit counter TM5n matches the value set in CR5n, simultaneous to clearing the value of TM5n to 0 and continuing the count, the interrupt request signal (INTTM5n) is generated.
The TM5n count clock can be selected by bits 0 to 2 (TCL0 to TCL2) in the timer clock select register (TCL5n).

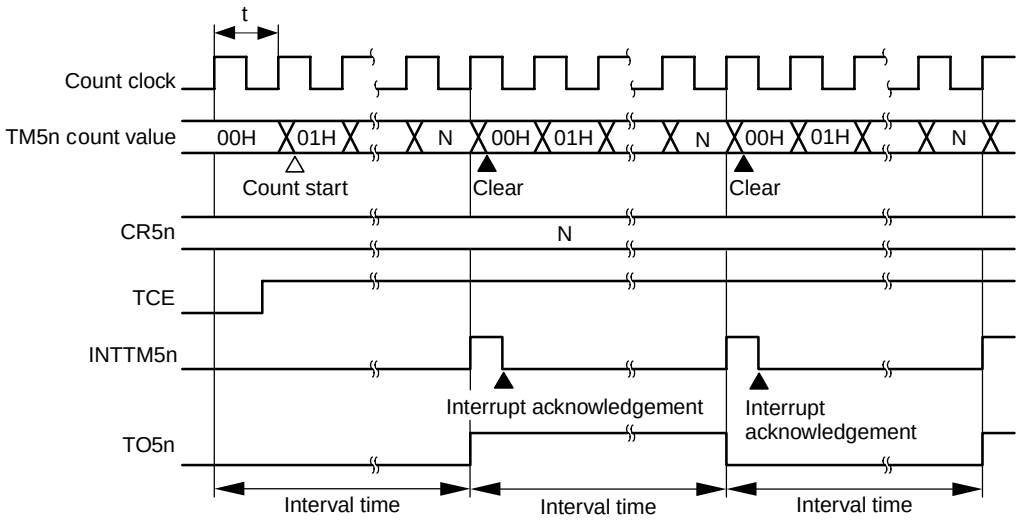
Setting Method

(1) Set each register:
- TCL5n: Selects the count clock.
- CR5n: Compare value
- TMCn: Selects the clear and start mode when TM5n and CR5n match.
  (TMCn = 0000xxx0B, x = doesn't care)

(2) When bit TCE = 1 is set, counting starts.
(3) When the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).
(4) Then, INTTM5n is repeatedly generated during the same interval. When counting stops, set TCE = 0.

Figure 6-54: Timing of Interval Timer Operation (1/3)

(a) Basic Operation

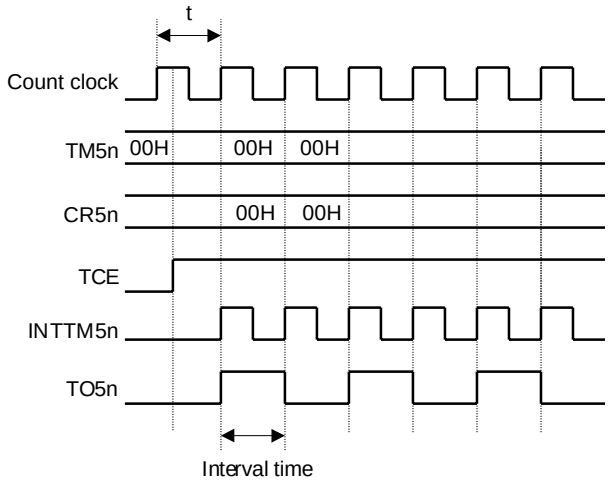


Remarks: 1. Interval time = (N + 1) × t; N = 00H to FFH

2. n = 0, 1

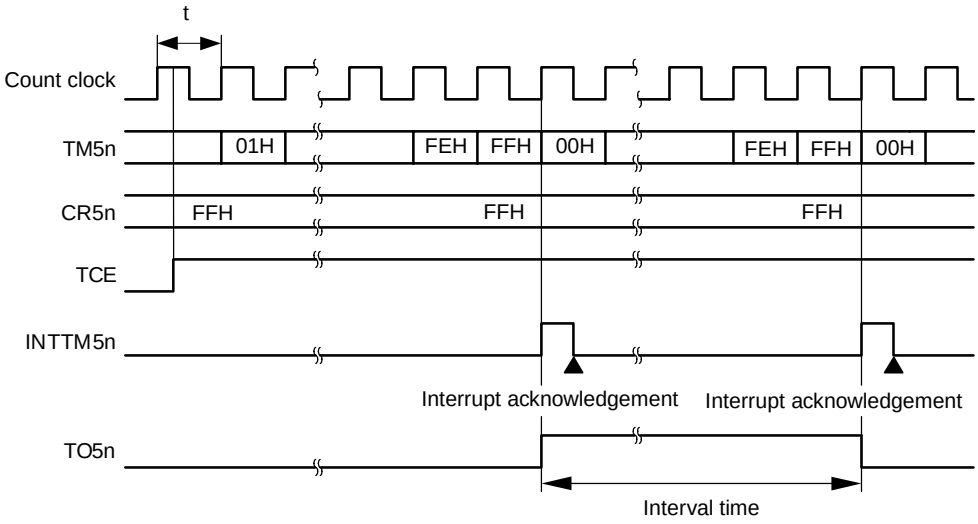
Figure 6-54: Timing of Interval Timer Operation (2/3)

(b) When CR5n = 00H



Remark: n = 0, 1

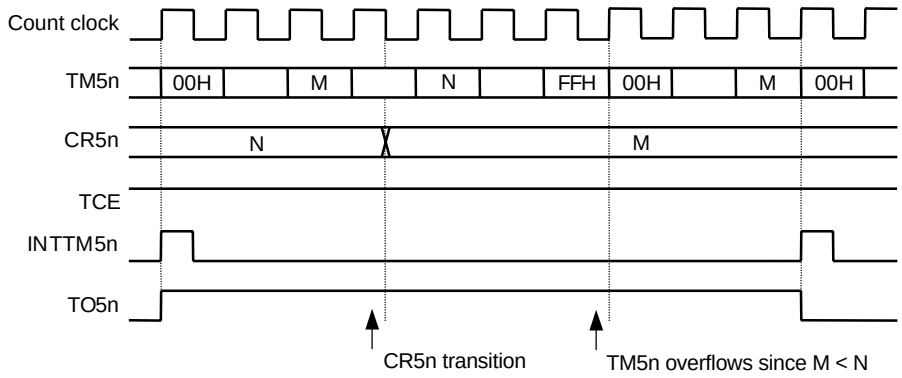
(c) When CR5n = FFH



Remark: n = 0, 1

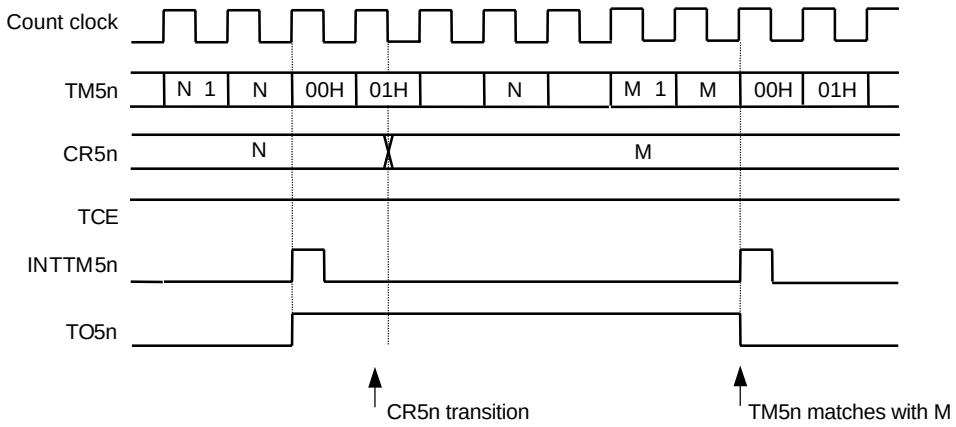
Figure 6-54: Timing of Interval Timer Operation (3/3)

(d) Operated by CR5n transition (M < N)



Remark: n = 0, 1

(e) Operated by CR5n transition (M > N)



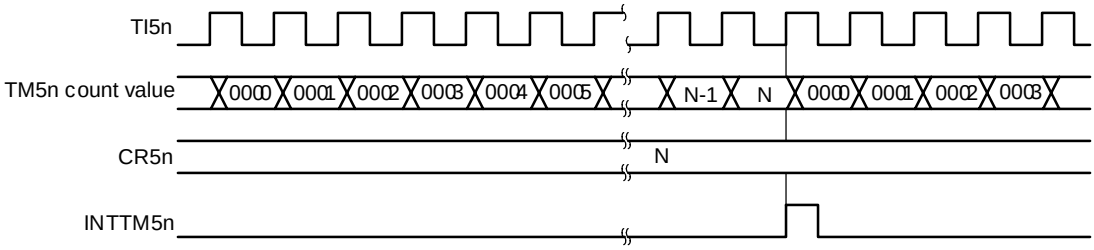
Remark: n = 0, 1

(2) Operating as external event counter

The external event counter counts the number of external clock pulses that are input to TI5n.
Each time a valid edge specified with the timer clock select register (TCL5n) is input, it increments.
The edge setting can be selected to be either a rising or falling edge.
If the value of TM5n and the value of the 8-bit compare register n (CR5n) match, TM5n is cleared to 0 and the interrupt request signal (INTTM5n) is generated.
INTTM5n is generated each time the TM5n value matches the CR5n value.

Remark: n = 0, 1

Figure 6-55: Timing of external event counter operation



For edge detection the rising edge is set.

Remark: n = 0, 1

(3) Operating as square wave output (8-bit resolution)

A square wave having any frequency is output at the interval preset in the 8-bit compare register (CR5n).

By setting TOE bit of the 8-bit timer mode control register (TMCn) to “1”, the output state of TO5n is inverted with the count preset in CR5n as the interval. Therefore, a square wave output having any frequency (duty factor = 50%) is possible.

Setting method

(1) Set the registers.

- Set the port latch and port mode register to 0
- TCL5n: Selects the count clock
- CR5n: Compare value
- TMCn: Clear and start mode when TM5n and CR5n match

LVS	LVR	Setting State of Timer Output Flip-flop
1	0	High level output
0	1	Low level output

- Inversion of timer output flip-flop enabled
- Timer output enabled → TOE = 1

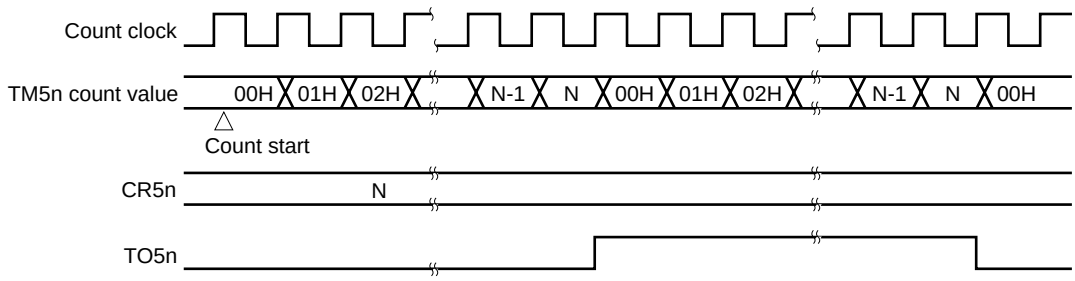
(2)When TCE = 1 is set, the counter starts operating.

(3)If the values of TM5n and CR5n match, the timer output flip-flop inverts. Also, INTTM5n is generated and TM5n is cleared to 00H.

(4)Then, the timer output flip-flop is inverted for the same interval to output a square wave from TO5n.

Remark: n = 0, 1

Figure 6-56: Timing of Square Wave Output Operation Timing



Note: The initial value of TO5n output can be set with TMCn register bits 3 and 2 (LVS, LVR).

Remark: n = 0, 1

(4) Operating as 8-bit PWM output

By setting the TMC56 bit of the 8-bit timer mode control register (TMC5n) to “1”, the timer operates as a PWM output.
Pulses with the duty factor determined by the value set in the 8-bit compare register (CR5n) is output from TO5n.
Set the width of the active level of the PWM pulse in CR5n. The active level can be selected by TMCn bit in register TMC5n.
The count clock can be selected by bits 0 to 2 (TCL0 to TCL2) of timer clock select register (TCL5n).
The PWM output can be enabled and disabled by TOE bit of register TMC5n.

Caution: CR5n can be rewritten only once in one period while in the PWM mode.

Remark: n = 0, 1

(a) Basic operation of the PWM output

Setting method

(1) Set the port latch and port mode register n to 0.
(2) Set the active level width in the 8-bit compare register (CR5n).
(3) Select the count clock with the timer clock select register (TCL5n).
(4) Set the active level in TMC51 bit of TMC5n, set TMC56 bit of TMC5n to “1” for PWM
(5) If bit TCE bit of TMC5n is set to “1”, counting starts. When counting stops, set TCE to “0”.

PWM output operation

(1) When counting starts, the PWM output (output at TO5n-pin) outputs the inactive level until an overflow occurs.
(2) When the overflow occurs, the active level specified in step (1) in the setting method is output. The active level is output until CR5n register and the count of the 8-bit counter (TM5n) match.
(3) The PWM output after CR5n register and the count match is the inactive level until an overflow occurs again.
(4) Steps (2) and (3) repeat until counting stops.
(5) If counting is stopped by TCE = 0, the PWM output goes to the inactive level.

Remark: n = 0, 1

Figure 6-57: Timing of PWM Output (1/2)

(a) Basic operation (active level = H)

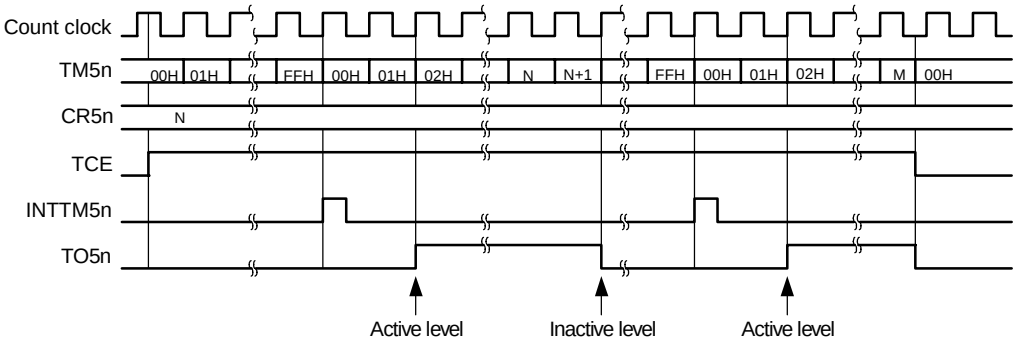
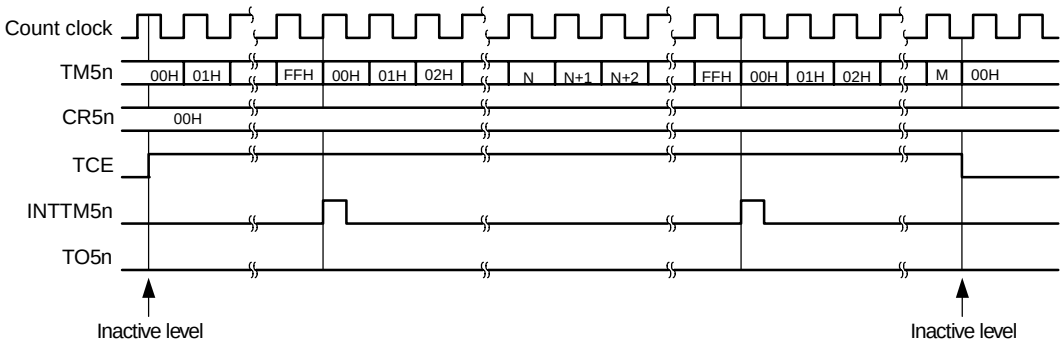
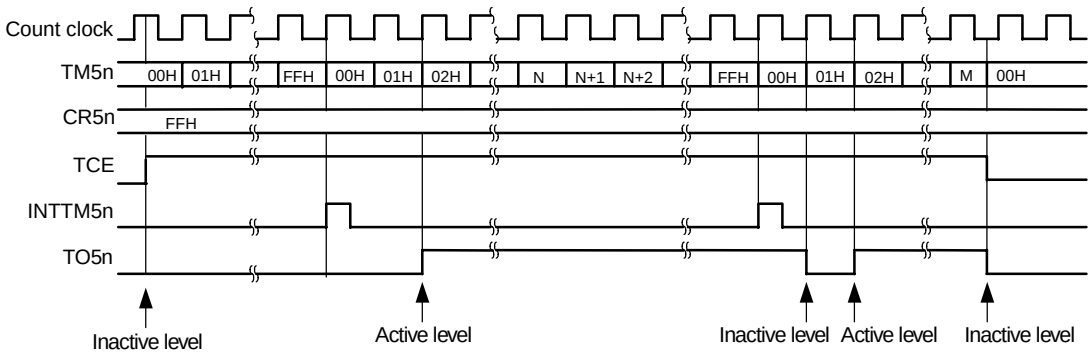


Figure 6-57: Timing of PWM Output (2/2)

(b) When CR5n = 0



(c) When CR5n = FFH

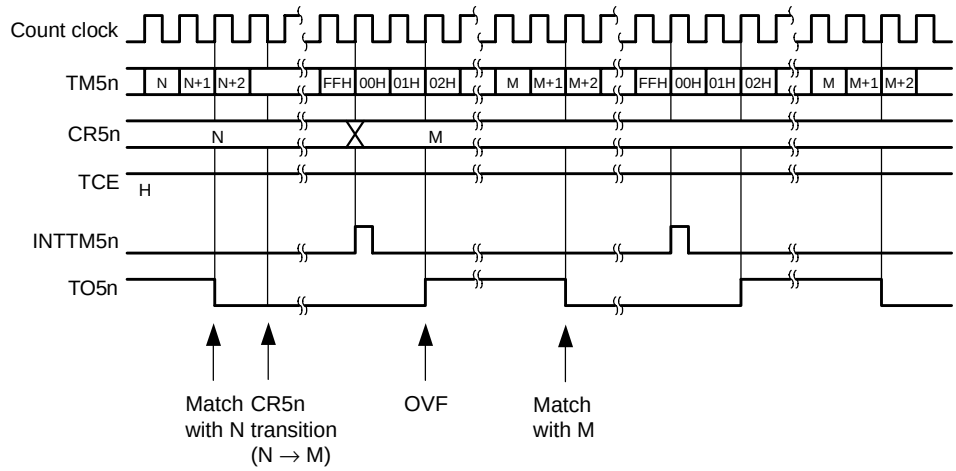


Remark: n = 0, 1

(b) Operation based on CR5n transitions

Figure 6-58: Timing of Operation Based on CR5n Transitions (1/2)

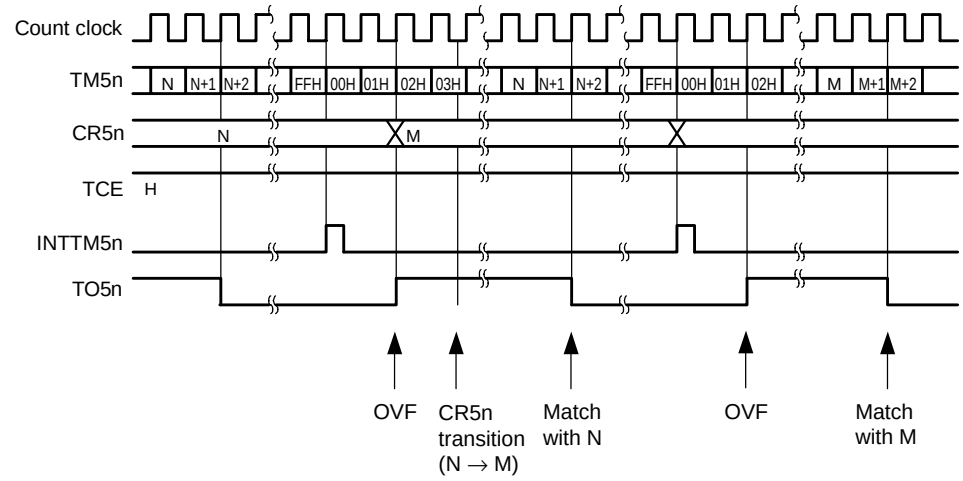
(a) When the CR5n value changes from N to M before TM5n overflows



Remark: n = 0, 1

After the next overflow M will be use for match.

(b) When the CR5n value changes from N to M after TM5n overflows

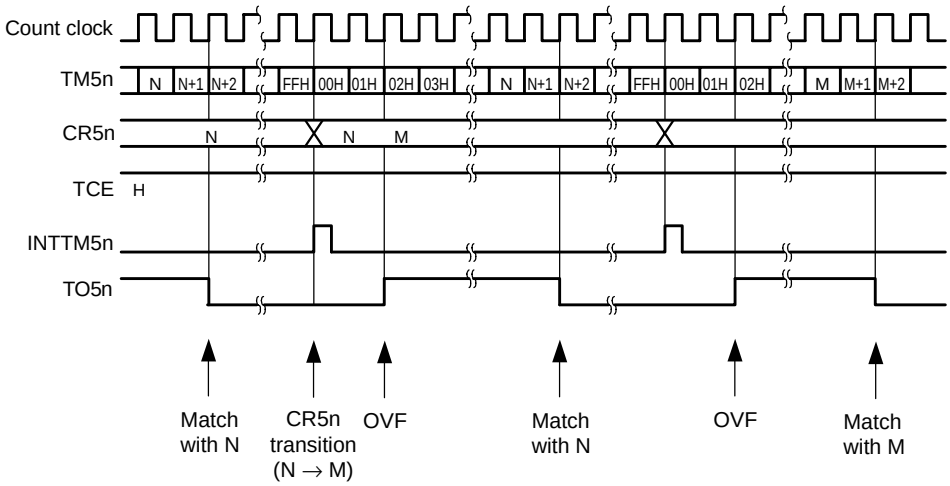


Remark: n = 0, 1

In this cycle N will be use for match. M will be used after the next overflow.

Figure 6-58: Timing of Operation Based on CR5n Transitions (2/2)

(c) When the CR5n value changes from N to M during two clocks (00H, 01H) immediately after TM5n overflows



Remark: n = 0, 1

It doesn’t change anything if the value changes immediately (within 2 clocks) after the overflow or later. In this cycle N will be use for match. M will be used after the next overflow.

6.3.5 Operating as interval timer (16 bits)

(1) Cascade connection (16-bit timer) mode

The V850/DB1 provides a 16-bit register that can be used when connecting in cascade mode.

By setting bit 4 (TMC54) of the 8-bit timer mode control register (TMC51) of Timer 51 to “1”, the timer enters the timer/counter mode with 16-bit resolution.
With the count preset in the 8-bit compare register (CR5n) as the interval, the timer operates as an interval timer by repeatedly generating interrupts.
The following shows a setting method when using TM50 and TM51 in cascade.

Setting method (TM51, TM50 in cascade connection)

(1) Setting registers
- TCL50, TCL51: Select the count clock for TM50 (setting not necessary for TM51 because of cascade connection).
- CR50, CR51: Compare value (00H to FFH can be set for compare values)
- TMC50: Selects clear & start mode on a match of TM50 and CR50 (x: doesn’t care)
  TM50 → TMC50 = 0000xxx0B
  TM51 → TMC51 = 0001xxx0B

(2) Set the TCE bit of TMC51 to “1”. After that, set the TCE bit of TMC50 to “1” to start the count operation.
(3) When the TM50/51 and CR50/51 values of the timer connected in cascade match, INTTM50 of TM50 is generated (TM50 and TM51 are cleared to 00H).
(4) IMTTM50 is then repeatedly generated at the same interval.

Cautions: 1. When 8-bit timers (TM50, TM51) are connected in cascade and used as a 16-bit timer, change the setting value of the compare register (CR50, CR51) after stopping the count operation of the 8-bit timers connected in cascade.

2. If the value of CR50, CR51 is changed without stopping the timers, the values of the higher 8 bits (TM51) become undefined.

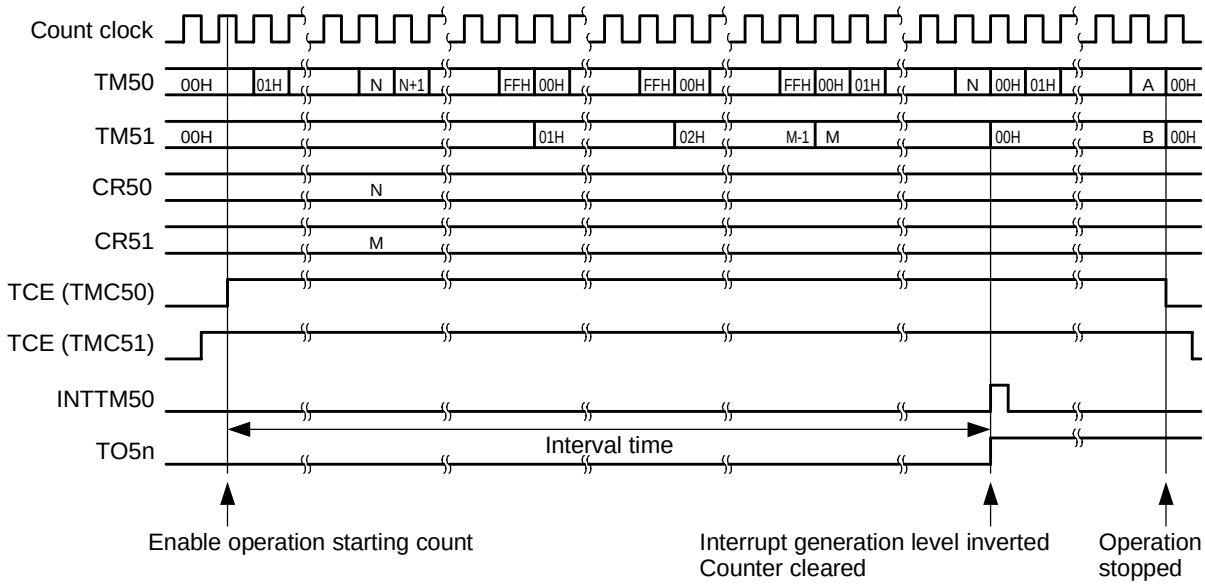
3. Even during cascade connection, the interrupt request (INTTM51) of higher timer (TM51) is generated when the count value of higher timer (TM51) matches CR51. Be sure to mask INTTM51 to disable this interrupt.

4. To set the TCE bit, set TCE50 after TCE51 is set.

5. The count can be started or stopped by setting the TCE bit of TMC50.

A timing example of the cascade connection mode with 16-bit resolution is shown below.

Figure 6-59: Cascade connection mode with 16 bit resolution



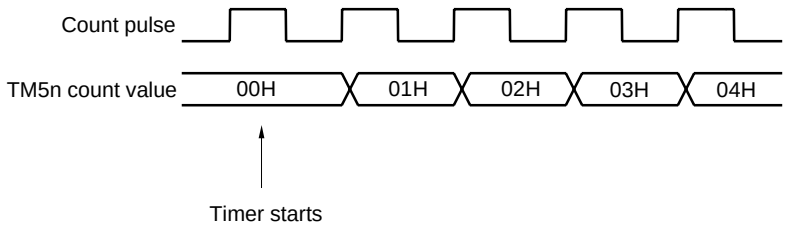
Remark: n = 0, 1

6.3.6 Precautions Timer 5

(1) Error when the timer starts

The time until the coincidence signal is generated after the timer starts has a maximum error of one clock. The reason is the starting of the 8-bit counter (TM5n) is asynchronous with respect to the count pulse.

Figure 6-60: Start Timing of Timer n

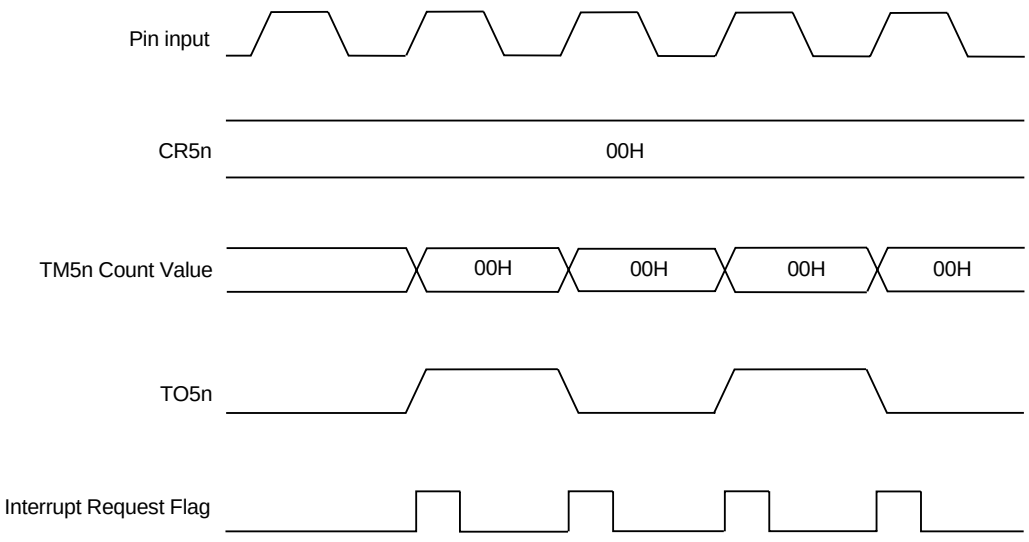


Remark: n = 0, 1

(2) Compare register (CR50 and CR51) set to 00H

Thus, when an 8-bit compare register is used as an event counter, one-pulse count operation can be carried out.

Figure 6-61: External Event Counter Operation Timings



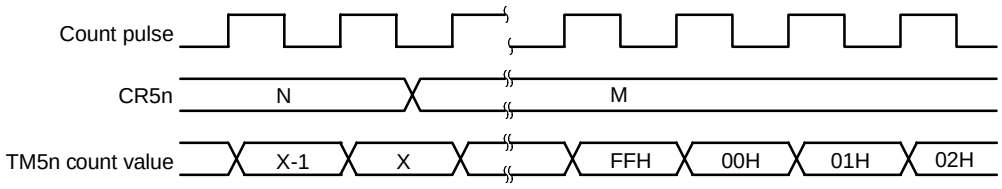
Remark: n = 50, 51

(3) Operation after the compare register is changed while the timer is counting

If the value after the 8-bit compare register (CR5n) changes is less than the value of the 8-bit timer register (TM5n), counting continues, overflows, and counting starts again from 0x00.
Consequently, when the value (M) after CR5n changes is less than the value (N) before the change, the timer must restart after CR5n changes.

(n = 0, 1).

Figure 6-62: Timing after compare register changes during timer count operation



Remarks: 1. N > X > M

2. n = 0, 1

Caution: Except when the TI5n input is selected, always set TCE = 0 before setting the stop state.

(4) TM5n read out during timer operation

Since reading out TM5n (n = 0, 1) during operation occurs while the selected clock is temporarily stopped, select some high or low level waveform that is longer than the selected clock.

[MEMO]

Chapter 7 Watch Timer

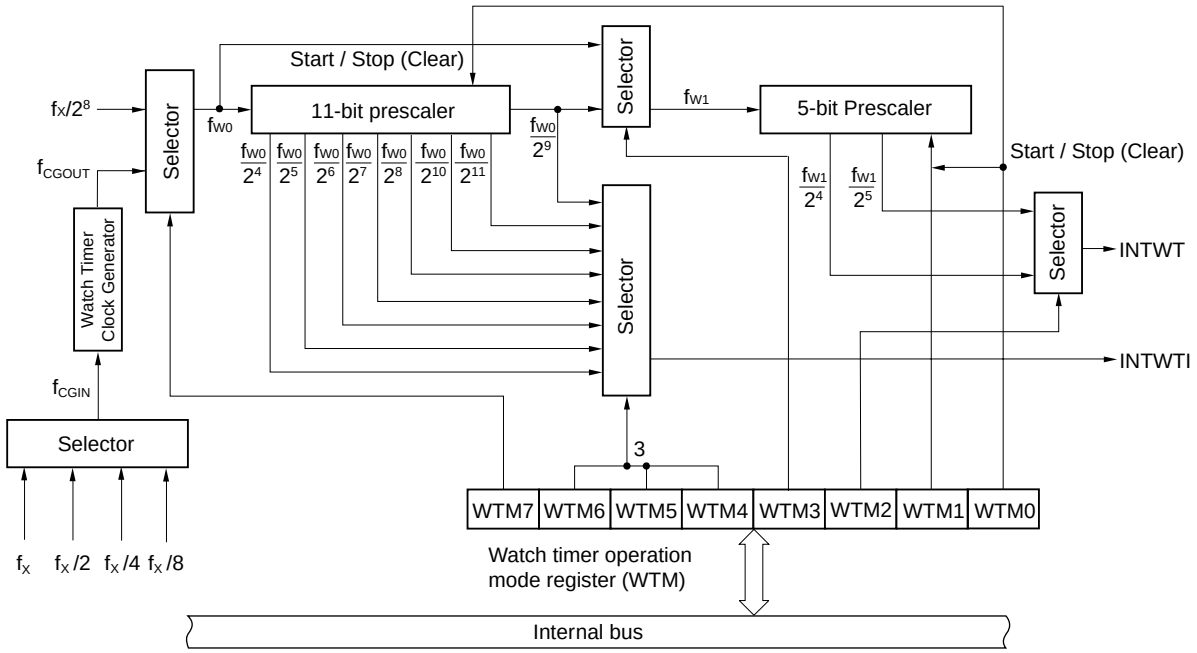
7.1 Function

The watch timer has the following functions:

- Watch timer (interrupt intervals from 8 µs up to 16.8 s)
- Interval timer (interrupt intervals from 8 µs up to 2.1 s)
- The watch timer and interval timer functions can be used at the same time.
- A special Watch timer Clock Generator avoids software based compensation for special interval times.

Figure 7-1 shows the block diagram of the watch timer.

Figure 7-1: Block Diagram of Watch Timer



(1) Watch timer

The watch timer generates an interrupt request (INTWT) at time intervals of 8 µs to 16.8 s by using the system clock prescaler $f_X/2^8$ or f_{CGOUT} (see Chapter **5.2 “Configuration” on page 141**).

(2) Interval timer

The interval timer generates an interrupt request (INTWTI) at time intervals of 8 µs to 2.1 s.

7.2 Configuration

The watch timer consists of the following hardware:

Table 7-1: Configuration of Watch Timer

Item	Configuration
Counter	5 bits × 1
Prescaler	11 bits × 1
Control register	Watch timer mode control register (WTM)

7.3 Watch Timer Control Register

The watch timer mode control register (WTM) controls the watch timer.

(1) Watch timer mode control register (WTM)

This register enables or disables the count clock and operation of the watch timer, sets the interval time of the prescaler, controls the operation of the 5-bit counter, and sets the set time of the watch flag.

WTM is set by a 1-bit or 8-bit memory manipulation instruction.

Caution: When the watch timer is used, the prescaler should not be cleared frequently. When rewriting WTM7 to WTM2 bit to other data, set WTM0 bit to "0" beforehand to stop the timer operation.

Figure 7-2: Watch Timer Mode Control Register (WTM)

	7	6	5	4	3	2	1	0	Address	R/W	After Reset
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0	FFFFF360H	R/W	00H

WTM7	Selects main input frequency from prescaler
0	$f_X / 2^8$ (f_{W0} = 15.6 KHz)
1	f_{CGOUT} (Watch Timer clock generator output frequency)

WTM6	WTM5	WTM4	11-bit Prescaler interval time selection for interval function
0	0	0	$2^4/f_{W0}$
0	0	1	$2^5/f_{W0}$
0	1	0	$2^6/f_{W0}$
0	1	1	$2^7/f_{W0}$
1	0	0	$2^8/f_{W0}$
1	0	1	$2^9/f_{W0}$
1	1	0	$2^{10}/f_{W0}$
1	1	1	$2^{11}/f_{W0}$

WTM3	WTM2	5-bit Prescaler interval time selection for watch timer function
0	0	$2^{14}/f_{W0}$
0	1	$2^{13}/f_{W0}$
1	0	$2^5/f_{W0}$
1	1	$2^4/f_{W0}$

Note: Calculated times for f_{CGOUT} = 32.786 KHz

WTM1	Controls Operation of Watch Timer function
0	Stops operation (clears 5-bit Counter after stop)
1	Starts operation of Watch Timer function if WTM0 bit = 1

WTM0	Controls Operation of the whole Watch Timer
0	Stops operation (clears both prescaler)
1	Starts operation of the 11-bit Prescaler

Remark: f_X: Main system clock oscillation frequency

7.4 Watch Timer Clock Generator

The Watch Timer Clock Generator provides a comfortable adjusting frequency (f_{CGOUT}) for the Watch Timer clock (f_{W0}). It allows the fine tuning of the system clock (f_X) with a compare register (PRSCM). Therefore no further software compensation is required to generate a special Watch Timer interval.

The PRSCM register is read/written by an 8-bit instruction.

Figure 7-3: Prescaler Compare Register (PRSCM)

	7	6	5	4	3	2	1	0	Address	R/W	After Reset
PRSCM	PRSCM7	PRSCM6	PRSCM5	PRSCM4	PRSCM3	PRSCM2	PRSCM1	PRSCM0	FFFFF36AH	R/W	00H

The configuration of the Watch Timer Clock Generator is done by the prescaler mode register (PRSM).

The PRSM register is read/written by an 8-bit instruction.

Caution: CE bit must be set to “0”, before changing the value of TODIS, BGCS1 or BGCS0.

Figure 7-4: Prescaler Mode Register

	7	6	5	4	3	2	1	0	Address	R/W	After Reset
PRSM	0	0	0	CE	0	TODIS	BRGCS1	BRGCS0	FFFFF368H	R/W	00H

CE	TODIS	Watch Timer Clock Generator Output
0	X	Fixed on low level
1	0	Active
1	1	Fixed on low level

Remark: X: Don’t Care
CE: Watch Timer Clock Generator enable
TODIS: Watch Timer Clock Generator output disable

BRGCS1	BRGCS0	Watch Timer Clock Generator Operating Frequency (f_{CGIN})
0	0	f_X
0	1	$f_X/2$
1	0	$f_X/4$
1	1	$f_X/8$

The output frequency of the watch timer clock generator f_{CGOUT} is calculated by the following method:

In case of PRSCM = N

$$f_{CGOUT} = \frac{f_{CGIN}}{N \times 2}$$

Remark: In case of PRSCM = 00H, N = 256

Example:

PRSCM register = 3DH (N = 61)
PRSM register = 10H ($f_{CGIN} = f_X$)

f_{CGOUT} = f_{CGIN} / (61 × 2)
= 4 MHz / 122
= 32.786 KHz

7.5 Operations

7.5.1 Operation as watch timer

The watch timer operates with time intervals from 8 µs to 16.8 s.
The watch timer generates at it overflow the INWT interrupt request at fixed time intervals.

The watch timer function is controlled with the WTM1 bit and the WTM0 bit of the WTM register.

With the WTM1 bit and the WTM0 bit the watch timer function can be started.
With the WTM1 bit the watch timer function can be stopped independently from the interval timer function.

For synchronous watch and interval timer function operation:
The count operation of the watch timer is started when the WTM0 bit and the WTM1 bit of the watch timer mode control register (WTM) are set to “1”.
Both prescalers are stopped and cleared if the WTM0 bit is set to “0“.

For independent start or stop of watch timer function operation:
This functionality is only available, when the 11-bit Prescaler is running, too.
The count operation of the watch timer is started when the WTM1 bit and the WTM0 bit of the watch timer mode control register (WTM) are set to “1”.
The WTM0 bit has to be set to “1” either it was “1” before. In that case the frequency of the running 11-bit prescaler is not influenced.
The 5-bit watch timer function prescaler is stopped and cleared if the WTM1 bit is set to “0“.

Caution: If the 5-bit watch timer function prescaler is clocked by $f_{W0}/2^9$ (WTM3 = 0) the prescaler is started with the next edge of the $f_X/2^9$ clock. Therefore if the 11-bit prescaler was running before the 5-bit prescaler watch timer is started, the INTWT interrupt is generated up to one $f_{W0}/2^9$ period later then the time that the WTM1 bit was set to “1” (if $f_{W0} = f_X/2^8$ this can be up to 32.8 ms). This happens only at the first starting edge of the $f_{W0}/2^9$ clock.

Table 7-2: Watch Timer (INTWT) Interval Times

WTM7	WTM3	WTM2	Count clock	Interval time
0	0	0	$f_X/2^{22}$	1.049 s
0	0	1	$f_X/2^{21}$	0.524 s
0	1	0	$f_X/2^{13}$	2.048 ms
0	1	1	$f_X/2^{12}$	1.024 ms
1	0	0	f_{CGOUT} **Note 1** $/2^{14}$	0.500 s **Note 2**
1	0	1	f_{CGOUT} **Note 1** $/2^{13}$	0.250 s **Note 2**
1	1	0	f_{CGOUT} **Note 1** $/2^5$	0.976 ms **Note 2**
1	1	1	f_{CGOUT} **Note 1** $/2^4$	0.488 ms **Note 2**

Notes: 1. Watch Timer clock generator Output Clock (f_{CGOUT}).

2. When the Watch Timer Counter Generator output (f_{CGOUT}) clock is set to 32.9 KHz.

7.5.2 Operation as interval timer

The watch timer can also be used as an interval timer that repeatedly generates an interrupt at intervals specified by a count value set in advance.

The interval timer function is controlled with the WTM0 bit. Therefore the interval timer and the watch timer function can be only stopped at the same time. If the WTM0 bit is set to “1”, the 11-bit prescaler and therefore the interval timer function is started.

The interval time can be selected by the WTM4 through WTM6 bits of the watch timer mode control register (WTM).

Table 7-3: Interval Timer (INTWTI) Interval Times

WTM7	WTM6	WTM5	WTM4	Count clock	Interval time
0	0	0	0	$f_X/2^{12}$	1.02 ms
0	0	0	1	$f_X/2^{13}$	2.05 ms
0	0	1	0	$f_X/2^{14}$	4.10 ms
0	0	1	1	$f_X/2^{15}$	8.19 ms
0	1	0	0	$f_X/2^{16}$	16.4 ms
0	1	0	1	$f_X/2^{17}$	32.8 ms
0	1	1	0	$f_X/2^{18}$	65.5 ms
0	1	1	1	$f_X/2^{19}$	131 ms
1	0	0	0	f_{CGOUT}**Note 1**$/2^4$	488 us**Note 2**
1	0	0	1	f_{CGOUT}**Note 1**$/2^5$	976 us**Note 2**
1	0	1	0	f_{CGOUT}**Note 1**$/2^6$	1.95 ms**Note 2**
1	0	1	1	f_{CGOUT}**Note 1**$/2^7$	3.90 ms**Note 2**
1	1	0	0	f_{CGOUT}**Note 1**$/2^8$	7.81 ms**Note 2**
1	1	0	1	f_{CGOUT}**Note 1**$/2^9$	15.6 ms**Note 2**
1	1	1	0	f_{CGOUT}**Note 1**$/2^{10}$	31.2 ms**Note 2**
1	1	1	1	f_{CGOUT}**Note 1**$/2^{11}$	62.5 ms**Note 2**

Notes: 1. Watch Timer clock generator Output Clock (f_{CGOUT}).

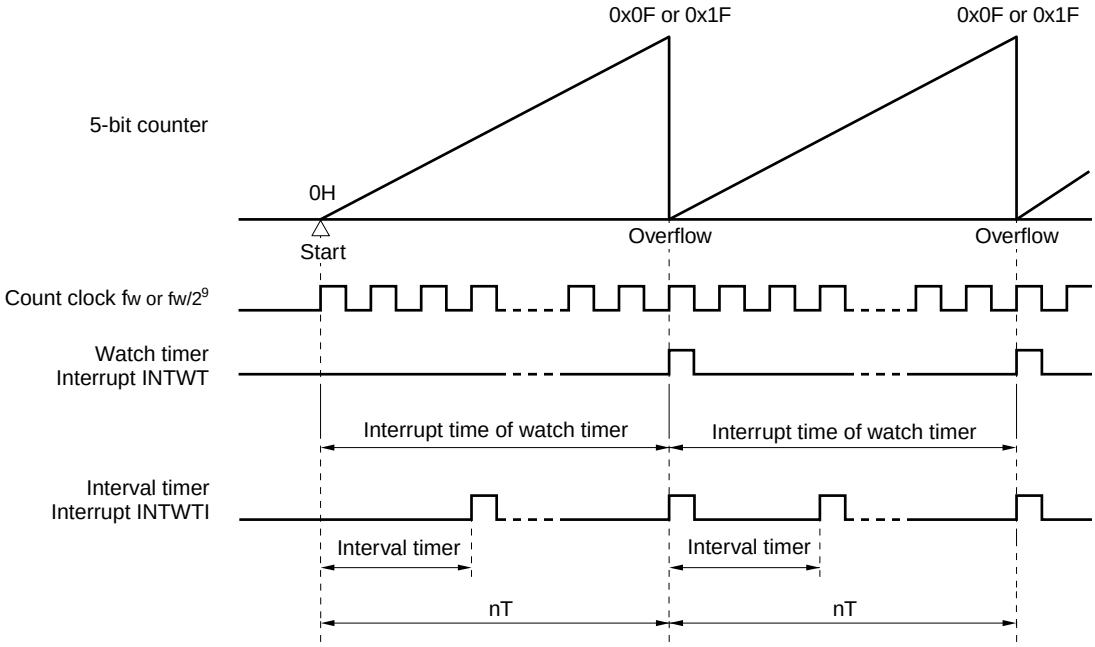
2. When f_{CGOUT} output clock is set to 32.9 KHz.

Remark: fx: Main system clock oscillation frequency.

7.5.3 Watch timer and Interval timer simultaneously

Operation of Watch timer and Interval timer can be used simultaneously.

Figure 7-5: Example Watch Timer and Interval Timer simultaneously



Remarks: **1.** f_W: Watch timer clock frequency

2. n : Interval timer operation numbers

Chapter 8 Watchdog Timer Function

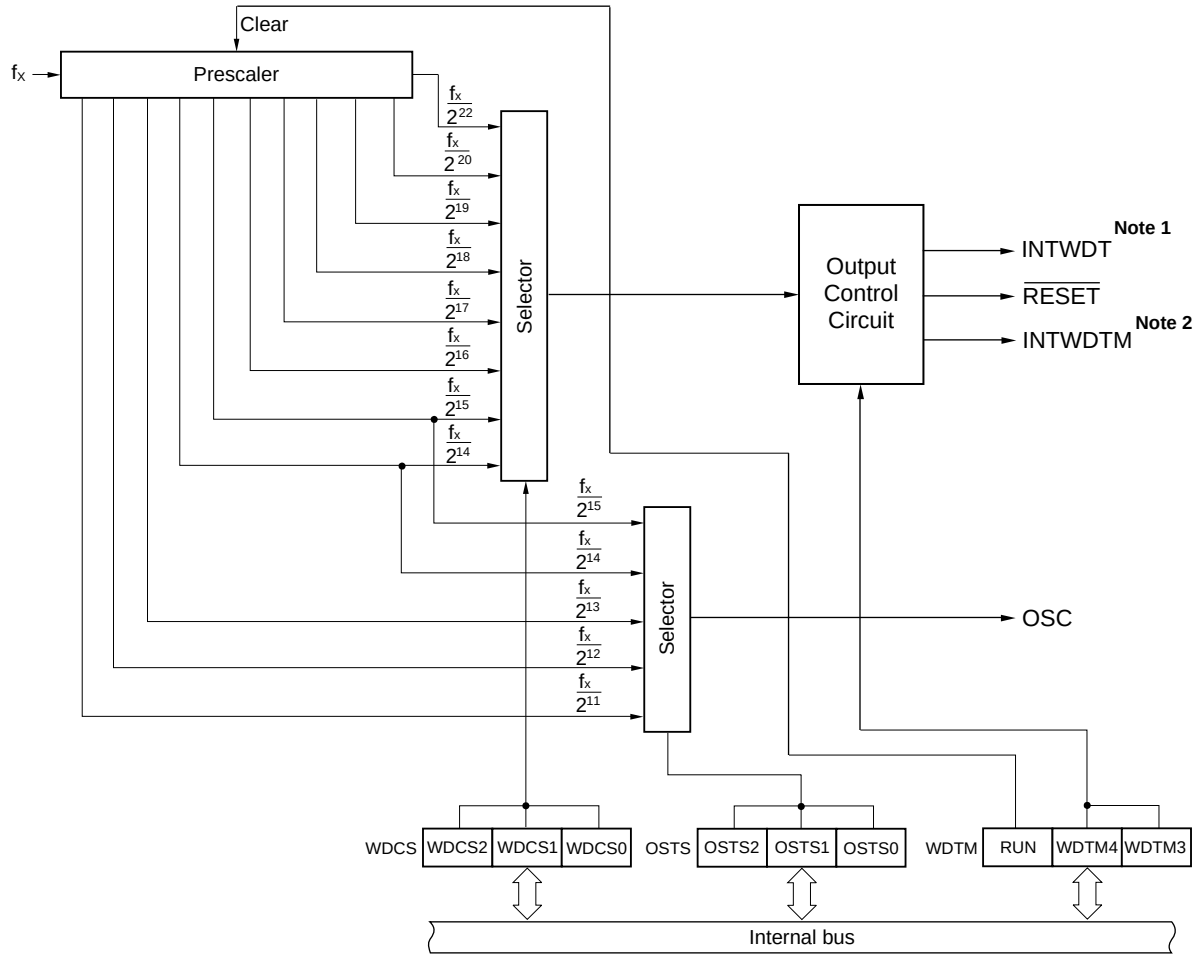
8.1 Functions

The watchdog timer has the following functions.

- Watchdog timer (hardware RESET)
- Watchdog timer (non maskable interrupt)
- Interval timer (maskable interrupt)

Caution: Use the watchdog timer mode register (WDTM) to select one of the 3 watchdog timer modes.

Figure 8-1: Block Diagram of Watchdog Timer



Notes: 1. In watchdog timer mode

2. In interval timer mode

Remark: f_X: Main clock frequency (4 MHz)

(1) Watchdog timer mode 2 (hardware RESET)

This mode detects program runaway. When runaway is detected, a hardware RESET is generated.

(2) Watchdog timer mode 1 (non maskable interrupt INTWDT)

This mode detects program runaway. When runaway is detected, the non-maskable interrupt INTWDT is generated.

(3) Interval timer mode

A maskable interrupt is generated at a preset time interval.

8.2 Configuration

The watchdog timer consists of the following hardware.

Table 8-1: Watchdog Timer Configuration

Item	Configuration
Control registers	Watchdog timer clock selection register (WDCS) Watchdog timer mode register (WDTM) Oscillation stabilization time selection register (OSTS)**Note**

Note: The functionality of the OSTS register is described in Chapter 5.4.5 “Oscillation stabilization time selection register (OSTS)” on page 147.

8.3 Watchdog Timer Control Register

The registers to control the watchdog timer is shown below.

- Watchdog timer clock selection register (WDCS)
- Watchdog timer mode register (WDTM)

(1) Watchdog timer clock selection register (WDCS)

This register selects the overflow times of the watchdog timer and the interval timer.

WDCS is set by an 8-bit memory manipulation instruction.

Figure 8-2: Watchdog Timer Clock Selection Register (WDCS)

	7	6	5	4	3	2	1	0	Address	R/W	After reset
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0	FFFF F382H	R/W	00H

WDCS2	WDCS1	WDCS0	Clock	Overflow/Interval Time
0	0	0	$2^{14}/f_X$	4.1 ms **Note**
0	0	1	$2^{15}/f_X$	8.2 ms
0	1	0	$2^{16}/f_X$	16.4 ms
0	1	1	$2^{17}/f_X$	32.8 ms
1	0	0	$2^{18}/f_X$	65.5 ms
1	0	1	$2^{19}/f_X$	131.1 ms
1	1	0	$2^{20}/f_X$	262.1 ms
1	1	1	$2^{22}/f_X$	1.05 s

Note: Reset Value

(2) Watchdog timer mode register (WDTM)

This register sets the operating mode of the watchdog timer, and enables and disables counting.

WDTM is set by an 8/1-bit memory manipulation instruction.

Figure 8-3: Watchdog Timer Mode Register (WDTM)

	7	6	5	4	3	2	1	0	Address	R/W	After reset
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FFFF F384H	R/W	00H

RUN	Operating Mode Selection for the Watchdog Timer **Note 1**
0	Disable count
1	Clear count and start counting

WDTM4	WDTM3	Watchdog Timer Operation Mode Selection **Note 2**
0	X	Interval timer mode (Maskable interrupt occurs upon generation of an overflow)
1	0	Watchdog timer mode 1 (Non-maskable interrupt occurs upon generation of an overflow)
1	1	Watchdog timer mode 2 (Reset operation is activated upon generation of an overflow)

Notes: 1. If RUN is set (1) once, the register cannot be cleared to “0” by software. Therefore, when the count starts, the count cannot be stopped except by $\overline{\text{RESET}}$ input.

2. Once set to “1”, WDTM3 and WDTM4 bit cannot be cleared to “0” by software.

Remark: X = don't care

Caution: If RUN is set to “1” and the watchdog timer is cleared, the actual overflow time may be up to $2^{12}/f_{XX}$ seconds less than the set time.

8.4 Operation

8.4.1 Operating as watchdog timer

(1) Watchdog Timer Mode 1 (non maskable interrupt INTWDT)

Set WDTM4 bit of the watchdog timer mode register (WDTM) to “1” and WDTM3 bit to “0” to operate as a watchdog timer in interrupt-request-mode to detect program runaway.
Setting RUN bit of WDTM register to “1” starts the count. After counting starts, if RUN bit is set to “1” again within the set time interval for runaway detection, the watchdog timer is cleared and counting starts again.
If RUN is not set to “1” and the runaway detection time has elapsed, a non-maskable interrupt (INTWDT) is generated (no reset functions).
The watchdog timer stops running in the STOP mode and WATCH mode. Consequently, set RUN to “1” and clear the watchdog timer before entering the STOP mode or WATCH mode. Do not set the watchdog timer when operating the HALT mode since the watchdog timer running in HALT mode.

For details of the possible time settings please refer to Figure 8-2, “Watchdog Timer Clock Selection Register (WDCS),” on page 253.

(2) Watchdog Timer Mode 2 (hardware RESET)

Set WDTM4 bit and WDTM3 bit of the watchdog timer mode register (WDTM) to “1” to operate as a watchdog timer in reset-request-mode to detect program runaway.

Setting RUN bit of WDTM to “1” starts the count. After counting starts, if RUN bit is set to “1” again within the set time interval for runaway detection, the watchdog timer is cleared and counting starts again.

If RUN bit is not set to “1” and the runaway detection time has elapsed, a reset functions is generated.
The watchdog timer stops running in the STOP mode and WATCH mode. Consequently, set RUN to “1” and clear the watchdog timer before entering the STOP mode or WATCH mode. Do not set the watchdog timer when operating the HALT mode since the watchdog timer running in HALT mode.

For details of the possible time settings please refer to Figure 8-2, “Watchdog Timer Clock Selection Register (WDCS),” on page 253.

Caution: The actual runaway detection time may be up to $2^{12}/f_{XX}$ seconds less than the set time.

8.4.2 Operating as interval timer (maskable interrupt INTWDTM)

Set WDTM4 bit to “0” in the watchdog timer mode register (WDTM) to operate the watchdog timer as an interval timer that repeatedly generates interrupts with a preset count value as the interval.

When operating as an interval timer, the interrupt mask flag (WDTMK bit) of the WDTIC register and the priority setting flags (WDTPR0 to WDTPR2 bit) become valid, and a maskable interrupt (INTWDTM) can be generated. The default priority of INTWDTM has the highest priority setting of the maskable interrupts.

The interval timer continues operating in the HALT mode and stops in the STOP and WATCH mode. Therefore, after the RUN bit of WDTM register is set to “1” and the interval timer is cleared before entering the STOP or WATCH mode, execute the STOP instruction.

For details of the possible time settings please refer to Figure 8-2, “Watchdog Timer Clock Selection Register (WDCS),” on page 253.

Cautions: 1. If WDTM4 bit of WDTM is set to “1” once (selecting the watchdog timer mode), the interval timer mode is not entered as long as $\overline{\text{RESET}}$ is not input.

2. The interval time immediately after being set by WDTM may be up to $2^{12}/f_{XX}$ seconds less than the set time.

Chapter 9 Serial Interface Function

9.1 Features

The serial interface function provides three types of serial interfaces combining a total of 7 transmit/receive channels. All seven channels can be used simultaneously.
The four interface formats are as follows.

(1) Asynchronous serial interfaces (UART50, UART51): 2 channels
(2) Clocked serial interfaces (CSI00, CSI01, CSI02): 3 channels
(3) DCAN controller: 2 channels

Remark: For details about the DCAN controller, refer to **Chapter 10 “DCAN” on page 317**.

UART50 and UART51, transmit/receive 1-byte serial data following a start bit and support full-duplex communication.
CSI00, CSI01 and CSI02 perform data transfer according to three types of signals, namely serial clocks ($\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK02}}$) serial inputs (SI00, SI01, SI02), and serial outputs (SO00, SO01, SO02) (3-wire serial I/O).

9.2 Asynchronous Serial Interfaces 50, 51 (UART50, UART51)

9.2.1 Features

- Transfer rate: 300 bps to 312,5 Kbps
  (using a dedicated baud rate generator and an internal system clock of 16 MHz)

- Full-duplex communications
  - On-chip reception buffer register (RXB5n)
  - On-chip transmission buffer register (TXB5n)

- Two-pin configuration
  - TXD5n: Transmit data output pin
  - RXD5n: Receive data input pin

- Reception error detection functions
  - Parity error
  - Framing error
  - Overrun error

- Interrupt sources: 3 types
  - Reception error interrupt (INTSER5n): Interrupt is generated according to the logical OR of the three types of reception errors.
  - Reception completion interrupt (INTSR5n): Interrupt is generated when receive data is transferred from the shift register to the reception buffer register after serial transfer is completed during a reception enabled state.
  - Transmission completion interrupt (INTST5n): Interrupt is generated when the serial transmission of transmit data (8 or 7 bits) from the shift register is completed.

- Character length: 7 or 8 bits

- Parity functions: Odd, even, 0, or none

- Transmission stop bits: 1 or 2 bits

- On-chip dedicated baud rate generator

Remark: n = 0, 1

9.2.2 Configuration

UART5n is controlled by the asynchronous serial interface mode register (ASIM5n), asynchronous serial interface status register (ASIS5n), and asynchronous serial interface transmission status register (ASIF5n). Receive data is maintained in the reception buffer register (RXB5n), and transmit data is written to the transmission buffer register (TXB5n).
Figure 9-1, “Asynchronous Serial Interfaces 50, 51 Block Diagram,” on page 260 shows the configuration of the asynchronous serial interface (UART5n) (n = 0, 1).

(1) Asynchronous serial interface mode registers 0, 1 (ASIM50, ASIM51) (n = 0, 1)

The ASIM5n register is an 8-bit register for specifying the operation of the asynchronous serial interface.

(2) Asynchronous serial interface status registers 0, 1 (ASIS50, ASIS51) (n = 0, 1)

The ASIS5n register consists of a set of flags that indicate the error contents when a reception error occurs. The various reception error flags are set (1) when a reception error occurs and are reset (0) when the ASIS5n register is read.

(3) Asynchronous serial interface transmission status registers 0, 1 (ASIF50, ASIF51) (n = 0, 1)

The ASIF5n register is an 8-bit register that indicates the status when a transmit operation is performed.
This register consists of a transmission buffer data flag, which indicates the hold status of TXB5n data, and the transmission shift register data flag, which indicates whether transmission is in progress.

(4) Reception control parity check

The receive operation is controlled according to the contents set in the ASIM5n register (n = 0, 1). A check for parity errors is also performed during a receive operation, and if an error is detected, a value corresponding to the error contents is set in the ASIS5n register (n = 0, 1).

(5) Reception shift register

This is a shift register that converts the serial data that was input to the RXD5n pin to parallel data. One byte of data is received, and if a stop bit is detected, the receive data is transferred to the reception buffer register (RXB5n).
This register cannot be directly manipulated.

(6) Reception buffer registers 0, 1 (RXB50, RXB51) (n = 0, 1)

RXB5n is an 8-bit buffer register for holding receive data. When 7 characters are received, 0 is stored in the MSB.
During a reception enabled state, receive data is transferred from the reception shift register to the RXB5n, synchronized with the end of the shift-in processing of one frame.
Also, the reception completion interrupt request (INTSR5n) is generated by the transfer of data to the RXB5n.

(7) Transmission shift register

This is a shift register that converts the parallel data that was transferred from the transmission buffer register (TXB5n) to serial data.
When one byte of data is transferred from the TXB5n, the shift register data is output from the TXD5n pin (n = 0, 1).
This register cannot be directly manipulated.

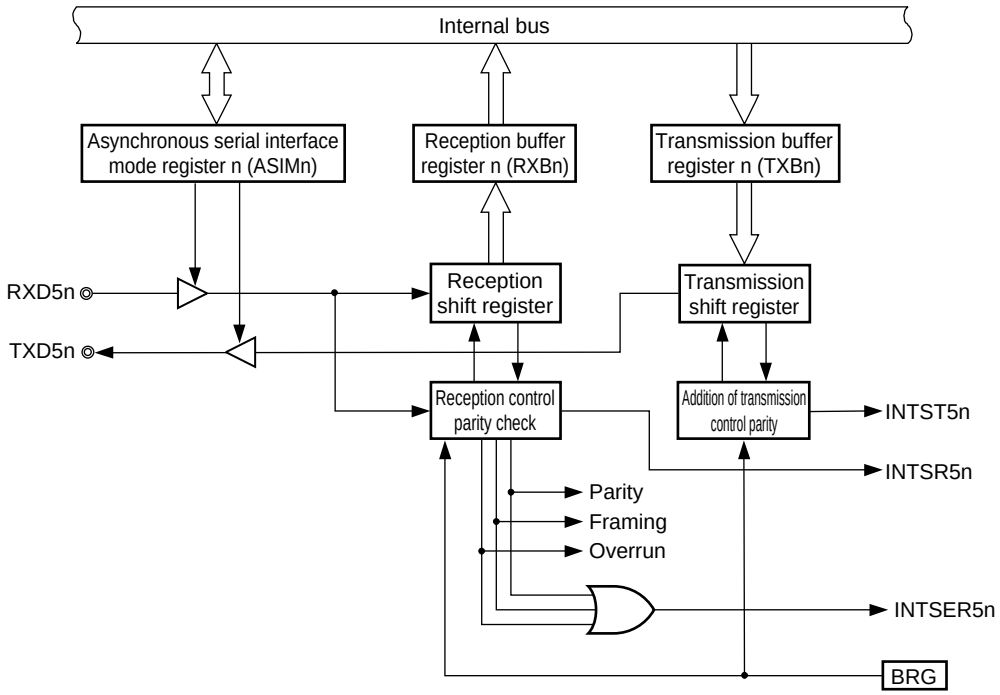
(8) Transmission buffer registers 0, 1 (TXB50, TXB51) (n = 0, 1)

TXB5n is an 8-bit buffer for transmit data. A transmit operation is started by writing transmit data to TXB5n. The transmission completion interrupt request (INTST5n) is generated synchronized with the completion of transmission of one frame.

(9) Addition of transmission control parity

A transmit operation is controlled by adding a start bit, parity bit, or stop bit to the data that is written to the TXB5n register, according to the contents that were set in the ASIM5n register (n = 0, 1).

Figure 9-1: Asynchronous Serial Interfaces 50, 51 Block Diagram



Remark: n = 0, 1

9.2.3 Control registers

(1) Asynchronous serial interface mode registers 0, 1 (ASIM50, ASIM51)

The ASIM5n register is an 8-bit register that controls the UART5n transfer operation.
This register can be read/written in 8 bit or 1-bit units (n = 0, 1).

Figure 9-2: Asynchronous Serial Interface Mode Registers 0, 1 (ASIM50, ASIM51) (1/3)

	7	6	5	4	3	2	1	0	Address	Initial value
ASIM50	CAE	TXE	RXE	PS1	PS0	CL	SL	ISRM	FFFFF320H	01H

	7	6	5	4	3	2	1	0	Address	Initial value
ASIM51	CAE	TXE	RXE	PS1	PS0	CL	SL	ISRM	FFFFF340H	01H

Bit Position	Bit Name	Function
7	CAE	Enables/disables clock operation. 0: Disable clock operation (reset internal circuit asynchronously.) 1: Enable clock operation UART5n operation clock control and asynchronous reset of the internal circuit are performed with the CAE bit. When the CAE bit is set to 0, the UART5n operation clock stops (fixed to low level), and an asynchronous reset is applied to internal UART5n latch. The TXD5n pin output is low level when the CAE bit = 0, and high level when the CAE bit = 1. Therefore, perform CAE setting in combination with port mode register (PM1, PM2, PM6) so as to avoid malfunction on the other side at start-up (Set the port to the output mode after setting the CAE bit to 1). Input from the RXD5n pin is fixed to high level with CAE bit = 0.
6	TXE	Enables/disables transfer. 0: Disable transfer (Perform synchronized reset of transfer circuit.) 1: Enable transfer **Cautions: 1. Set the TXE bit to 1 after setting the CAE bit to 1 when starting transfer. Set the CAE bit to 0 after setting the TXE bit to 0 when stopping transfer.** **2. To initialize the transfer unit, clear (0) the TXE bit, and after letting 2 Clock cycles (base clock) elapse, set (1) the TXE bit again. If the TXE bit is not set again, initialization may not be successful. (For details about the base clock, refer to 9.2.6 "Dedicated baud rate generators (BRG) of UART5n (n = 0, 1)" on page 278.)**

Figure 9-2: Asynchronous Serial Interface Mode Registers 0, 1 (ASIM50, ASIM51) (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>5</td><td>RXE</td><td>Enables/disables reception.
0: Disable reception (Perform synchronous reset of reception circuit)
1: Enable reception
Cautions: 1. Set the RXE bit to 1 after setting the CAE bit to 1 when starting transfer. Set the CAE bit to 0 after setting the RXE bit to 0 when stopping transfer.
2. To initialize the reception unit status, clear (0) the RXE bit, and after letting 2 Clock cycles (base clock) elapse, set (1) the RXE bit again. If the RXE bit is not set again, initialization may not be successful. (For details about the base clock, refer to 9.2.6 “Dedicated baud rate generators (BRG) of UART5n (n = 0, 1)” on page 278.)</td></tr>
<tr><td>4, 3</td><td>PS1, PS0</td><td>Controls parity bit.
<table>
<tr><th>PS1</th><th>PS0</th><th>Transmit Operation</th><th>Receive Operation</th></tr>
<tr><td>0</td><td>0</td><td>Don’t output parity bit</td><td>Receive with no parity</td></tr>
<tr><td>0</td><td>1</td><td>Output 0 parity</td><td>Receive as 0 parity</td></tr>
<tr><td>1</td><td>0</td><td>Output odd parity</td><td>Judge as odd parity</td></tr>
<tr><td>1</td><td>1</td><td>Output even parity</td><td>Judge as even parity</td></tr>
</table>
Cautions: 1. To overwrite the PS1 and PS0 bits, first clear (0) the TXE and RXE bits.
2. If “0 parity” is selected for reception, no parity judgment is performed. Therefore, no error interrupt is generated because the PE bit of the ASIS5n register is not set.
• Even parity
If the transmit data contains an odd number of bits with the value “1”, the parity bit is set (1). If it contains an even number of bits with the value “1”, the parity bit is cleared (0). This controls the number of bits with the value “1” contained in the transmit data and the parity bit so that it is an even number.
During reception, the number of bits with the value “1” contained in the receive data and the parity bit is counted, and if the number is odd, a parity error is generated.
• Odd parity
In contrast to even parity, odd parity controls the number of bits with the value “1” contained in the transmit data and the parity bit so that it is an odd number.
During reception, the number of bits with the value “1” contained in the receive data and the parity bit is counted, and if the number is even, a parity error is generated.</td></tr>
</table>

Remark: When reception is disabled, the reception shift register does not detect a start bit. No shift-in processing or transfer processing to the reception buffer register (RXB5n) is performed, and the contents of the RXB5n register are retained.
When reception is enabled, the reception shift operation starts, synchronized with the detection of the start bit, and when the reception of one frame is completed, the contents of the reception shift register are transferred to the RXB5n register. A reception completion interrupt (INTSR5n) is also generated in synchronization with the transfer to the RXB5n register.

Figure 9-2: Asynchronous Serial Interface Mode Registers 0, 1 (ASIM50, ASIM51) (3/3)

Bit Position	Bit Name	Function
4, 3	PS1, PS0	• 0 parity During transmission, the parity bit is cleared (0) regardless of the transmit data. During reception, no parity error is generated because no parity bit is checked. • No parity No parity bit is added to transmit data. During reception, the receive data is considered to have no parity bit. No parity error is generated because there is no parity bit.
2	CL	Specifies character length of transmit/receive data. 0: 7 bits 1: 8 bits **Caution: To overwrite the CL bit, first clear (0) the TXE and RXE bits.**
1	SL	Specifies stop bit length of transmit data. 0: 1 bit 1: 2 bits **Caution: To overwrite the SL bit, first clear (0) the TXE bit. Since reception is always done using a single stop bit, the SL bit setting does not affect receive operations.**
0	ISRM	Enables/disables generation of reception completion interrupt requests when an error occurs. 0: Generate a reception error interrupt request (INTSER5n) as an interrupt when an error occurs. In this case, no reception completion interrupt request (INTSR5n) is generated. 1: Generate a reception completion interrupt request (INTSR5n) as an interrupt when an error occurs. In this case, no reception error interrupt request (INTSER5n) is generated. **Caution: To overwrite the ISRM bit, first clear (0) the RXE bit.**

(2) Asynchronous serial interface status registers 0, 1 (ASIS50, ASIS51)

The ASIS5n register, which consists of 3-bit error flags (PE, FE and OVE), indicates the error status when UART5n reception is completed.

The status flag, which indicates a reception error, always indicates the status of the error that occurred most recently. That is, if the same error occurred several times before the receive data was read, this flag would hold only the status of the error that occurred last.

The ASIS5n register is cleared to 00H by a read operation. When a reception error occurs, the reception buffer register (RXB5n) should be read and the error flag should be cleared after the ASIS5n register is read.

This register is read-only in 8-bit units (n = 0, 1).

Caution: When the CAE bit or RXE bit of the ASIM5n register is set to 0, or when the ASIS50 register is read, the PE, FE, and OVE bits of the ASIS5n register are cleared (0).

Figure 9-3: Asynchronous Serial Interface Status Registers 0, 1 (ASIS50, ASIS51)

	7	6	5	4	3	2	1	0	Address	Initial value
ASIS50	0	0	0	0	0	PE	FE	OVE	FFFFF326H	00H

	7	6	5	4	3	2	1	0	Address	Initial value
ASIS51	0	0	0	0	0	PE	FE	OVE	FFFFF346H	00H

Bit Position	Bit Name	Function
2	PE	This is a status flag that indicates a parity error. 0: When the ASIM5n register's CAE and RXE bits are both set to 0, or when the ASIS5n register has been read 1: When reception was completed, the transmit data parity did not match the parity bit **Caution: The operation of the PE bit differs according to the settings of the PS1 and PS0 bits of the ASIM5n register.**
1	FE	This is a status flag that indicates a framing error. 0: When the ASIM5n register's CAE and RXE bits are both set to 0, or when the ASIS5n register has been read 1: When reception was completed, no stop bit was detected **Caution: For receive data stop bits, only the first bit is checked regardless of the number of stop bits.**
0	OVE	This is a status flag that indicates an overrun error. 0: When the ASIM5n register's CAE and RXE bits are both 0, or when the ASIS5n register has been read. 1: UART5n completed the next receive operation before reading the RXB5n receive data. **Caution: When an overrun error occurs, the next receive data value is not written to the RXB5n register and the data is discarded.**

(3) Asynchronous serial interface transmission status registers 0, 1 (ASIF50, ASIF51)

The ASIF5n register, which consists of 2-bit status flags, indicates the status during transmission. By writing the next data to the TXB5n register after data is transferred from the TXB5n register to the transmission shift register, transmit operations can be performed continuously without suspension even during an interrupt interval. When transmission is performed continuously, data should be written after referencing the ASIF5n register to prevent writing to the TXB5n register by mistake.

This register is read-only in 8-bit units (n = 0, 1).

Figure 9-4: Asynchronous Serial Interface Transmit Status Registers 0, 1 (ASIF50, ASIF51)

	7	6	5	4	3	2	1	0	Address	Initial value
ASIF50	0	0	0	0	0	0	TXBF	TXSF	FFFFF32AH	00H

	7	6	5	4	3	2	1	0	Address	Initial value
ASIF51	0	0	0	0	0	0	TXBF	TXSF	FFFFF34AH	00H

Bit Position	Bit Name	Function
1	TXBF	This is a transmission buffer data flag. 0: When the ASIM5n register's CAE or TXE bits is 0, or when data has been transferred to the transmission shift register (Data to be transferred next to TXB5n register does not exist). 1: Data exists in TXB5n register when the TXB5n register has been written to (Data to be transferred next exists in TXB5n register).
0	TXSF	This is a transmission shift register data flag. It indicates the transmission status of UART5n. 0: When the ASIM5n register's CAE or TXE bits is set to 0, or when following transfer completion, the next data transfer from the TXB5n register is not performed (waiting transmission) 1: When data has been transferred from the TXB5n register (Transmission in progress)

The following table shows relationships between the transmission status and write operations to TXB5n register.

TXBF	TXSF	Transmission Status	Write Operation to TXB5n
0	0	Initial status or transmission completed	Writing is permitted
0	1	Transmission in progress (no data is in TXB5n)	Writing is permitted
1	0	Waiting transmission (data is in TXB5n)	Writing is not permitted
1	1	Transmission in progress (data is in TXB5n)	Writing is not permitted

Caution: When transmission is performed continuously, data should be written to TXB5n register after confirming the TXBF bit value. If writing is not permitted, transmit data cannot be guaranteed when data is written to TXB5n register.

(4) Reception buffer registers 0, 1 (RXB50, RXB51)

The RXB5n register is an 8-bit buffer register for storing parallel data that had been converted by the reception shift register.

When reception is enabled (RXE bit = 1 in the ASIM5n register), receive data is transferred from the reception shift register to the RXB5n register, synchronized with the completion of the shift-in processing of one frame. Also, a reception completion interrupt request (INTSR5n) is generated by the transfer to the RXB5n register. For information about the timing for generating this interrupt request, refer to (4)“Receive operation” on page 274.

If reception is disabled (RXE bit = 0 in the ASIM5n register), the contents of the RXB5n register are retained, and no processing is performed for transferring data to the RXB5n register even when the shift-in processing of one frame is completed. Also, no reception completion interrupt is generated.

When 7 bits is specified for the data length, bits 6 to 0 of the RXB5n register are transferred for the receive data and the MSB (bit 7) is always 0. However, if an overrun error (OVE) occurs, the receive data at that time is not transferred to the RXB5n register.

Except when a reset is input, the RXB5n register becomes FFH even when CAE bit = 0 in the ASIM5n register.

This register is read-only in 8-bit units (n = 0, 1).

Figure 9-5: Reception Buffer Registers 0, 1 (RXB50, RXB51)

	7	6	5	4	3	2	1	0	Address	Initial value
RXB50	RXB7	RXB6	RXB5	RXB4	RXB3	RXB2	RXB1	RXB0	FFFFF324H	FFH

	7	6	5	4	3	2	1	0	Address	Initial value
RXB51	RXB7	RXB6	RXB5	RXB4	RXB3	RXB2	RXB1	RXB0	FFFFF344H	FFH

Bit Position	Bit Name	Function
7 to 0	RXB7 to RXB0	Stores receive data. 0 can be read for RXB7 when 7-bit or character data is received.

(5) Transmission buffer registers 0, 1 (TXB50, TXB51)

The TXB5n register is an 8-bit buffer register for setting transmit data.
When transmission is enabled (TXE bit = 1 in the ASIM5n register), the transmit operation is started by writing data to TXB5n register.
When transmission is disabled (TXE bit = 0 in the ASIM5n register), even if data is written to TXB5n register, the value is ignored.
The TXB5n register data is transferred to the transmission shift register, and a transmission completion interrupt request (INTST5n) is generated, synchronized with the completion of the transmission of one frame from the transmission shift register. For information about the timing for generating this interrupt request, refer to (2)“Transmit operation” on page 270.

Caution: When TXBF bit = 1 in the ASIF5n register, writing must not be performed to TXB5n register.

This register can be read or written in 8-bit units (n = 0, 1).

Figure 9-6: Transmission Buffer Registers 0, 1 (TXB50, TXB51)

	7	6	5	4	3	2	1	0	Address	Initial value
TXB50	TXB7	TXB6	TXB5	TXB4	TXB3	TXB2	TXB51	TXB50	FFFFF328H	FFH

	7	6	5	4	3	2	1	0	Address	Initial value
TXB51	TXB7	TXB6	TXB5	TXB4	TXB3	TXB2	TXB51	TXB50	FFFFF348H	FFH

Bit Position	Bit Name	Function
7 to 0	TXB7 to TXB50	Writes transmit data.

9.2.4 Interrupt requests

The following three types of interrupt requests are generated from UART50 and UART51.

- Reception error interrupt (INTSER5n)
- Reception completion interrupt (INTSR5n)
- Transmission completion interrupt (INTST5n)

The default priorities among these three types of interrupt requests is, from high to low, reception error interrupt, reception completion interrupt, and transmission completion interrupt (n = 0, 1).

Table 9-1: Generated Interrupts and Default Priorities

Interrupt	Priority
Reception error	1
Reception completion	2
Transmission completion	3

(1) Reception error interrupt (INTSER50, INTSER51)

When reception is enabled, a reception error interrupt is generated according to the logical OR of the three types of reception errors explained for the ASIS5n register. Whether a reception error interrupt (INTSER5n) or a reception completion interrupt (INTSR5n) is generated when an error occurs can be specified according to the ISRM bit of the ASIM5n register.
When reception is disabled, no reception error interrupt is generated.

(2) Reception completion interrupt (INTSR50, INTSR51)

When reception is enabled, a reception completion interrupt is generated when data is shifted in to the reception shift register and transferred to the reception buffer register (RXB5n).
A reception completion interrupt request can be generated in place of a reception error interrupt according to the ISRM bit of the ASIM5n register even when a reception error has occurred.
When reception is disabled, no reception completion interrupt is generated.

(3) Transmission completion interrupt (INTST50, INTST51)

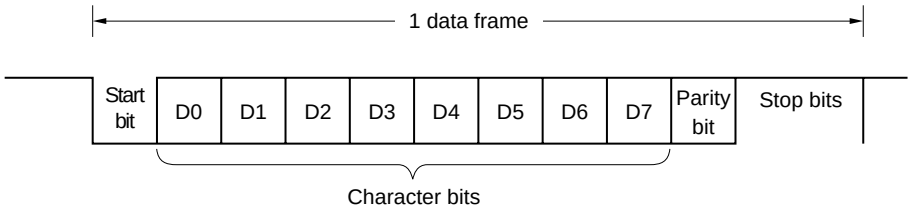
A transmission completion interrupt is generated when one frame of transmit data containing 7-bit or 8-bit characters is shifted out from the transmission shift register.

9.2.5 Operation

(1) Data format

Full-duplex serial data transmission and reception can be performed.
The transmit/receive data format consists of one data frame containing a start bit, character bits, a parity bit, and stop bits as shown in Figure 9-7.
The character bit length within one data frame, the type of parity, and the stop bit length are specified according to the asynchronous serial interface mode register (ASIM5n) (n = 0, 1).
Also, data is transferred with LSB first.

Figure 9-7: Asynchronous Serial Interface Transmit/Receive Data Format



- Start bit ⋯ 1 bit
- Character bits ⋯ 7 bits or 8 bits
- Parity bit ⋯ Even parity, odd parity, 0 parity, or no parity
- Stop bits ⋯ 1 bit or 2 bits

(2) Transmit operation

When CAE bit is set to 1 in the ASIM5n register, a high level is output from the TXD5n pin. Then, when TXE bit is set to 1 in the ASIM5n register, transmission is enabled, and the transmit operation is started by writing transmit data to transmission buffer register (TXB5n) (n = 0, 1).

(a) Transmission enabled state

This state is set by the TXE bit in the ASIM5n register.

- TXE = 1: Transmission enabled state
- TXE = 0: Transmission disabled state

Since UART5n does not have a CTS (transmission enabled signal) input pin, a port should be used to confirm whether the destination is in a reception enabled state.

(b) Starting a transmit operation

In transmission enabled state, a transmit operation is started by writing transmit data to transmission buffer register (TXB5n). When a transmit operation is started, the data in TXB5n is transferred to transmission shift register. Then, the transmission shift register outputs data to the TXD5n pin (the transmit data is transferred sequential starting with the start bit). The start bit, parity bit, and stop bits are added automatically.

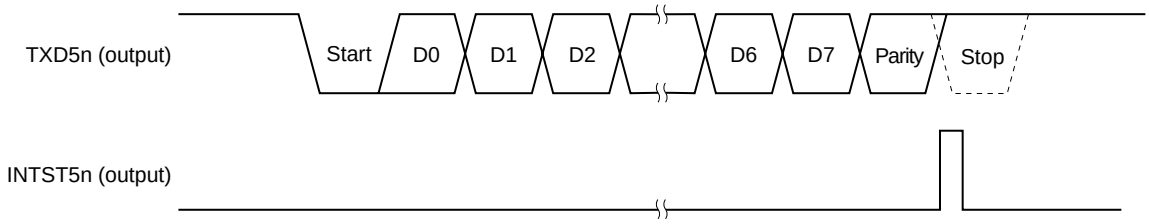
(c) Transmission interrupt request

When the transmission shift register becomes empty, a transmission completion interrupt request (INTST5n) is generated. The timing for generating the INTST5n interrupt differs according to the specification of the number of stop bits. The INTST5n interrupt is generated at the same time that the last stop bit is output.
If the data to be transmitted next has not been written to the TXB5n register, the transmit operation is suspended.

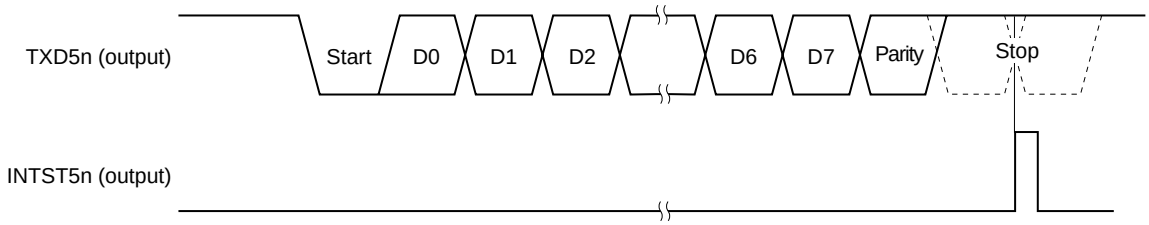
Caution: Normally, when the transmission shift register becomes empty, a transmission completion interrupt (INTST5n) is generated. However, no transmission completion interrupt (INTST5n) is generated if the transmission shift register becomes empty due to the input of a $\overline{\text{RESET}}$.

Figure 9-8: Asynchronous Serial Interface Transmission Completion Interrupt Timing (n = 0, 1)

(a) Stop bit length: 1



(b) Stop bit length: 2



(3) Continuous transmission operation

UART5n can write the next transmit data to the TXB5n register at the time that the transmission shift register starts the shift operation. This enables an efficient transmission rate to be realized by continuously transmitting data even during the INTST5n interrupt service after the transmission of one data frame.

When continuous transmission is performed, data should be written after referencing the ASIF5n register to confirm the transmission status and whether or not data can be written to the TXB5n register (n = 0, 1).

Caution: Transmit data should be written when the TXBF bit is 0. The transmission unit should be initialized when the TXSF bit is 0. If these actions are performed at other times, the transmit data cannot be guaranteed.

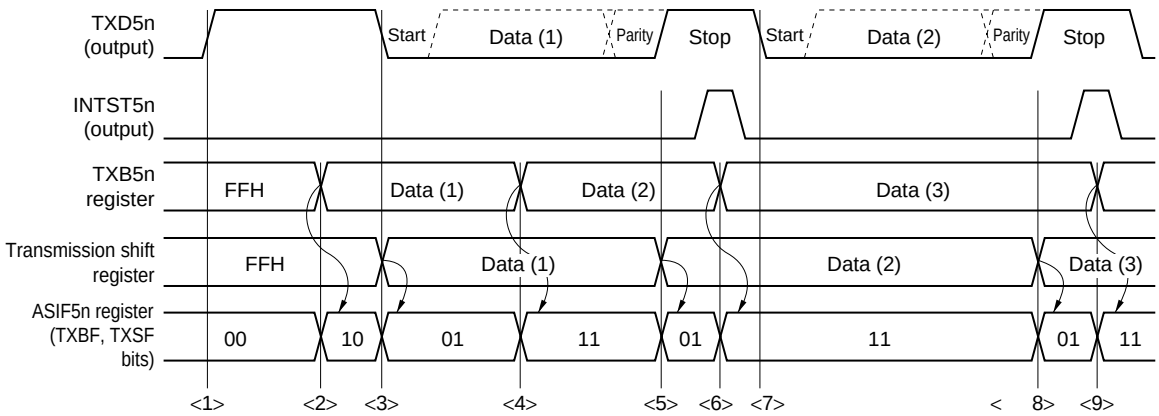
Table 9-2: Transmission Status and Whether or Not Writing Is Enabled

TXBF	TXSF	Transmission Status	Whether or not Write Operation to TXB5n is Enabled
0	0	Initial status or transmission completed	Writing is enabled
0	1	Transmission in progress (no data is in TXB5n register)	Writing is enabled
1	0	Awaiting transmission (data is in TXB5n register)	Writing is not enabled
1	1	Transmission in progress (data is in TXB5n register)	Writing is not enabled

(a) Starting procedure

Figure 9-9 shows the procedure to start continuous transmission.

Figure 9-9: Continuous Transmission Starting Procedure

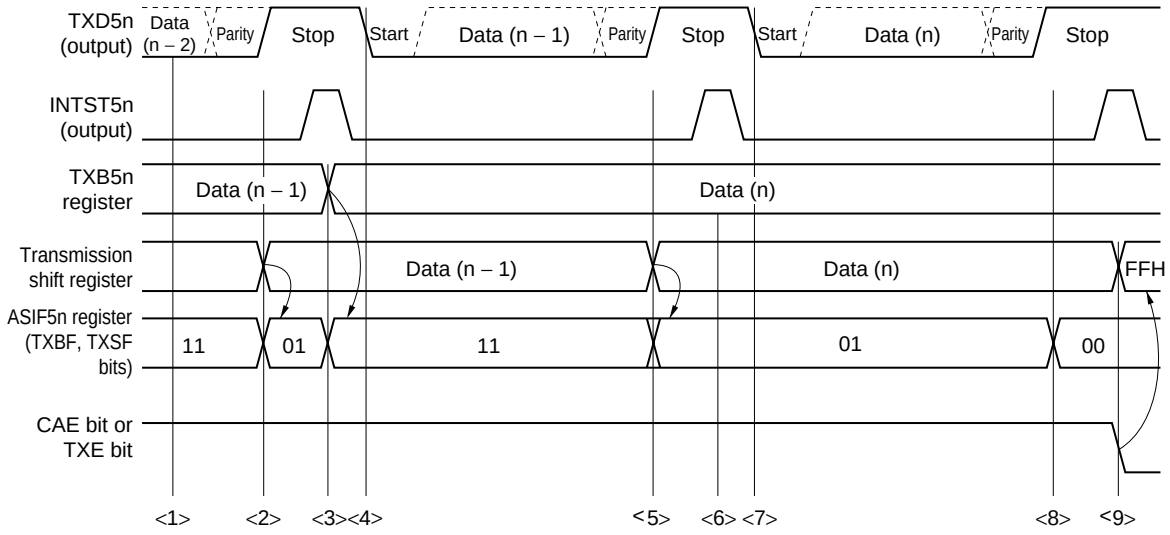


Transmission Starting Procedure	Internal Operation	ASIF5n Register	
		TXBF bit	TXSF bit
<1> Set transmission mode		0	0
<2> Write data (1) to TXB5n register		1	0
	<3> Generate start bit Start data (1) transmission[Note]	0	1
<4> Read ASIF5n register (confirm that TXBF bit = 0) Write data (2)		1	1
	<<Transmission in progress>> <5> Generate transmission completion interrupt (INTST5n)	0	1
<6> Read ASIF5n register (confirm that TXBF bit = 0) Write data (3)		1	1
	<7> Generate start bit Start data (2) transmission[Note]		
	<<Transmission in progress>> <8> Generate transmission completion interrupt (INTST5n)	0	1
<9> Read ASIF5n register (confirm that TXBF bit = 0) Write data (4)		1	1

Note: During the change from phase <2> to <3> (<6> to <7>) the ASIFn-register may be read either as 11H or 00H.

(b) Ending procedure

Figure 9-10: Continuous Transmission End Procedure



Transmission End Procedure	Internal Operation	ASIF5n Register	
		TXBF Bit	TXSF Bit
	<1> Transmission of data (n - 2) is in progress	1	1
	<2> Generate transmission completion interrupt (INTST5n)**Note**	0	1
<3> Read ASIF5n register (confirm that TXBF bit = 0) Write data (n)			
	<4> Generate start bit Start data (n - 1) transmission <<Transmission in progress>>	1	1
	<5> Generate transmission completion interrupt (INTST5n)**Note**		
<6> Read ASIF5n register (confirm that TXSF bit = 1) There is no write data		0	1
	<7> Generate start bit Start data (n) transmission <<Transmission in progress>>		
	<8> Generate transmission completion interrupt (INTST5n)		
<9> Read ASIF5n register (confirm that TXSF bit = 0)		0	0
Clear (0) the CAE bit or TXE bit of ASIM5n register	Initialize internal circuits		

Note: During the change from phase <2> to <3> (<4> to <5>) the ASIFn-register may be read either as 11H or 00H.

(4) Receive operation

An awaiting reception state is set by setting CAE bit to 1 in the ASIM5n register and then setting RXE bit to 1 in the ASIM5n register. To start a receive operation, detects a start bit first. The start bit is detected by sampling RXD5n pin. When the receive operation begins, serial data is stored sequential in the reception shift register according to the baud rate that was set. A reception completion interrupt (INTSR5n) is generated each time the reception of one frame of data is completed. Normally, the receive data is transferred from the reception buffer register (RXB5n) to memory by this interrupt servicing (n = 0, 1).

(a) Reception enabled state

The receive operation is set to reception enabled state by setting the RXE bit in the ASIM50 register to 1.

- RXE bit = 1: Reception enabled state
- RXE bit = 0: Reception disabled state

In reception disabled state, the reception hardware stands by in the initial state. At this time, the contents of the reception buffer register (RXB5n) are retained, and no reception completion interrupt or reception error interrupt is generated.

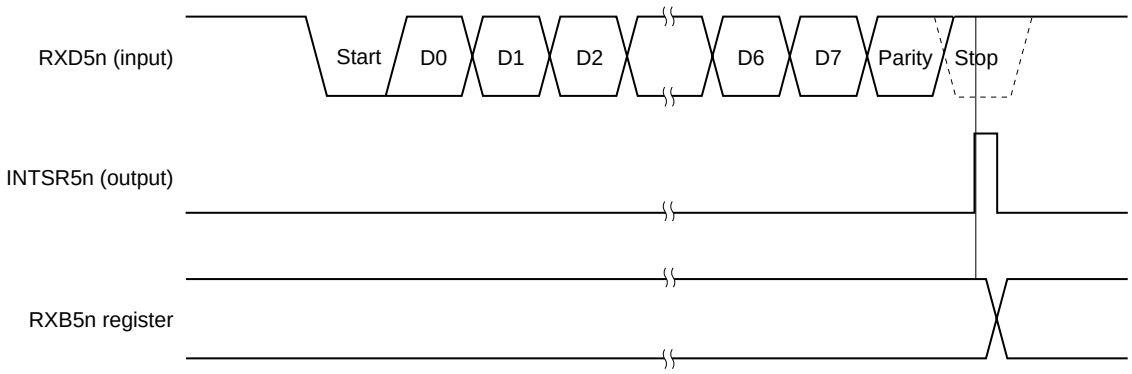
(b) Starting a receive operation

A receive operation is started by the detection of a start bit.
The RXD5n pin is sampled according to the serial clock from the dedicated baud rate generator (BRG) of UART5n (n= 0, 1).

(c) Reception completion interrupt

When RXE bit = 1 in the ASIM5n register and the reception of one frame of data is completed (the stop bit is detected), a reception completion interrupt (INTSR5n) is generated and the receive data within the reception shift register is transferred to RXB5n at the same time.
Also, if an overrun error (OVE) occurs, the receive data at that time is not transferred to the reception buffer register (RXB5n), and either a reception completion interrupt (INTSR5n) or a reception error interrupt (INTSER5n) is generated (the receive data within the reception shift register is transferred to RXB5n) according to the ISRM bit setting in the ASIM5n register.
Even if a parity error (PE) or framing error (FE) occurs during a reception operation, the receive operation continues until stop bit is received, and after reception is completed, either a reception completion interrupt (INTSR5n) or a reception error interrupt (INTSER5n) is generated according to the ISRM bit setting in the ASIM5n register.
If the RXE bit is reset (0) during a receive operation, the receive operation is immediately stopped. The contents of the reception buffer register (RXB5n) and of the asynchronous serial interface status register (ASIS5n) at this time do not change, and no reception completion interrupt (INTSR5n) or reception error interrupt (INTSER5n) is generated.
No reception completion interrupt is generated when RXE bit = 0 (reception is disabled).

Figure 9-11: Asynchronous Serial Interface Reception Completion Interrupt Timing



(5) Reception error

The three types of error that can occur during a receive operation are a parity error, framing error, or overrun error. The data reception result is that the various flags of the ASIS5n register are set (1), and a reception error interrupt (INTSER5n) or a reception completion interrupt (INTSR5n) is generated at the same time. The ISRM bit of the ASIM5n register specifies whether INTSER5n or INTSR5n is generated.

The type of error that occurred during reception can be detected by reading the contents of the ASIS5n register during the INTSER5n or INTSR5n interrupt servicing (n = 0, 1).

The contents of the ASIS5n register are reset (0) by reading it.

Table 9-3: Reception Error Causes

Error Flag	Reception Error	Cause
PE	Parity error	The parity specification during transmission did not match the parity of the reception data
FE	Framing error	No stop bit was detected
OVE	Overrun error	The reception of the next data was completed before data was read from the reception buffer register (RXB5n)

(a) Separation of reception error interrupt

A reception error interrupt can be separated from the INTSR5n interrupt and generated as an INTSER5n interrupt by clearing the ISRM bit of the ASIM5n register to 0.

Figure 9-12: When Reception Error Interrupt Is Separated from INTSR5n Interrupt (ISRM Bit = 0)

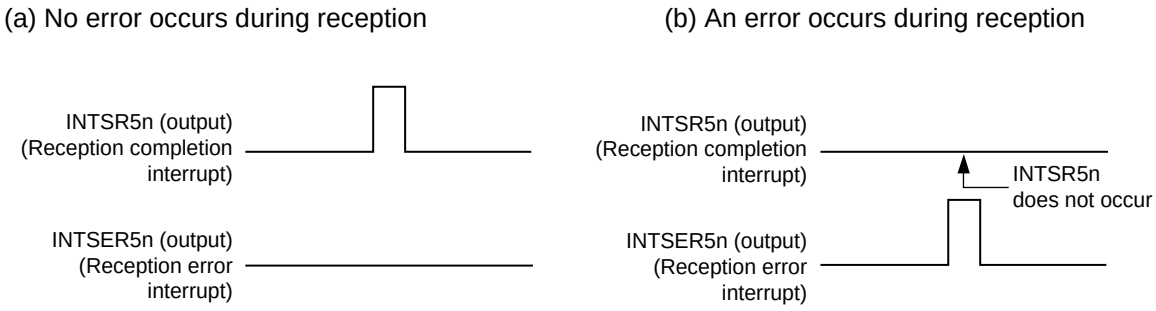
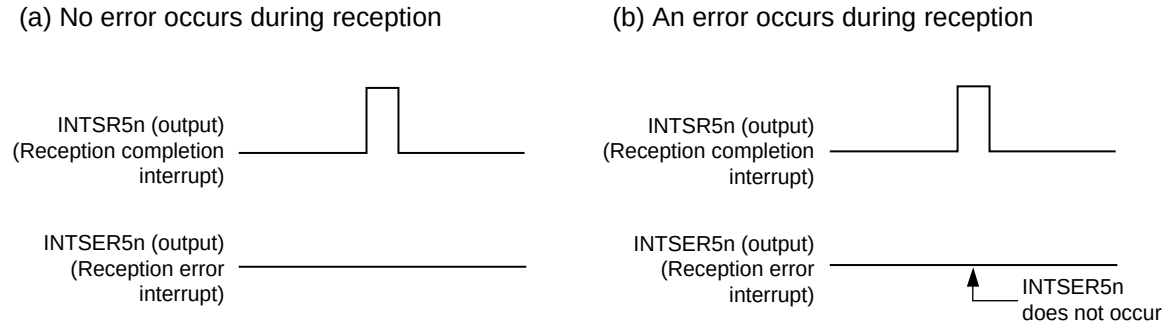


Figure 9-13: When Reception Error Interrupt Is Included in INTSR5n Interrupt (ISRM Bit = 1)



(6) Parity types and corresponding operation

A parity bit is used to detect a bit error in communication data. Normally, the same type of parity bit is used at the transmission and reception sides.

(a) Even parity

-During transmission
The parity bit is controlled so that the number of bits with the value “1” within the transmit data including the parity bit is even. The parity bit value is as follows.

- If the number of bits with the value “1” within the transmit data is odd: 1
- If the number of bits with the value “1” within the transmit data is even: 0

-During reception
The number of bits with the value “1” within the receive data including the parity bit is counted, and a parity error is generated if this number is odd.

(b) Odd parity

- During transmission
In contrast to even parity, the parity bit is controlled so that the number of bits with the value “1” within the transmit data including the parity bit is odd. The parity bit value is as follows.

- If the number of bits with the value “1” within the transmit data is odd: 0
- If the number of bits with the value “1” within the transmit data is even: 1

- During reception
The number of bits with the value “1” within the receive data including the parity bit is counted, and a parity error is generated if this number is even.

(c) 0 parity

During transmission the parity bit is set to “0” regardless of the transmit data.
During reception, no parity bit check is performed. Therefore, no parity error is generated regardless of whether the parity bit is “0” or “1”.

(d) No parity

No parity bit is added to the transmit data.
During reception, the receive operation is performed as if there were no parity bit. Since there is no parity bit, no parity error is generated.

(7) Receive data noise filter

The RXDn signal is sampled at the rising edge of the prescaler output basic clock (Clock). If the same sampling value is obtained twice, the match detector output changes, and this output is sampled as input data. Therefore, data not exceeding one clock width is judged to be noise and is not delivered to the internal circuit (see Figure 9-15). Refer to (a)“Basic clock (Clock)” on page 278 regarding the basic clock.

Also, since the circuit is configured as shown in Figure 9-4, internal processing during a receive operation is delayed by up to 2 clocks according to the external signal status.

Figure 9-14: Noise Filter Circuit

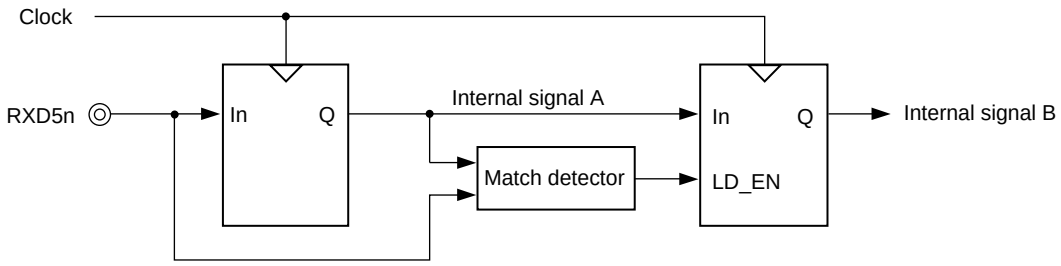
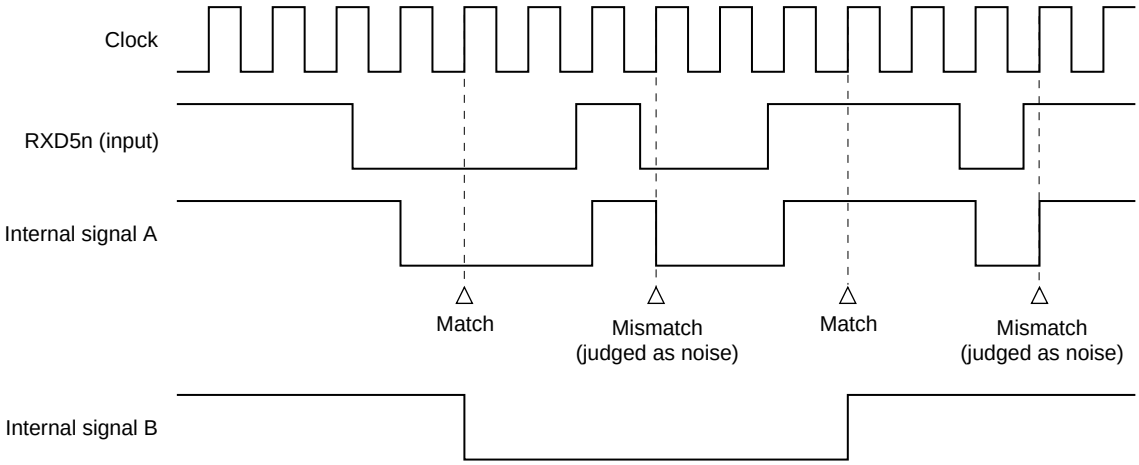


Figure 9-15: Timing of RXD5n Signal Judged as Noise



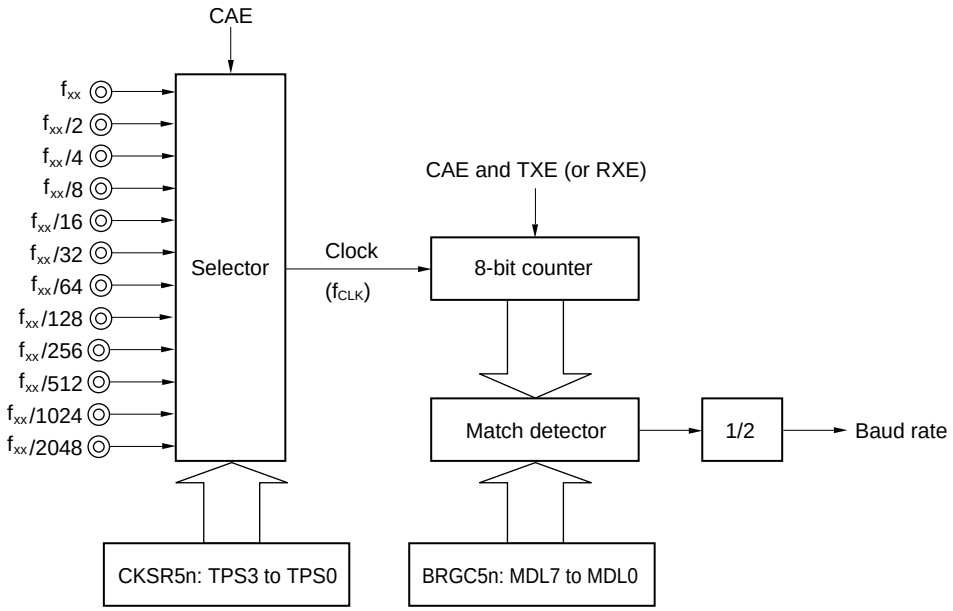
Remark: n = 0, 1

9.2.6 Dedicated baud rate generators (BRG) of UART5n (n = 0, 1)

A dedicated baud rate generator, which consists of a source clock selector and an 8-bit programmable counter, generates serial clocks during transmission/reception at UART5n (n = 0, 1). The dedicated baud rate generator output can be selected as the serial clock for each channel.
Separate 8-bit counters exist for transmission and for reception.

(1) Baud rate generator configuration

Figure 9-16: Baud Rate Generator (BRG) Configuration of UART5n (n = 0, 1)



Remark: n = 0, 1

(a) Basic clock (Clock)

When CAE bit = 1 in the ASIM5n register, the clock selected according to the TPS3 to TPS0 bits of the CKSRm register is supplied to the transmission/reception unit. This clock is called the basic clock (Clock), and its frequency is referred to as f_{CLK}. When CAE bit = 0, Clock is fixed at low level.

(2) Serial clock generation

A serial clock can be generated according to the settings of the CKSR5n and BRGCn registers.
The basic clock to the 8-bit counter is selected according to the TPS3 to TPS0 bits of the CKSR5n register.
The 8-bit counter divisor value can be set according to the MDL7 to MDL0 bits of the BRGCn register (n = 0, 1).

(a) Clock select registers 0, 1(CHKSR0 to CHKSR1)

The CKSR5n register is an 8-bit register for selecting the basic block according to the TPS3 to TPS0 bits. The clock selected by the TPS3 to TPS0 bits becomes the basic clock (Clock) of the transmission/ reception module. Its frequency is referred to as f_{CLK}.

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

Figure 9-17: Clock Select Registers 0, 1 (CHKSR0 to CHKSR1)

	7	6	5	4	3	2	1	0	Address	Initial value
CHKSR0	0	0	0	0	TPS3	TPS2	TPS1	TPS0	FFFFF32CH	00H

	7	6	5	4	3	2	1	0	Address	Initial value
CHKSR1	0	0	0	0	TPS3	TPS2	TPS1	TPS0	FFFFF34CH	00H

Bit Position	Bit Name	Function
3 to 0	TPS3 to TPS0	Specifies the basic clock

TPS3	TPS2	TPS1	TPS0	Basic Clock
0	0	0	0	f_{XX}
0	0	0	1	$f_{XX}/2$
0	0	1	0	$f_{XX}/4$
0	0	1	1	$f_{XX}/8$
0	1	0	0	$f_{XX}/16$
0	1	0	1	$f_{XX}/32$
0	1	1	0	$f_{XX}/64$
0	1	1	1	$f_{XX}/128$
1	0	0	0	$f_{XX}/256$
1	0	0	1	$f_{XX}/512$
1	0	1	0	$f_{XX}/1024$
1	1	X	X	Setting prohibited

Remark: f_{XX}: Internal system clock. X: don't care

(b) Baud rate generator control registers 0, 1(BRGC50, BRGC51)

The BRGC5n register is an 8-bit register that controls the baud rate (serial transfer speed) of UART5n.

This register can be read or written in 8-bit units (n = 0, 1).

Figure 9-18: Baud Rate Generator Control Registers 0, 1 (BRGC50, BRGC51)

	7	6	5	4	3	2	1	0	Address	Initial value
BRGC50	MDL7	MDL6	MDL5	MDL4	MDL3	MDL2	MDL1	MDL0	FFFFF32EH	FFH

	7	6	5	4	3	2	1	0	Address	Initial value
BRGC51	MDL7	MDL6	MDL5	MDL4	MDL3	MDL2	MDL1	MDL0	FFFFF34EH	FFH

Bit Position	Bit Name	Function									
		MDL7	MDL6	MDL5	MDL4	MDL3	MDL2	MDL1	MDL0	Divisor Value (k)	Serial Clock
7 to 0	MDL7 to MDL0	0	0	0	0	0	x	x	x	–	Setting prohibited
		0	0	0	0	1	0	0	0	8	$f_{CLK}/8$
		0	0	0	0	1	0	0	1	9	$f_{CLK}/9$
		0	0	0	0	1	0	1	0	10	$f_{CLK}/10$
		:	:	:	:	:	:	:	:	:	:
		1	1	1	1	1	0	1	0	250	$f_{CLK}/250$
		1	1	1	1	1	0	1	1	251	$f_{CLK}/251$
		1	1	1	1	1	1	0	0	252	$f_{CLK}/252$
		1	1	1	1	1	1	0	1	253	$f_{CLK}/253$
		1	1	1	1	1	1	1	0	254	$f_{CLK}/254$
		1	1	1	1	1	1	1	1	255	$f_{CLK}/255$

Caution: If the MDL7 to MDL0 bits are to be overwritten, TXE bit and RXE bit should be set to 0 in the ASIM5n register first.

Remarks: 1. f_{CLK}: Frequency [Hz] of basic clock selected according to TPS3 to TPS0 bits of CKSRm register

2. k: Value set according to MDL7 to MDL0 bits (k = 8, 9, 10,..., 255)

3. The baud rate is the output clock for the 8-bit counter divided by 2

4. x: don't care.

(c) Baud rate

The baud rate is the value obtained according to the following formula.

$$\text{Baud rate} = \frac{f_{CLK}}{2 \cdot k} \quad [bps]$$

f_{CLK} = Frequency [Hz] of basic clock selected according to TPS3 to TPS0 bits of CKSRm register.

k = Value set according to MDL7 to MDL0 bits of BRGC5n register (k = 8, 9, 10,..., 255)

(d) Baud rate error

The baud rate error is obtained according to the following formula.

$$Error = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (normal baud rate)}} - 1 \right) \times 100 \quad [\%]$$

Cautions: 1. Make sure that the baud rate error during transmission does not exceed the allowable error of the reception destination.

2. Make sure that the baud rate error during reception is within the allowable baud rate range during reception, which is described in chapter (3)“Allowable baud rate range during reception” on page 283.

Example: Basic clock frequency = 16 MHz / 8 = 2,000,000 Hz
Settings of MDL7 to MDL0 bits in BRGC50 register = 00001101B (k = 13)
Target baud rate = 153,600 bps

Baud rate = 16M/(2 × 65)
= 16000000/(2 × 65) = 153,846 [bps]

Error = (1553846/153600 - 1) × 100
= 0.160 [%]

(e) Baud rate setting example

Table 9-4: Baud Rate Generator Setting Data

Baud Rate [bps]	f_{XX} = 16 MHz		
	CKSR5n f_{CLK}	BRGC5n k	ERR [%]
300	08H	68H	0.16
600	07H	68H	0.16
1200	06H	68H	0.16
2400	08H	0DH	0.16
4800	07H	0DH	0.16
9600	06H	0DH	0.16
10400	06H	0CH	0.16
19200	05H	0DH	0.16
24000	04H	15H	-0.79
31250	05H	08H	0.0
33600	04H	15H	-0.79
38400	04H	0DH	0.16
48000	03H	15H	-0.79
56000	04H	09H	-0.79
62500	04H	08H	0.00
76800	03H	0DH	0.16
115200	02H	11H	2.12
153600	02H	0DH	0.16
312500	01H	0DH	-1.54

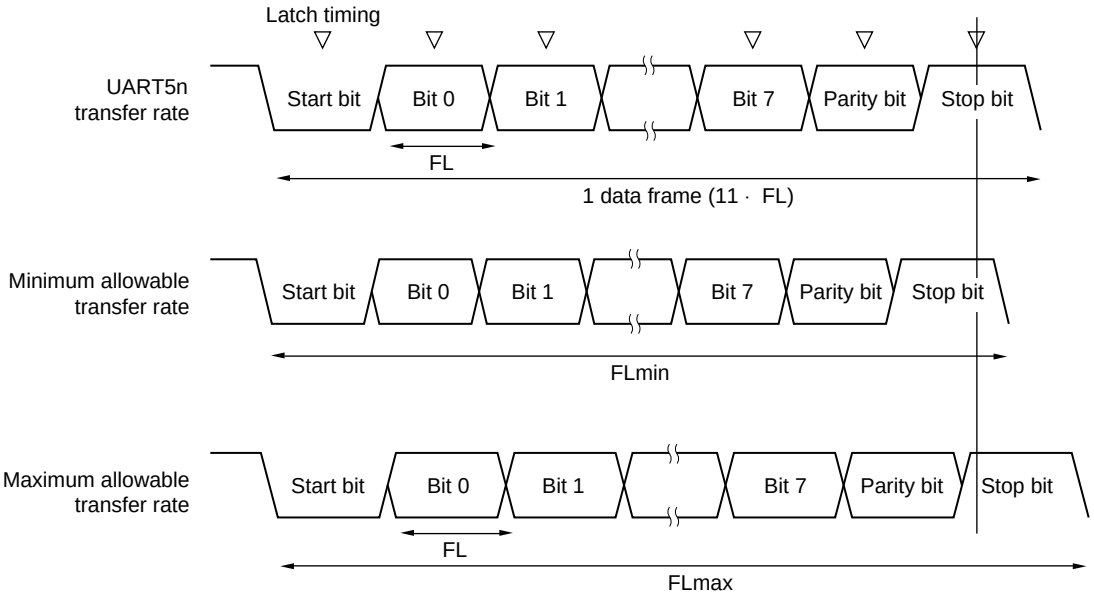
Remark: f_{XX}: Basic clock frequency
ERR: Baud rate error [%]

(3) Allowable baud rate range during reception

The degree to which a discrepancy from the transmission destination's baud rate is allowed during reception is shown below.

Caution: The equations described below should be used to set the baud rate error during reception so that it always is within the allowable error range.

Figure 9-19: Allowable Baud Rate Range During Reception



As shown in Figure 9-19, after the start bit is detected, the receive data latch timing is determined according to the counter that was set by the BRGC5n register. If all data up to the final data (stop bit) is in time for this latch timing, the data can be received normally.
Applying this to 11-bit reception is, theoretically, as follows.

FL = BR $^{-1}$

BR: UART5n baud rate
k: BRGC5n register setting value
FL: 1-bit data length

When the latch timing margin is made 2 basic clocks (Clock), the minimum allowable transfer rate (FLmin) is as follows.

$$\text{FLmin} = 11 \times \text{FL} - \frac{k-2}{2k} \times \text{FL} = \frac{21k+2}{2k} \times FL$$

Therefore, the transfer destination's maximum baud rate (BRmax) that can be received is as follows.

$$\text{BRmax} = \left(\frac{\text{FLmin}}{11}\right)^{-1} = \frac{22k}{21k+2} \times BR$$

Similarly, the maximum allowable transfer rate (FLmax) can be obtained as follows.

$$\frac{10}{11} \times \text{FLmax} = 11 \times \text{FL} - \frac{k+2}{2k} \times FL = \frac{21k-2}{2k} \times FL$$

$$\text{FLmax} = \frac{21k-2}{20k} \times FL \times 11$$

Therefore, the transfer destination's minimum baud rate (BRmin) that can be received is as follows.

$$\text{BRmin} = \left(\frac{\text{FLmax}}{11}\right)^{-1} = \frac{22k}{21k-2} \times \text{BR}$$

The allowable baud rate error of UART5n and the transfer destination can be obtained as follows from the expressions described above for computing the minimum and maximum baud rate values.

Table 9-5: Maximum and Minimum Allowable Baud Rate Error

Division 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%

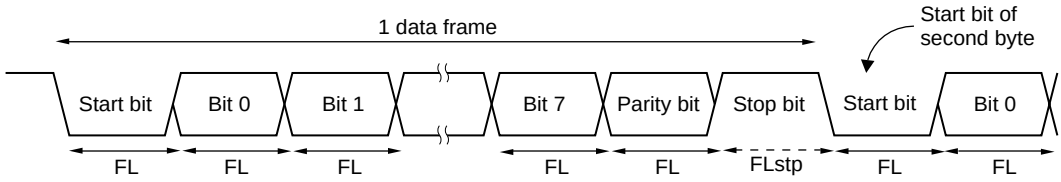
Remarks: 1. The reception precision depends on the number of bits in one frame, the basic clock frequency, and the division ratio (k). The higher the basic clock frequency and the larger the division ratio (k), the higher the precision.

2. k: BRGC5n setting value

(4) Transfer rate during continuous transmission

During continuous transmission, the transfer rate from a stop bit to the next start bit is extended two clocks of basic clock (Clock) longer than normal. However, on the reception side, the transfer result is not affected since the timing is initialized by the detection of the start bit.

Figure 9-20: Transfer Rate During Continuous Transmission



Representing the 1-bit data length by FL, the stop bit length by FLstp, and the basic clock frequency by f_{CLK} fields the following equation.

$$FLstp = FL + 2/f_{CLK}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL = 2/f_{CLK}$$

9.2.7 Precautions

When the supply of clocks to UART5n (n = 0, 1) is stopped (for example, IDLE or STOP mode), operation stops with each register retaining the value it had immediately before the supply of clocks was stopped. The TXD5n pin output also holds and outputs the value it had immediately before the supply of clocks was stopped. However, operation is not guaranteed after the supply of clocks is restarted. Therefore, after the supply of clocks is restarted, the circuits should be initialized by setting CAE bit = 0, RXE bit = 0, and TXE bit = 0 in the ASIM5n register.

9.3 Clocked Serial Interfaces 0 to 2 (CSI00, CSI01, CSI02)

9.3.1 Features

- High-speed transfer: Maximum 4 Mbps (at 16 MHz system clock)
- Master mode or slave mode can be selected
- Transmission data length: 8 bits or 16 bits
- Transfer data direction can be switched between MSB first and LSB first
- Eight clock signals can be selected (7 master clocks and 1 slave clock)
- 3-wire type
  - SO0n :Serial transmit data output
  - SI0n :Serial transmit data input
  - $\overline{\text{SCK0n}}$:Serial clock input/output
- Interrupt sources: 1 type
  - Transmission/reception completion interrupt (INTCSI0n)
- Transmission/reception mode and reception-only mode can be specified
- 3 buffers for each of the 3 CSI-channels are provided on chip:
  - first stage transmission buffer (SOTBF0n/SOTBFL0n)
  - transmission buffer (SOTB0n/SOTBL0n)
  - reception buffers (SIRB0n/SIRBL0n)
- Single transfer mode and repeat transfer mode can be specified

Remark: n = 0 to 2

9.3.2 Configuration

CSI0n is controlled via the clocked serial interface mode register (CSIM0n) (n = 0 to 2).
Transmission/reception of data is performed with reading SIO0n register (n = 0 to 2).

(1) Clocked serial interface mode registers (CSIM00, CSIM01, CSIM02)

The CSIM0n register is an 8-bit register that specifies the operation of CSI0n.

(2) Clocked serial interface clock selection registers (CSICK01, CSICK01, CSICK02)

The CSICK0n register is an 8-bit register that controls the CSI0n serial transfer operation.

(3) Serial I/O shift registers (SIO00, SIO01, SIO02)

The SIO0n register is a 16-bit shift register that converts parallel data into serial data.
The SIO0n register is used for both transmission and reception.
Data is shifted in (reception) and shifted out (transmission) from the MSB or LSB side.
The actual transmission/reception operations are started up by access of the buffer register.

(4) Serial I/O LSB shift registers (SIOL00, SIOL01, SIOL02)

The SIOL0n register is an 8-bit shift register that converts parallel data into serial data.
The SIOL0n register is used for both transmission and reception.
Data is shifted in (reception) and shifted out (transmission) from the MSB or LSB side.
The actual transmission/reception operations are started up by access of the buffer register.

(5) Clocked serial interface reception buffer registers (SIRB00, SIRB01, SIRB02)

The SIRB0n register is a 16-bit buffer register that stores receive data.

(6) Clocked serial interface LSB reception buffer registers (SIRBL00, SIRBL01, SIRBL02)

The SIRBL0n register is an 8-bit buffer register that stores receive data.

(7) Clocked serial interface transmission buffer registers (SOTB00, SOTB01, SOTB02)

The SOTB0n register is a 16-bit buffer register that stores transmit data.

(8) Clocked serial interface LSB transmission buffer registers (SOTBL00, SOTBL01, SOTBL02)

The SOTBL0n register is an 8-bit buffer register that stores transmit data.

(9) Clocked serial interface initial transmission buffer registers (SOTBF00, SOTBF01, SOTBF02)

The SOTBF0n register is a 16-bit buffer register that stores the initial transmit data in the repeat transfer mode.

(10) Clocked serial interface LSB initial transmission buffer registers (SOTBFL00, SOTBFL01, SOTBFL02)

The SOTBFL0n register is an 8-bit buffer register that stores initial transmit data in the repeat transfer mode.

(11) Selector

The selector selects the serial clock to be used.

(12) Serial clock control circuit

Controls the serial clock supply to the shift register. Also controls the clock output to the $\overline{\text{SCK0n}}$ pin when the internal clock is used.

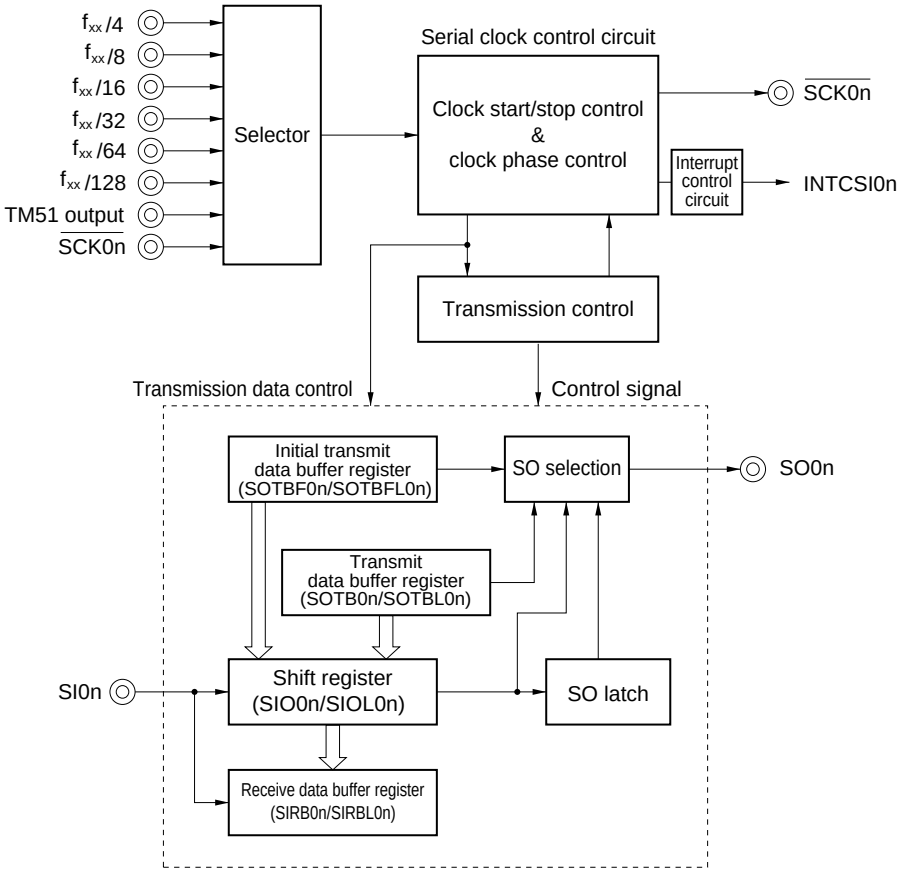
(13) Serial clock counter

Counts the serial clock output or input during transmission/reception operation, and checks whether 8-bit data transmission/reception has been performed.

(14) Interrupt control circuit

Controls the interrupt request timing.

Figure 9-21: Block Diagram of Clocked Serial Interfaces



Remark: n = 0 to 2

9.3.3 Control registers

(1) Clocked serial interface mode registers 0, 1, 2 (CSIM00, CSIM01, CSIM02)

The CSIM0n register controls the CSI0n operation (n = 0 to 2).

These registers can be read/written in 8-bit or 1-bit units (however, bit 1 is read-only).

Figure 9-22: Clocked Serial Interface Mode Registers (CSIM00, CSIM01, CSIM02) (1/2)

	7	6	5	4	3	2	1	0	Address	Initial value
CSIM00	CSIE	TRMD	CCL	DIR	CSIT	AUTO	0	CSOT	FFFF F2C0H	00H
CSIM01	CSIE	TRMD	CCL	DIR	CSIT	AUTO	0	CSOT	FFFF F2E0H	00H
CSIM02	CSIE	TRMD	CCL	DIR	CSIT	AUTO	0	CSOT	FFFF F300H	00H

Figure 9-22: Clocked Serial Interface Mode Registers (CSIM00, CSIM01, CSIM02) (2/2)

Bit Position	Bit Name	Function
7	CSIE	Enables/disables CSIn operation. 0: Disable CSIn operation. 1: Enable CSIn operation. The internal CSIn circuit can be reset asynchronously by setting the CSIE bit to 0. For the $\overline{SCK0n}$ and SO0n pin output status when the CSIE bit = 0, refer to 9.3.5 “Output pins” on page 314
6	TRMD	Specifies transmission/reception mode. 0: Receive-only mode 1: Transmission/reception mode When the TRMD bit = 0, receive-only transfer is performed and the SO0n pin output is fixed to low level. Data reception is started by reading the SIRB0n register. When the TRMD bit = 1, transmission/reception is started by writing data to the SOTB0n register.
5	CCL	Specifies data length. 0: 8 bits 1: 16 bits
4	DIR	Specifies transfer direction mode (MSB/LSB). 0: First bit of transfer data is MSB 1: First bit of transfer data is LSB
3	CSIT	Controls delay of interrupt request signal. 0: No delay 1: Delay mode (interrupt request signal is delayed 1/2 cycle). **Caution: The delay mode (CSIT bit = 1) is effective only in the master mode (CKS2 to CSK0 bits of the CSICn register are not 111B). In the slave mode (CKS2 to CKS0 bits are 111B), do not set the delay mode.**
2	AUTO	Specifies single transfer mode or repeat transfer mode. 0: Single transfer mode 1: Repeat transfer mode
0	CSOT	Flag indicating transfer status. 0: Idle status 1: Transfer execution status **Caution: The CSOT bit is cleared (0) by writing 0 to the CSIE bit.**

Remark: n = 0 to 2

Caution: Overwriting the TRMD, CCL, DIR, CSIT, and AUTO bits of the CSIM0n register can be done only when the CSOT bit = 0. If these bits are overwritten at any other time, the operation cannot be guaranteed.

(2) Clocked serial interface clock selection registers 0, 1, 2 (CSICK0n)

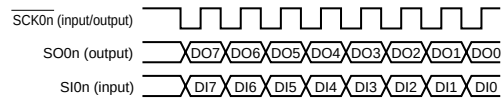
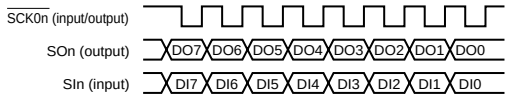
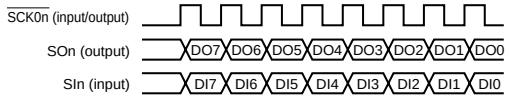
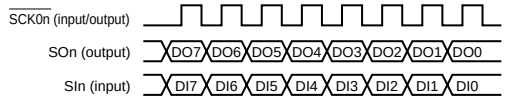
The CSICn register is an 8-bit register that controls the CSI0n transfer operation (n = 0 to 2).

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

Figure 9-23: Clocked Serial Interface Clock Selection Registers (CSICK00 to CSICK02) (1/2)

	7	6	5	4	3	2	1	0	Address	Initial value
CSICK00	0	0	0	CKP	DAP	CKS2	CKS1	CKS0	FFFF F2C2H	00H
CSICK01	0	0	0	CKP	DAP	CKS2	CKS1	CKS0	FFFF F2E2H	00H
CSICK02	0	0	0	CKP	DAP	CKS2	CKS1	CKS0	FFFF F302H	00H

Bit Position	Bit Name	Function
4, 3	CKP, DAP	Specifies operation mode see: CKP: clock phase selection bit DAP: data phase selection bit

CKP	DAP	Operation Mode
0	0	
0	1	
1	0	
1	1	

Remark: n = 0 to 2

Figure 9-23: Clocked Serial Interface Clock Selection Registers (CSICK00 to CSICK02) (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td rowspan="2">2 to 0</td><td rowspan="2">CKS2 to CKS0</td><td>Specifies input clock CSI00Note
<table>
<tr><th>CKS2</th><th>CKS1</th><th>CKS0</th><th>Input Clock</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>f_{XX} /4</td><td>Master mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>f_{XX} /8</td><td>Master mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>f_{XX}/16</td><td>Master mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>f_{XX}/32</td><td>Master mode</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>f_{XX} /64</td><td>Master mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>f_{XX} /256</td><td>Master mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>TM51 output</td><td>Master mode</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>External clock ($\overline{SCK0n}$)</td><td>Slave mode</td></tr>
</table>
Note: TM51 as clock input available</td></tr>
<tr><td>Specifies input clock CSI01, CSI02
<table>
<tr><th>CKS2</th><th>CKS1</th><th>CKS0</th><th>Input Clock</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>f_{XX} /4</td><td>Master mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>f_{XX} /8</td><td>Master mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>f_{XX}/16</td><td>Master mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>f_{XX}/32</td><td>Master mode</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>f_{XX} /64</td><td>Master mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>f_{XX} /128</td><td>Master mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>f_{XX} /256</td><td>Master mode</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>External clock ($\overline{SCK0n}$)</td><td>Slave mode</td></tr>
</table></td></tr>
</table>

Caution: The CSICKn register can be overwritten only when the CSIE bit of the CSIM0n register = 0.

(3) Clocked serial interface reception buffer registers (SIRB00, SIRB01, SIRB02)

The SIRB0n register is a 16-bit buffer register that stores receive data.
When the receive-only mode is set (TRMD bit of CSIM0n register = 0, n = 0,1), the reception operation is started by reading data from the SIRB0n register.

These registers are read-only, in 16-bit units.

In addition to reset input, these registers can also be initialized by clearing (0) the CSIE bit of the CSIM0n register.

Figure 9-24: Clocked Serial Interface Reception Buffer Registers (SIRB00 to SIRB02)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
SIRB00	SIRB015	SIRB014	SIRB013	SIRB012	SIRB011	SIRB010	SIRB9	SIRB8	SIRB7	SIRB6	SIRB5	SIRB4	SIRB3	SIRB2	SIRB01	SIRB00	FFFFF2C4H	0000H
SIRB01	SIRB015	SIRB014	SIRB013	SIRB012	SIRB011	SIRB010	SIRB9	SIRB8	SIRB7	SIRB6	SIRB5	SIRB4	SIRB3	SIRB2	SIRB01	SIRB00	FFFF2E4H	0000H
SIRB02	SIRB015	SIRB014	SIRB013	SIRB012	SIRB011	SIRB010	SIRB9	SIRB8	SIRB7	SIRB6	SIRB5	SIRB4	SIRB3	SIRB2	SIRB01	SIRB00	FFFFF304H	0000H

Bit Position	Bit Name	Function
15 to 0	SIRB015 to SIRB00	Store receive data.

Cautions: 1. Read the SIRB0n register only when the 16-bit data length has been set (CCL bit of CSIM0n register = 1).

2. When the single transfer mode has been set (AUTO bit of CSIM0n register = 0), perform read operation only in the idle state (CSOT bit of CSIM0n register = 0). If the SIRB0n register is read during data transfer, the data cannot be guaranteed.

(4) Clocked serial interface LSB reception buffer registers (SIRBL00, SIRBL01, SIRBL02)

The SIRBL0n register is an 8-bit buffer register that stores receive data (n = 0 to 2).
When the receive-only mode is set (TRMD bit of CSIM0n register = 0), the reception operation is started by reading data from the SIRBL0n register.
In addition to reset input, these registers can also be initialized by clearing (0) the CSIE bit of the CSIM0n register.
The SIRBL0n register is the same as the lower bytes of the SIRB0n register.

These registers are read-only, in 8-bit units.

Figure 9-25: Clocked Serial Interface Reception Buffer Registers (SIRBL00 to SIRBL02)

	7	6	5	4	3	2	1	0	Address	Initial value
SIRBL00	SIRB7	SIRB6	SIRB5	SIRB4	SIRB3	SIRB2	SIRB01	SIRB00	FFFF F2C4H	00H
SIRBL01	SIRB7	SIRB6	SIRB5	SIRB4	SIRB3	SIRB2	SIRB01	SIRB00	FFFF F2E4H	00H
SIRBL02	SIRB7	SIRB6	SIRB5	SIRB4	SIRB3	SIRB2	SIRB01	SIRB00	FFFF F304H	00H

Bit Position	Bit Name	Function
7 to 0	SIRB7 to SIRB00	Stores receive data.

Cautions: 1. Read the SIRBL0n register only when the 8-bit data length has been set (CCL bit of CSIM0n register = 0).

2. When the single transfer mode is set (AUTO bit of CSIM0n register = 0), perform read operation only in the idle state (CSOT bit of CSIM0n register = 0). If the SIRBL0n register is read during data transfer, the data cannot be guaranteed.

(5) Clocked serial interface LSB transmission buffer registers (SOTB00, SOTB01, SOTB02)

The SOTB0n register is a 16-bit buffer register that stores transmit data (n = 0 to 2).
When the transmission/reception mode is set (TRMD bit of CSIM0n register = 1), the transmission operation is started by writing data to the SOTB0n register.

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

Figure 9-26: Clocked Serial Interface Transmission Buffer Registers 0, 1, 2 (SOTB0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
SOTB00	SOTB15	SOTB14	SOTB13	SOTB12	SOTB11	SOTB10	SOTB9	SOTB8	SOTB7	SOTB6	SOTB5	SOTB4	SOTB3	SOTB2	SOTB1	SOTB0	FFFF F2C8H	0000H
SOTB01	SOTB15	SOTB14	SOTB13	SOTB12	SOTB11	SOTB10	SOTB9	SOTB8	SOTB7	SOTB6	SOTB5	SOTB4	SOTB3	SOTB2	SOTB1	SOTB0	FFFF F2E8H	0000H
SOTB02	SOTB15	SOTB14	SOTB13	SOTB12	SOTB11	SOTB10	SOTB9	SOTB8	SOTB7	SOTB6	SOTB5	SOTB4	SOTB3	SOTB2	SOTB1	SOTB0	FFFF F308H	0000H

Bit Position	Bit Name	Function
15 to 0	SOTB15 to SOTB0	Store transmit data.

Cautions: 1. Access the SOTB0n register only when the 16-bit data length is set (CCL bit of CSIM0n register = 1).

2. When the single transfer mode is set (AUTO bit of CSIM0n register = 0), perform access only in the idle state (CSOT bit of CSIM0n register = 0). If the SOTB0n register is accessed during data transfer, the data cannot be guaranteed.

(6) Clocked serial interface LSB transmission buffer registers (SOTBL00, SOTBL01, SOTBL02)

The SOTBL0n register is an 8-bit buffer register that stores transmit data (n = 0 to 2).
When the transmission/reception mode is set (TRMD bit of CSIM0n register = 1), the transmission operation is started by writing data to the SOTBL0n register.

These registers can be read/written in 8-bit units.

The SOTBL0n register is the same as the lower bytes of the SOTB0n register.

Figure 9-27: Clocked Serial Interface Transmission Buffer Registers (SOTBL0n)

	7	6	5	4	3	2	1	0	Address	Initial value
SOTBL00	SOTB7	SOTB6	SOTB5	SOTB4	SOTB3	SOTB2	SOTB1	SOTB0	FFFF F2C8H	00H
SOTBL01	SOTB7	SOTB6	SOTB5	SOTB4	SOTB3	SOTB2	SOTB1	SOTB0	FFFF F2E8H	00H
SOTBL02	SOTB7	SOTB6	SOTB5	SOTB4	SOTB3	SOTB2	SOTB1	SOTB0	FFFF F308H	00H

Bit Position	Bit Name	Function
7 to 0	SOTB7 to SOTB0	Store transmit data.

Cautions: 1. Access the SOTBL0n register only when the 8-bit data length has been set (CCL bit of CSIM0n register = 0).

2. When the single transfer mode is set (AUTO bit of CSIM0n register = 0), perform access only in the idle state (CSOT bit of CSIM0n register = 0). If the SOTBL0n register is accessed during data transfer, the data cannot be guaranteed.

3. When SOTBL0n is written by 8 bit access, then undefined data are written in the higher bits 15 to 8 of SOTB0n.

(7) Clocked serial interface initial transmission buffer registers (SOTBF00, SOTBF01, SOTBF02)

The SOTBF0n register is a 16-bit buffer register that stores initial transmission data in the repeat transfer mode (n = 0 to 2).
The transmission operation is not started even if data is written to the SOTBF0n register.

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

Figure 9-28: Clocked Serial Interface Initial Transmission Buffer Registers 0, 1, 2 (SOTBF0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
SOTBF00	SOTBF15	SOTBF14	SOTBF13	SOTBF12	SOTBF11	SOTBF10	SOTBF9	SOTBF8	SOTBF7	SOTBF6	SOTBF5	SOTBF4	SOTBF3	SOTBF2	SOTBF1	SOTBF0	FFFFF2D0H	0000H
SOTBF01	SOTBF15	SOTBF14	SOTBF13	SOTBF12	SOTBF11	SOTBF10	SOTBF9	SOTBF8	SOTBF7	SOTBF6	SOTBF5	SOTBF4	SOTBF3	SOTBF2	SOTBF1	SOTBF0	FFFFF2F0H	0000H
SOTBF02	SOTBF15	SOTBF14	SOTBF13	SOTBF12	SOTBF11	SOTBF10	SOTBF9	SOTBF8	SOTBF7	SOTBF6	SOTBF5	SOTBF4	SOTBF3	SOTBF2	SOTBF1	SOTBF0	FFFFF310H	0000H

Bit Position	Bit Name	Function
15 to 0	SOTBF15 to SOTBF0	Stores initial transmission data in repeat transfer mode.

Caution: Access the SOTBF0n register only when the 16-bit data length has been set (CCL bit of CSIM0n register = 1), and only in the idle state (CSOT bit of CSIM0n register = 0). If the SOTBF0n register is accessed during data transfer, the data cannot be guaranteed.

(8) Clocked serial interface LSB initial transmission buffer registers (SOTBFL00, SOTBFL01, SOTBFL02)

The SOTBFL0n register is an 8-bit buffer register that stores initial transmission data in the repeat transfer mode (n = 0 to 2).
The transmission operation is not started even if data is written to the SOTBFL0n register.
The SOTBFL0n register is the same as the lower bytes of the SOTBF0n register.

These registers can be read/written in 8-bit units.

Figure 9-29: Clocked Serial Interface Initial Transmission Buffer Registers (SOTBFL0n)

	7	6	5	4	3	2	1	0	Address	Initial value
SOTBFL00	SOTBF7	SOTBF6	SOTBF5	SOTBF4	SOTBF3	SOTBF2	SOTBF1	SOTBF0	FFFF F2D0H	00H
SOTBFL01	SOTBF7	SOTBF6	SOTBF5	SOTBF4	SOTBF3	SOTBF2	SOTBF1	SOTBF0	FFFF F2F0H	00H
SOTBFL02	SOTBF7	SOTBF6	SOTBF5	SOTBF4	SOTBF3	SOTBF2	SOTBF1	SOTBF0	FFFF F310H	00H

Bit Position	Bit Name	Function
7 to 0	SOTBF7 to SOTBF0	Store initial transmission data in repeat transfer mode.

Cautions: 1. Access the SOTBFL0n register only when the 8-bit data length has been set (CCL bit of CSIM00 register = 0), and only in the idle state (CSOT bit of CSIM0n register = 0). If the SOTBFL0n register is accessed during data transfer, the data cannot be guaranteed.

2. When SOTBFL0n is written by 8 bit access, then undefined data are written in the higher bits 15 to 8 of SOTBF0n.

(9) Serial I/O shift registers (SIO00, SIO01, SIO02)

The SIO0n register is a 16-bit shift register that converts parallel data into serial data (n = 0 to 2).
The transfer operation is not started even if the SIO0n register is read.
These registers are read-only, in 16-bit units.
In addition to reset input, this register can also be initialized by clearing (0) the CSIE bit of the CSIM0n register.

Figure 9-30: Serial I/O Shift Registers 0, 1, 2 (SIO0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
SIO00	SIO15	SIO14	SIO13	SIO12	SIO11	SIO10	SIO9	SIO8	SIO7	SIO6	SIO5	SIO4	SIO3	SIO2	SIO1	SIO0	FFFFF2D4H	0000H
SIO01	SIO15	SIO14	SIO13	SIO12	SIO11	SIO10	SIO9	SIO8	SIO7	SIO6	SIO5	SIO4	SIO3	SIO2	SIO1	SIO0	FFFFF2F4H	0000H
SIO02	SIO15	SIO14	SIO13	SIO12	SIO11	SIO10	SIO9	SIO8	SIO7	SIO6	SIO5	SIO4	SIO3	SIO2	SIO1	SIO0	FFFFF314H	0000H

Bit Position	Bit Name	Function
15-0	SIO15 to SIO0	Data is shifted in (reception) or shifted out (transmission) from the MSB or LSB side.

Caution: Access the SIO0n register only when the 16-bit data length has been set (CCL bit of CSIM0n register = 1), and only in the idle state (CSOT bit of CSIM0n register = 0). If the SIO0n register is accessed during data transfer, the data cannot be guaranteed.

(10) Serial I/O LSB shift registers (SIOL00, SIOL01, SIOL02)

The SIOLn register is an 8-bit shift register that converts parallel data into serial data (n = 0 to 2).
The transfer operation is not started even if the SIOLn register is read.
These registers are read-only, in 8-bit units.
In addition to reset input, this register can also be initialized by clearing (0) the CSIE bit of the CSIM0n register.
The SIOLn register is the same as the lower bytes of the SIO0n register.

Figure 9-31: Serial I/O Shift Registers L00, L01, L02 (SIOL0n)

	7	6	5	4	3	2	1	0	Address	Initial value
SIOL00	SIO7	SIO6	SIO5	SIO4	SIO3	SIO2	SIO1	SIO0	FFFF F2D4H	00H
SIOL01	SIO7	SIO6	SIO5	SIO4	SIO3	SIO2	SIO1	SIO0	FFFF F2F4H	00H
SIOL02	SIO7	SIO6	SIO5	SIO4	SIO3	SIO2	SIO1	SIO0	FFFF F314H	00H

Bit Position	Bit Name	Function
7 to 0	SIO7 to SIO0	Data is shifted in (reception) or shifted out (transmission) from the MSB or LSB side.

Caution: Access the SIOLn register only when the 8-bit data length has been set (CCL bit of CSIM0n register = 0), and only in the idle state (CSOT bit of CSIM0n register = 0). If the SIOLn register is accessed during data transfer, the data cannot be guaranteed.

9.3.4 Operation

(1) Single transfer mode

(a) Usage

In the receive-only mode (TRMD bit of CSIM0n register = 0), transfer is started by reading[Note 1] the receive data buffer register (SIRB0n/SIRBL0n) (n = 0 to 2).
In the transmission/reception mode (TRMD bit of CSIM0n register = 1), transfer is started by writing[Note 2] to the transmit data buffer register (SOTB0n/SOTBL0n).
In the slave mode, the operation must be enabled beforehand (CSIE bit of CSIM0n register = 1).
When transfer is started, the value of the CSOT bit of the CSIM0n register becomes “1” (transmission execution status).
Upon transfer completion, the transmission/reception completion interrupt (INTCSI0n) is set (1), and the CSOT bit is cleared (0). The next data transfer request is then waited for.

Notes: 1. When the 16-bit data length (CCL bit of CSIM0n register = 1) has been set, read the SIRB0n register. When the 8-bit data length (CCL bit of CSIM0n register = 0) has been set, read the SIRBL0n register.

2. When the 16-bit data length (CCL bit of CSIM0n register = 1) has been set, write to the SOTB0n register. When the 8-bit data length (CCL bit of CSIM0n register = 0) has been set, write to the SOTBL0n register.

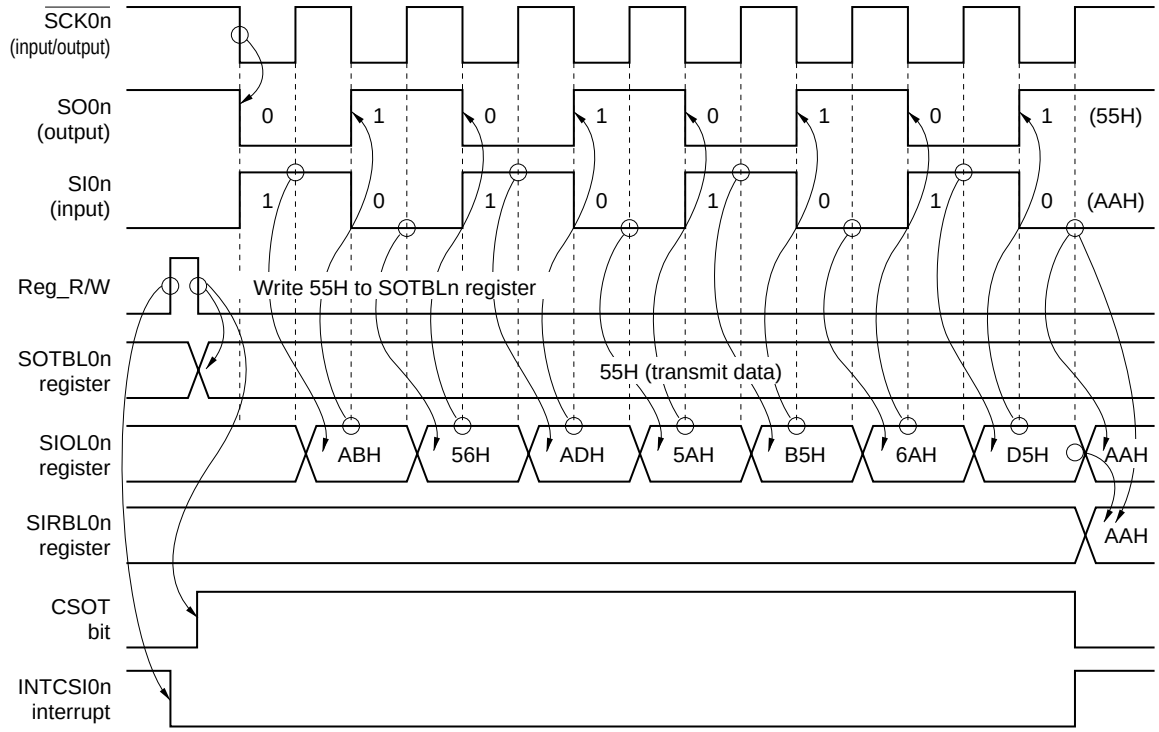
Caution: When the CSOT bit of the CSIM0n register = 1, do not manipulate the CSIn register.

As an example of the bidirectional communication the following example shows the sending of 55H and the receiving of AAH via the CSI. The following two timing charts shows the communication with different settings of the Data Phase Selection-bit (DAP)

For detailed information of the DAP and CKP bits please refer to (b)“Clock phase selection” on page 304.

Figure 9-32: Timing Chart in Single Transfer Mode (DAP = 0) (1/2)

(a) In transmission/reception mode, data length: 8 bits, transfer direction: MSB first, no interrupt delay, single transfer mode, operation mode: CKP bit = 0, DAP bit = 0

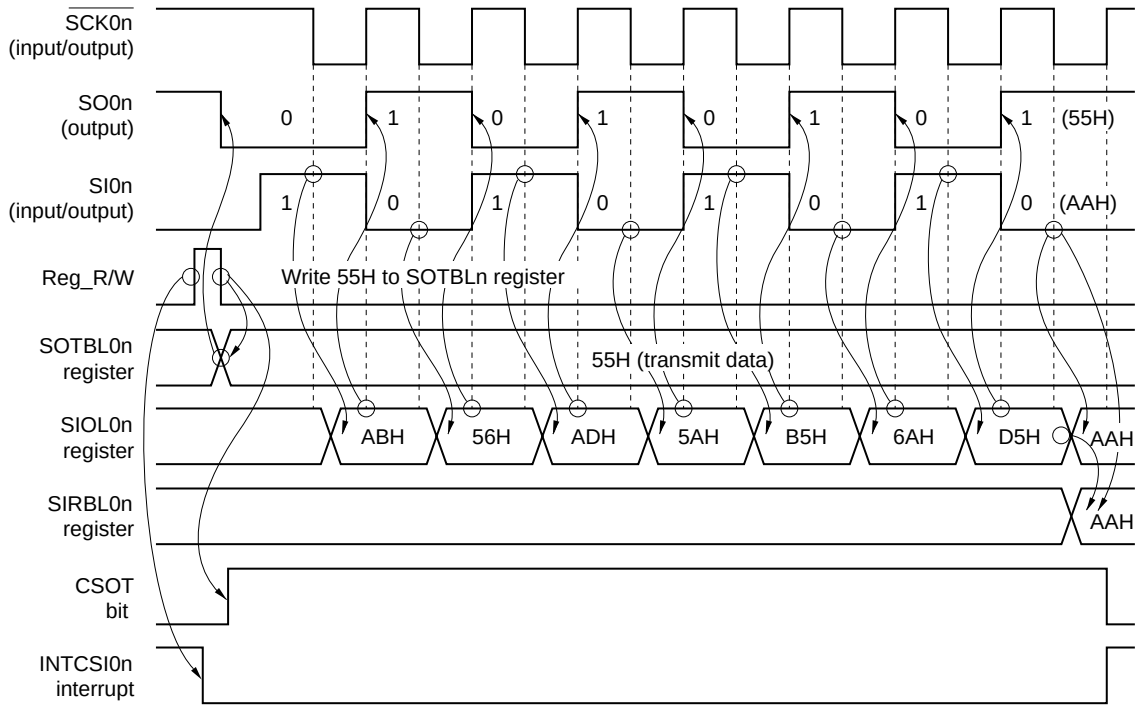


Remarks: 1. n = 0 to 2

2. Reg_R/W:Internal signal. This signal indicates that receive data buffer register (SIRB0n/SIRBL0n) read or transmit data buffer register (SOTB0n/SOTBL0n) write was performed.

Figure 9-32: Timing Chart in Single Transfer Mode (DAP = 1) (2/2)

(b) In transmission/reception mode, data length: 8 bits, transfer direction: MSB first, no interrupt delay, single transfer mode, operation mode: CKP bit = 0, DAP bit = 1



Remarks: 1. n = 0 to 2

2. Reg_R/W:Internal signal. This signal indicates that receive data buffer register (SIRB0n/SIRBL0n) read or transmit data buffer register (SOTB0n/SOTBL0n) write was performed.

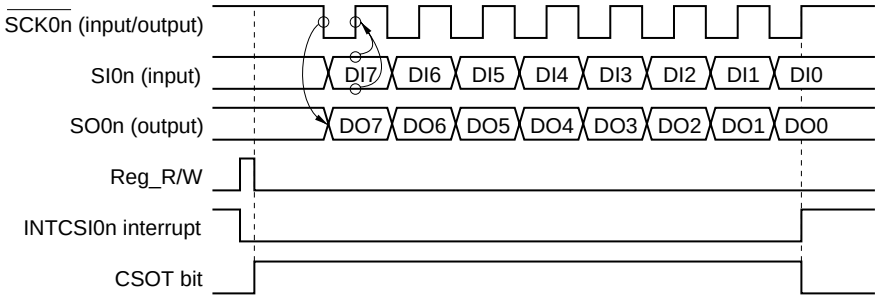
(b) Clock phase selection

The following shows the timing when changing the conditions for clock phase selection (CKP bit of CSICn register) and data phase selection (DAP bit of CSICn register) under the following conditions.

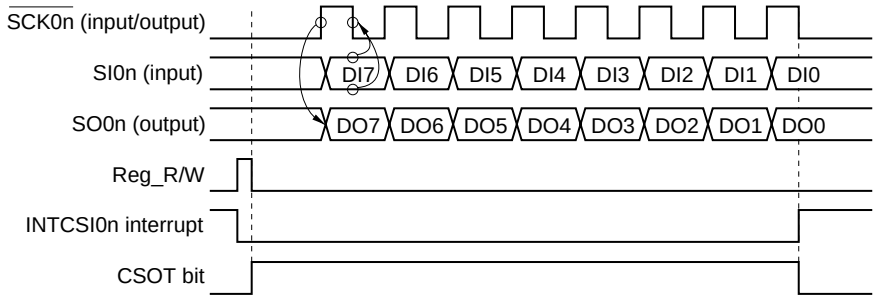
- Data length = 8 bits (CCL bit of CSIM0n register = 0)
- First bit of transfer data = MSB (DIR bit of CSIM0n register = 0)
- No interrupt request signal delay control (CSIT bit of CSIM0n register = 0)

Figure 9-33: Timing Chart According to Clock Phase Selection (1/2)

(a) When CKP bit = 0, DAP bit = 0



(b) When CKP bit = 1, DAP bit = 0

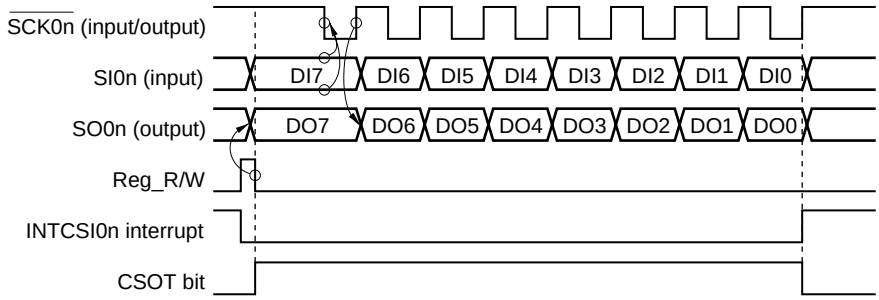


Remarks: 1. n = 0 to 2

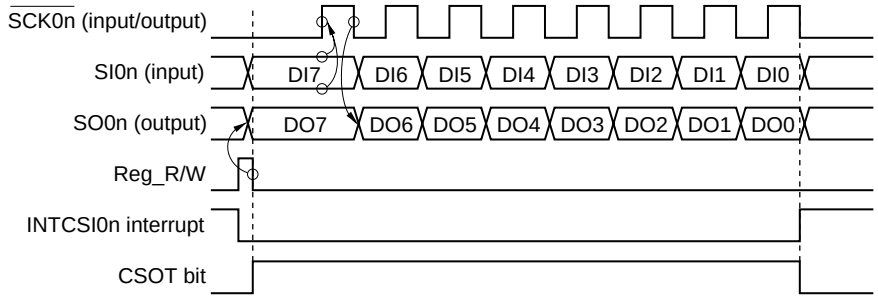
2. Reg_R/W:Internal signal. This signal indicates that receive data buffer register (SIRB0n/SIRBL0n) read or transmit data buffer register (SOTB0n/SOTBL0n) write was performed.

Figure 9-33: Timing Chart According to Clock Phase Selection (2/2)

(c) When CKP bit = 0, DAP bit = 1



(d) When CKP bit = 1, DAP bit = 1



Remarks: 1. n = 0 to 2

2. Reg_R/W:Internal signal. This signal indicates that receive data buffer register (SIRB0n/SIRBL0n) read or transmit data buffer register (SOTB0n/SOTBL0n) write was performed. Transmission/reception completion interrupt request signals (INTCSI00, INTCSI01, INTCSI02)

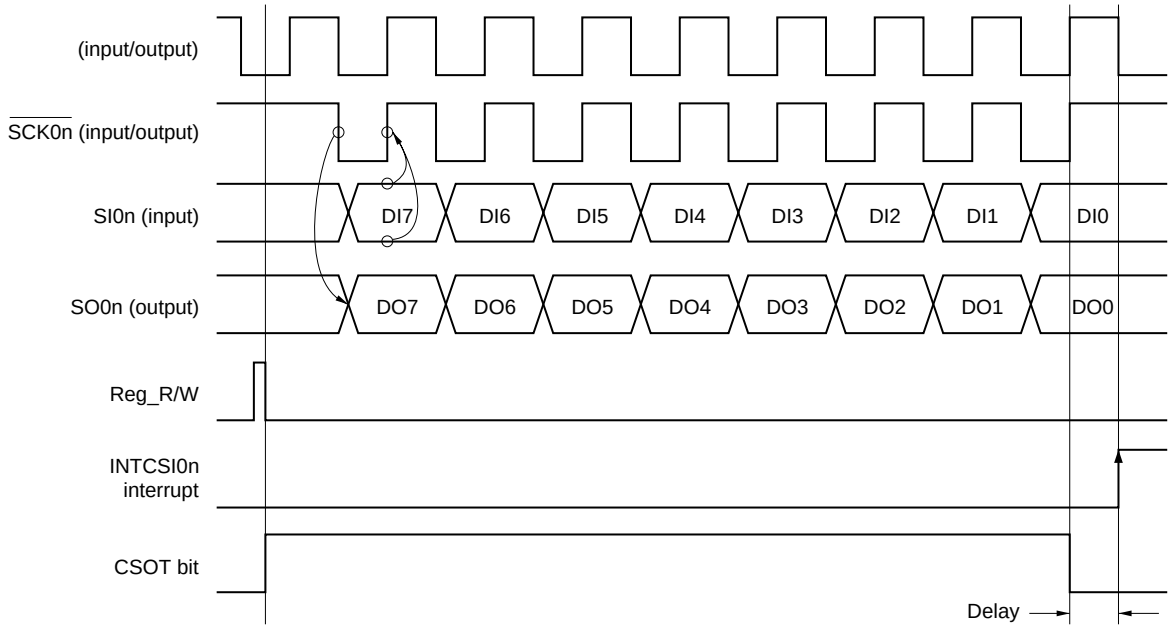
(c) Transmission/reception completion interrupt request signals (INTCSI00, INTCSI01, INTCSI02)

INTCSI0n is set (1) upon completion of data transmission/reception.

Caution: The delay mode (CSIT bit = 1) is valid only in the master mode (bits CKS2 to CKS0 of the CSICn register are not 111B). The delay mode cannot be set when the slave mode is set (bits CKS2 to CKS0 = 111B).

Figure 9-34: Timing Chart of Interrupt Request Signal Output in Delay Mode (1/2)

(a) When CKP bit = 0, DAP bit = 0

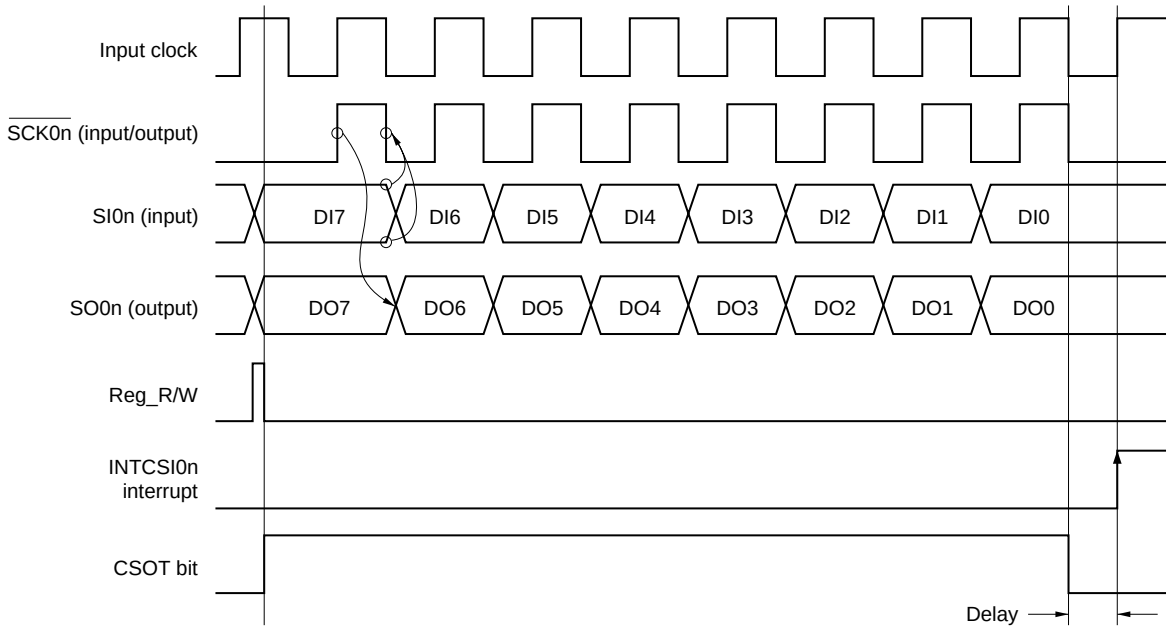


Remarks: 1. n = 0 to 2

2. Reg_R/W:Internal signal. This signal indicates that receive data buffer register (SIRB0n/SIRBL0n) read or transmit data buffer register (SOTB0n/SOTBL0n) write was performed.

Figure 9-34: Timing Chart of Interrupt Request Signal Output in Delay Mode (2/2)

(b) When CKP bit = 1, DAP bit = 1



Remarks: 1. n = 0 to 2

2. Reg_R/W:Internal signal. This signal indicates that receive data buffer register (SIRB0n/SIRBL0n) read or transmit data buffer register (SOTB0n/SOTBL0n) write was performed.

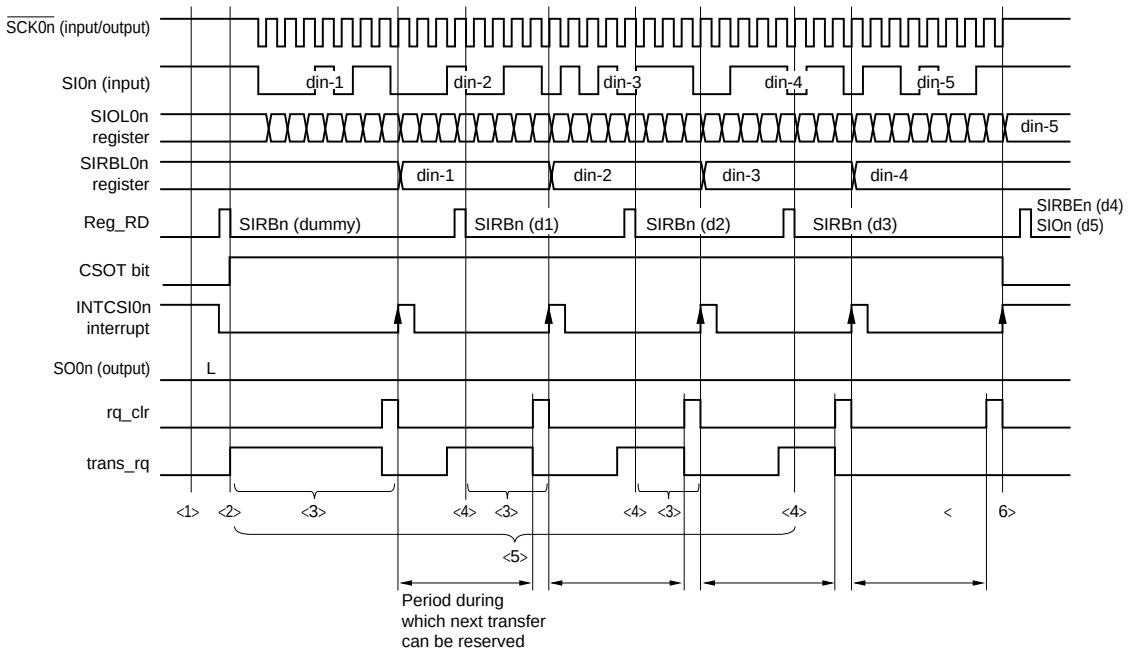
(2) Repeat transfer mode

(a) Usage (receive-only)

<1> Set the repeat transfer mode (AUTO bit of CSIM0n register = 1) and the receive-only mode (TRMD bit of CSIM0n register = 0).
<2> Read SIRB0n register (start transfer with dummy read).
<3> Wait for transmission/reception completion interrupt request (INTCSI0n).
<4> When the transmission/reception completion interrupt request (INTCSI0n) has been set to (1), read the SIRB0n register**Note** (reserve next transfer).
<5> Repeat steps <3> and <4> (n - 2) times (n: number of transfer data).
<6> Following output of the last transmission/reception completion interrupt request (INTCSI0n), read the SIRB0n register and the SIO0n register**Note**.

Note: When transferring n number of data, receive data is loaded by reading the SIRB0n register from the first data to the (n - 2)-th data. The n-th (last) data is loaded by reading the SIO0n register.

Figure 9-35: Repeat Transfer (Receive-Only) Timing Chart



Remarks: 1. n = 0 to 2

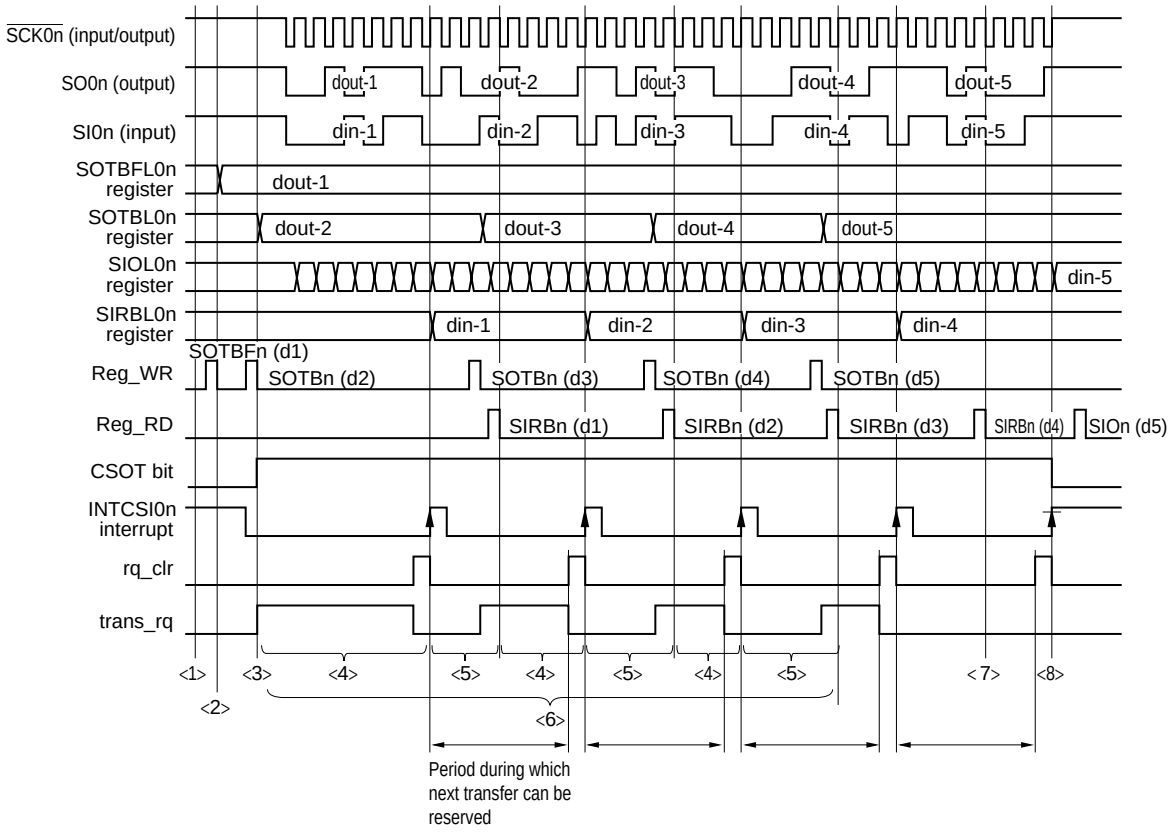
2. Reg_RD : Internal signal. This signal indicates that the receive data buffer register (SIRB0n/SIRBL0n) has been read.
rq_clr : Internal signal. Transfer request clear signal.
trans_rq : Internal signal. Transfer request signal.

In the case of the repeat transfer mode, two transfer requests are set at the start of the first transfer. Following the transmission/reception completion interrupt request (INTCSI0n), transfer is continued if the SIRB0n register can be read within the next transfer reservation period. If the SIRB0n register cannot be read, transfer ends and the SIRB0n register does not receive the new value of the SIO0n register.
The last data can be obtained by reading the SIO0n register following completion of the transfer.

(b) Usage (transmission/reception)

<1> Set the repeat transfer mode (AUTO bit of CSIM0n register = 1) and the transmission/reception mode (TRMD bit of CSIM0n register = 1).
<2> Write the first data to the SOTBF0n register.
<3> Write the 2nd data to the SOTB0n register (start transfer).
<4> Wait for transmission/reception completion interrupt request (INTCSI0n).
<5> When the transmission/reception completion interrupt request (INTCSI0n) has been set to (1), write the next data to the SOTB0n register (reserve next transfer), and read the SIRB0n register to load the receive data.
<6> Repeat steps <4> and <5> as long as data to be sent remains.
<7> Wait for the INTCSI0n interrupt. When the interrupt request signal is set to (1), read the SIRB0n register to load the (n - 1)-th receive data.
<8> Following the last transmission/reception completion interrupt request (INTCSI0n), read the SIO0n register to load the n-th (last) receive data.

Figure 9-36: Repeat Transfer (Transmission/Reception) Timing Chart



Remarks: 1. n = 0 to 2

2. Reg_WR : Internal signal. This signal indicates that the transmit data buffer register (SOTB0n/SOTBL0n) has been written.
Reg_RD : Internal signal. This signal indicates that the receive data buffer register (SIRB0n/SIRBL0n) has been read.
rq_clr : Internal signal. Transfer request clear signal.
trans_rq : Internal signal. Transfer request signal.

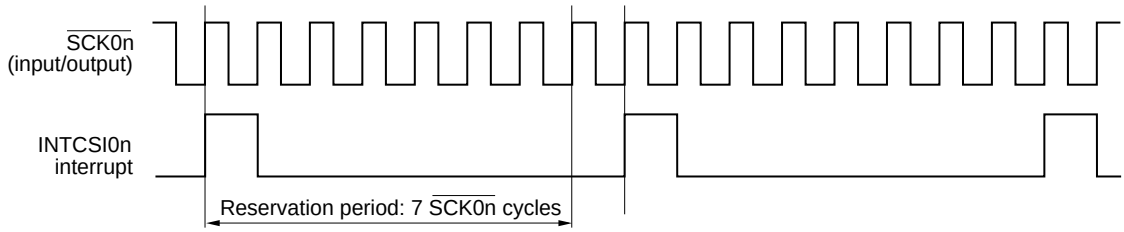
In the case of the repeat transfer mode, two transfer requests are set at the start of the first transfer. Following the transmission/reception completion interrupt request (INTCSI0n), transfer is continued if the SOTB0n register can be written within the next transfer reservation period. If the SOTB0n register cannot be written, transfer ends and the SIRB0n register does not receive the new value of the SIO0n register. The last receive data can be obtained by reading the SIO0n register following completion of the transfer.

(c) Next transfer reservation period

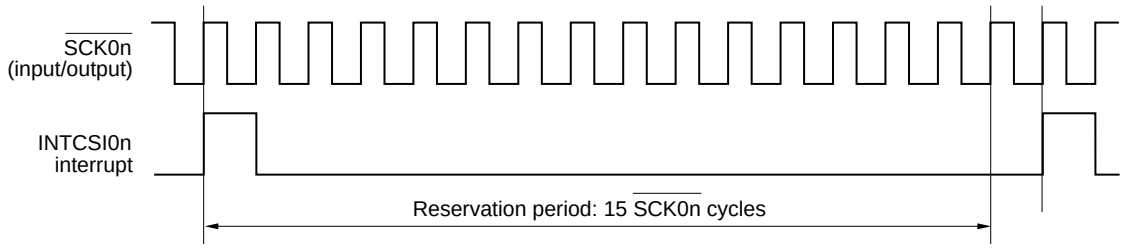
In the repeat transfer mode, the next transfer must be prepared with the period shown in Figure 9-37.

Figure 9-37: Timing Chart of Next Transfer Reservation Period

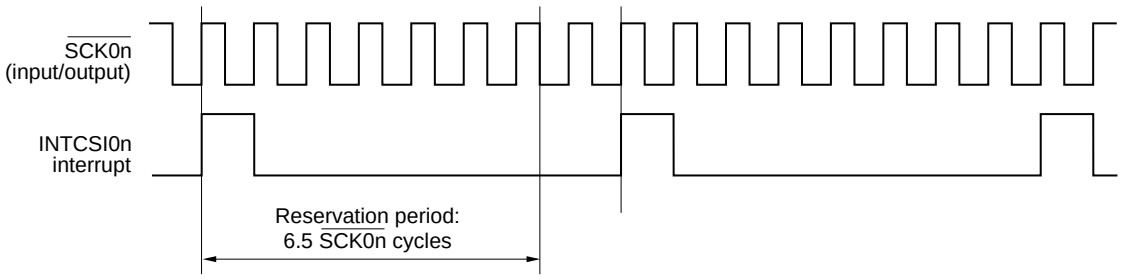
(a) When data length: 8 bits, operation mode: CKP bit = 0, DAP bit = 0



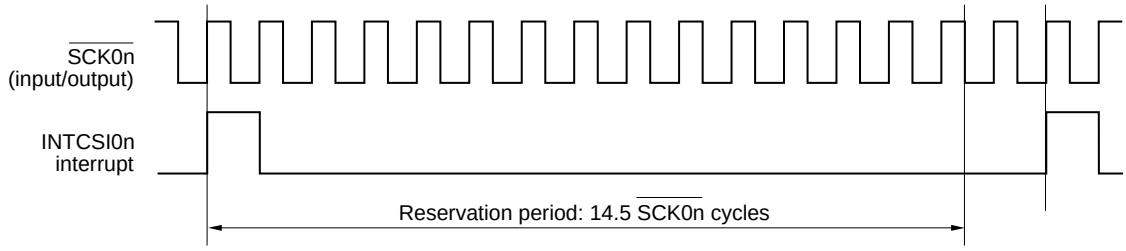
(b) When data length: 16 bits, operation mode: CKP bit = 0, DAP bit = 0



(c) When data length: 8 bits, operation mode: CKP bit = 0, DAP bit = 1



(d) When data length: 16 bits, operation mode: CKP bit = 0, DAP bit = 1



Remark: n = 0 to 2

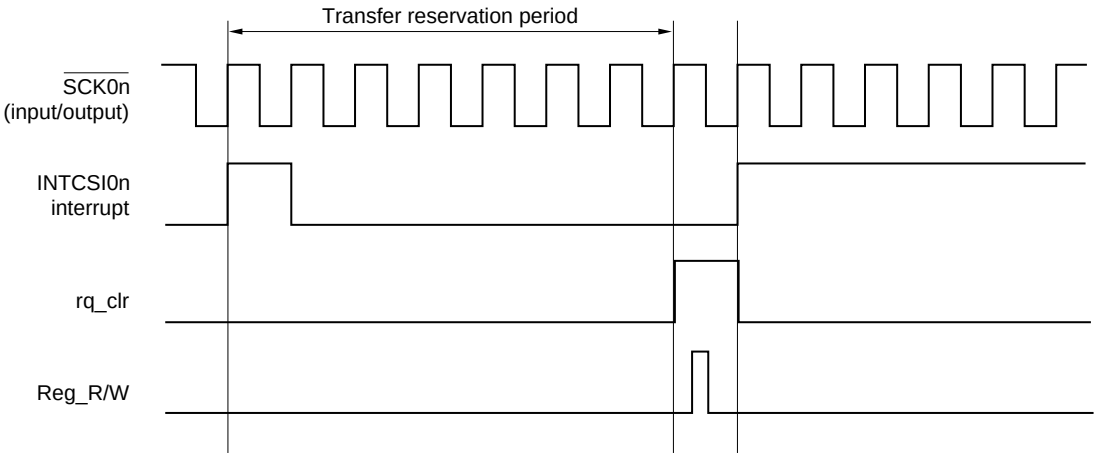
(d) Cautions

To continue repeat transfers, it is necessary to either read the SIRB0n register or write to the SOTB0n register during the transfer reservation period.
If access is performed to the SIRB0n register or the SOTB0n register when the transfer reservation period is over, the following occurs.

- In the case of contention between transfer request clear and register access

Since request cancellation has higher priority, the next transfer request is ignored. Therefore, transfer is interrupted, and normal data transfer cannot be performed.

Figure 9-38: Transfer Request Clear and Register Access Contention



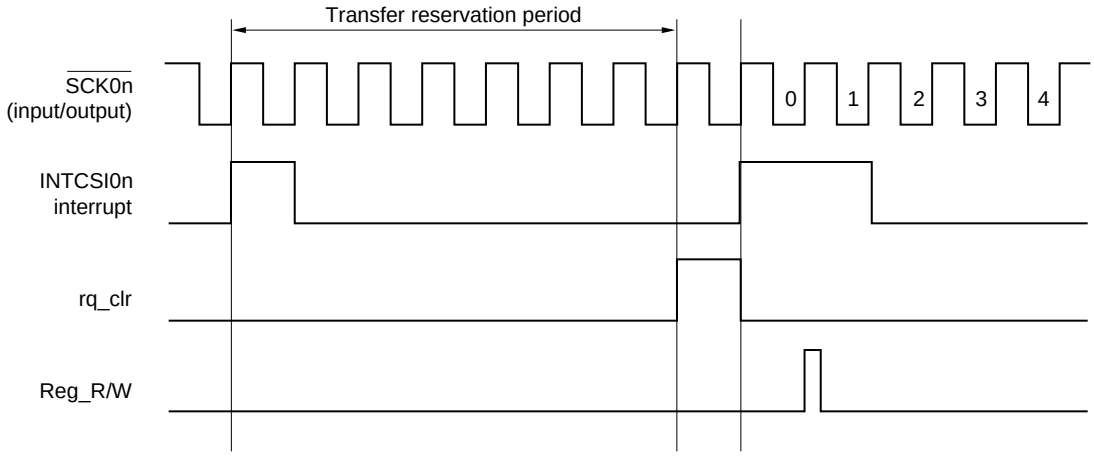
Remarks: 1. n = 0 to 2

2. rq_clr : Internal signal. Transfer request clear signal.
Reg_WR : Internal signal. This signal indicates that the transmit data buffer register (SOTB0n/SOTBL0n) has been written.

- In the case of contention between interrupt request and register access

Since continuous transfer has stopped once, executed as a new repeat transfer.
In the slave mode, a bit phase error transfer error results (refer to Figure 9-39).
In the transmission/reception mode, the value of the SOTBF0n register is retransmitted, and illegal data is sent.

Figure 9-39: Interrupt Request and Register Access Contention



Remarks: 1. n = 0 to 2

2. rq_clr : Internal signal. Transfer request clear signal.
Reg_WR : Internal signal. This signal indicates that the transmit data buffer register (SOTB0n/SOTBL0n) has been written.

9.3.5 Output pins

(1) $\overline{\text{SCK0n}}$ pin

When the CSIn operation is disabled (CSIE bit of CSIM0n register = 0), the $\overline{\text{SCK0n}}$ pin output status is as follows (n = 0 to 2).

CKP	CKS2	CKS1	CKS0	$\overline{\text{SCK0n}}$ Pin Output
0	Don't care	Don't care	Don't care	Fixed to high level
1	1	1	1	Fixed to high level
	Other than above			Fixed to low level

Remarks: 1. n = 0 to 2

2. When any of bits CKP and CKS2 to CKS0 of the CSICn register is overwritten, the $\overline{\text{SCK0n}}$ pin output changes.

(2) SO0n pin

When the CSIn operation is disabled (CSIE bit of CSIM0n register = 0), the SO0n pin output status is as follows (n = 0 to 2).

TRMD	DAP	AUTO	CCL	DIR	SO0n Pin Output
0	X	X	X	X	Fixed at low level
1	0	X	X	X	SO latch value (low level)
	1	0	0	0	SOTB7 value
				1	SOTB0 value
			1	0	SOTB15 value
				1	SOTB0 value
		1	0	0	SOTBF7 value
				1	SOTBF0 value
			1	0	SOTBF15 value
				1	SOTBF0 value

Remarks: 1. When any of bits TRMD, CCL, DIR, AUTO, and CSICn of the CSIM0n register or DAP bit of the CSICn register is overwritten, the SO0n pin output changes.

2. SOTBm: Bit m of SOTB0n register (m = 0, 7, 15)

3. SOTBFm: Bit m of SOTBF0n register (m = 0, 7, 15)

4. n = 0 to 2

5. X = Don't care

9.3.6 TM51 output as dedicated baud rate generator for CSI00

(1) Selecting TM51 as the baud rate generator

The CSI00 serial clock can be selected between TM51 as an baud rate generator output or internal system clock (f_{XX}) with prescaler.

The serial clock source is specified by bits CKS2 to CKS0 of registers CSICK00 and CSIC10 (refer to chapter 9.3.2 (2)“Clocked serial interface clock selection registers (CSICK01, CSICK01, CSICK02)” on page 287).

If the TM51 output is specified, TM51 is selected as the clock source.

Since the same serial clock can be shared for transmission and reception, baud rate is the same for the transmission/reception.

(2) Configuration

For configuration refer to Chapter **6.3 “Timer 5 (TM50, TM51)” on page 220**.

Caution: If TM51 is used as the baud rate generator for CSI00 don’t exceed 5 MHz as the output clock of TM51.

[MEMO]

Chapter 10 DCAN

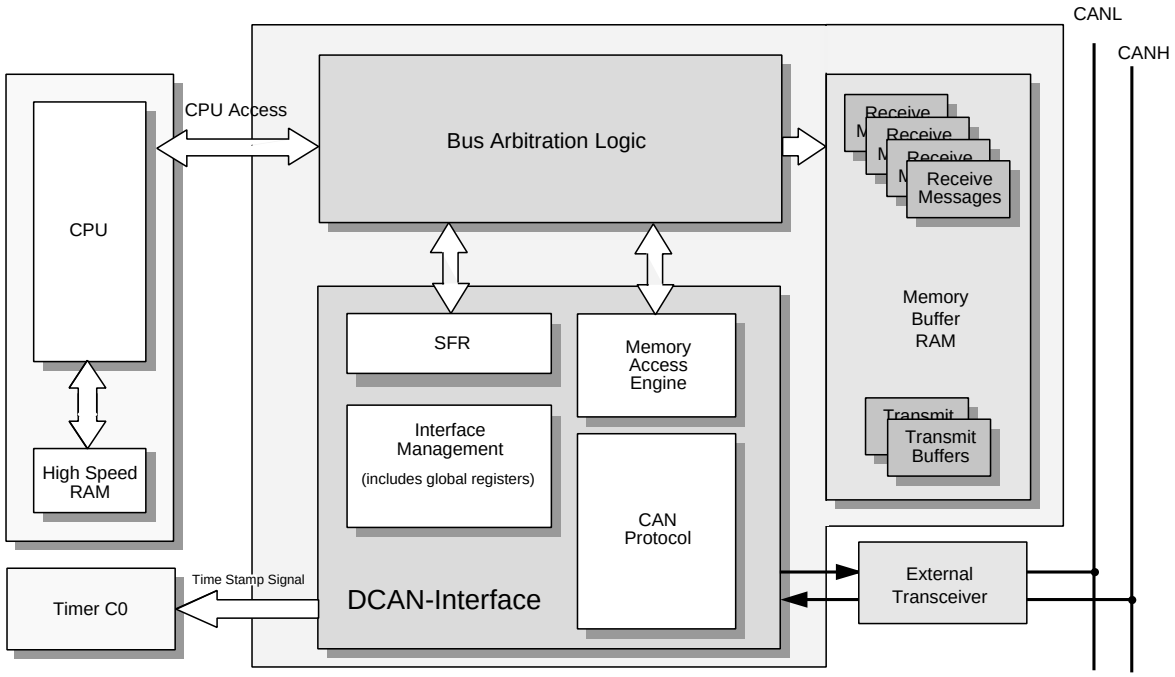
10.1 Outline Description

The V850/DB1 supports 2 DCAN interfaces, which got the same functionality. In this chapter both interfaces are described at the same time with index n (n = 0, 1). Where necessary the register of both DCAN interfaces are showed.

Remark: In this DCAN chapter following indexes were consequently used

- n = 0, 1 (for each of the 2 DCAN channels: DCAN0, DCAN1)
- m = 2, 4 (address offset index for the 2 Mask Buffers)
- r = 02 to 11 (address offset index for the 14 Receive Buffers)
- t = 00, 01 (address offset index for the 2 Transmit Buffers)

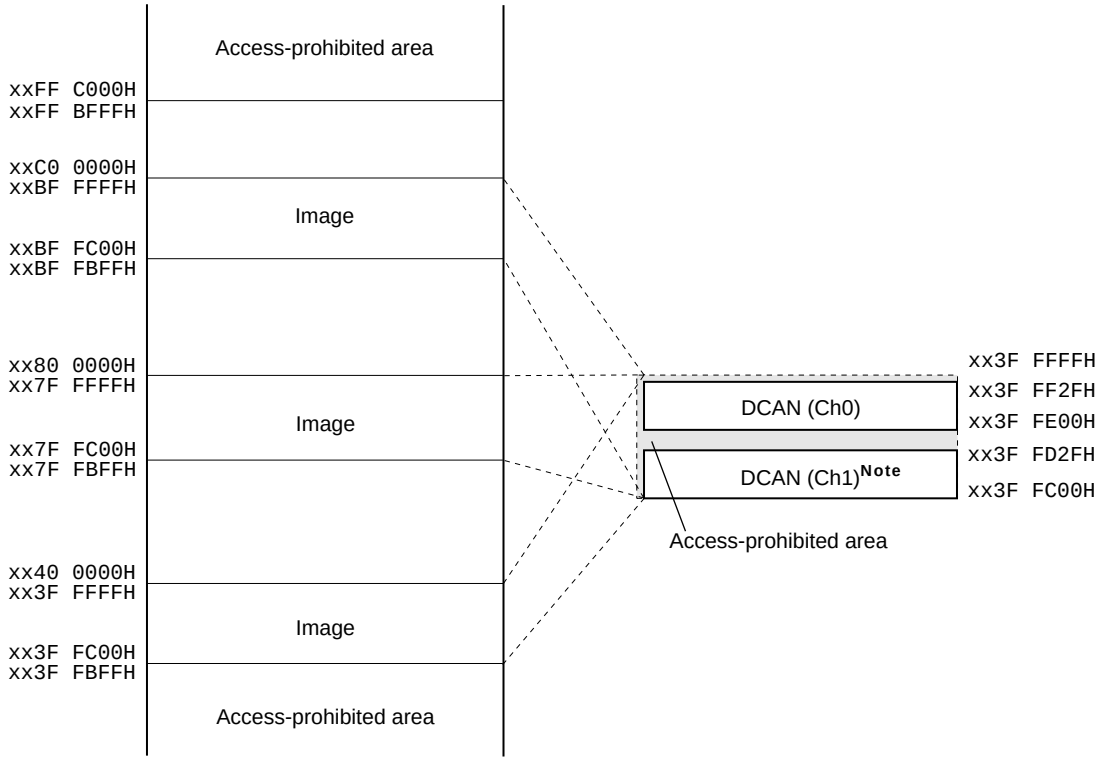
Figure 10-1: Structural Block Diagram



This interface part handles all protocol activities by hardware in the CAN protocol part. The memory access engine fetches information for the CAN protocol transmission from the dedicated RAM area to the CAN protocol part or compares and sorts incoming information and stores it into predefined RAM areas.
The DCAN interfaces directly to the RAM area that is accessible by the DCAN and by the CPU.
The DCAN part works with an external bus transceiver which converts the transmit data and receive data lines to the electrical characteristics of the CAN bus itself.

10.2 CAN memory

Figure 10-2: DCAN Memory



Note: µPD10F3080 (Flash Product only)

Each DCAN channel uses a fix address area in the RAM.

- For DCAN0 the start address is XXnF FE00
- For DCAN1 the start address is XXnF FC00

(X = don't care) (n = 3, 7, B)

This register defines the lower starting address for the DCAN area.

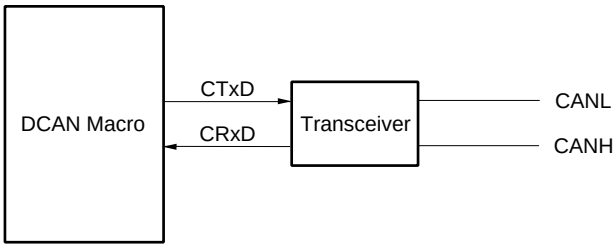
The memory layout sections that implemented the DCAN area, contain areas that are defined as Access-prohibited area.

Remark: Areas defined as Access-prohibited area contain neither DCAN SFR registers nor DCAN buffers. Those area must not be read nor written by user program.

10.3 Connection with Target System

The DCAN Macro has to be connected to the CAN bus with an external transceiver.

Figure 10-3: Connection to the CAN Bus



10.4 DCAN Controller Configuration

The DCAN-module consists of the following hardware.

Item	Configuration
Message definition	In RAM area
DCAN input/output	1 (CTXD1n) 1 (CRXD1n)
Control register	DCAN control register (DCANCn) CAN control register (CANCn) Transmit control register (TCRn) Received message register (RMESn) Redefinition control register (REDEFn) DCAN error status register (CANESn) Transmit error counter (TECn) Receive error counter (RECn) Message count register (MCNTn) Bit rate prescaler (BRPRSn) Synchronous control register 0 (SNYC0n) Synchronous control register 1 (SYNC1n) Mask control register (MASKCn)

Remark: n = 0, 1

10.5 Special Function Register for DCAN-module

Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
			1-bit	8-bit	16-bit	
DCAN control register	DCANC0	R/W	×	×	-	00H
CAN control register	CANC0	R/W	×	×	-	01H
Transmit control register	TCR0	R/W	-	×	-	00H
Received message register	RMES0	R	-	×	-	00H
Redefinition control register	REDEF0	R/W	×	×	-	00H
DCAN error status register	CANES0	R/W	-	×	-	00H
Transmit error counter	TEC0	R	-	×	-	00H
Receive error counter	REC0	R	-	×	-	00H
Message count register	MCNT0	R	-	×	-	C0H
Bit rate prescaler	BRPRS0	R/W	-	×	-	00H
Synchronous control register 0	SYNC00	R/W	-	×	-	18H
Synchronous control register 1	SYNC10	R/W	-	×	-	0EH
Mask control register	MASKC0	R/W	-	×	-	00H

Register Name	Symbol	R/W	Bit Manipulation Units			After Reset
			1-bit	8-bit	16-bit	
DCAN control register	DCANC1	R/W	×	×	-	00H
CAN control register	CANC1	R/W	×	×	-	01H
Transmit control register	TCR1	R/W	-	×	-	00H
Received message register	RMES1	R	-	×	-	00H
Redefinition control register	REDEF1	R/W	×	×	-	00H
DCAN error status register	CANES1	R/W	-	×	-	00H
Transmit error counter	TEC1	R	-	×	-	00H
Receive error counter	REC1	R	-	×	-	00H
Message count register	MCNT1	R	-	×	-	C0H
Bit rate prescaler	BRPRS1	R/W	-	×	-	00H
Synchronous control register 0	SYNC01	R/W	-	×	-	18H
Synchronous control register 1	SYNC11	R/W	-	×	-	0EH
Mask control register	MASKC1	R/W	-	×	-	00H

The following SFR bits can be accessed with 1-bit instructions. The other SFR registers have to be accessed with 8-bit instructions.

Name	Description	Bit
DCANEN	Enable/Disable DCANn	DCANCn.0
SOFE	Start of frame enable	CANCn.4
SLEEP	Sleep mode	CANCn.2
INIT	Initialize	CANCn.0
DEF	Redefinition enable	REDEFn.7

10.6 Message Buffer Configuration

Address Offset Note 2	Register Name	R/W	After Reset
000H to 00FH	Transmit buffer 0	R/W	undefined Note 1
010H to 01FH	Transmit buffer 1		
020H to 02FH	Receive buffer 0 / Mask 0		
030H to 03FH	Receive buffer 1		
040H to 04FH	Receive buffer 2 / Mask 1		
050H to 05FH	Receive buffer 3		
060H to 06FH	Receive buffer 4		
070H to 07FH	Receive buffer 5		
080H to 08FH	Receive buffer 6		
090H to 09FH	Receive buffer 7		
0A0H to 0AFH	Receive buffer 8		
0B0H to 0BFH	Receive buffer 9		
0C0H to 0CFH	Receive buffer 10		
0D0H to 0DFH	Receive buffer 11		
0E0H to 0EFH	Receive buffer 12		
0F0H to 0FFH	Receive buffer 13		
100H to 10FH	Receive buffer 14		
110H to 11FH	Receive buffer 15		

Notes: 1. Contents is undefined, because data resides in normal RAM area.

2. This address is an offset to the RAM area starting address which is fixed for

- DCAN0 to XXnF FE00 (X = don’t care) (n = 3, 7, B)
- DCAN1 to XXnF FC00 (X = don’t care) (n = 3, 7, B)

10.7 Transmit Buffer Structure

Each DCAN channel has 2 independent transmit buffers. The two buffers have a 16 byte data structure for standard and extended frames with the ability to send up to 8 data bytes per message. The structure of the transmit buffer is similar to the structure of the receive buffers. The CPU can use addresses, that are specified as “unused” in the transmit buffer layout. As well, the CPU may use unused ID addresses, unused data addresses**Note**, and an unused transmit buffer of the DCAN for its own purposes. The control bits, the identification and the message data has to be stored in the message RAM area.

The transmission control is done by the TCRn register. A transmission priority selection allows the customer to realize an application specific priority selection. After the priority selection the transmission can be started by setting the TXRQx bit (x = 0, 1).

In the case that both transmit buffers are used, the transmit priorities can be set. For this purpose the DCAN has the TXP bit in the TCRn register (n = 0, 1). The application software has to set this priority before the transmission is started.

The two transmit buffers of each DCAN channel (DCAN0, DCAN1) supply two independent interrupt lines for an interrupt controller.

Note: Message objects that need less than 8 data byte (DLC < 8) may use the remaining bytes (8 - DLC) for application purposes.

10.8 Transmit Message Buffer Format

Name	Address[Note]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCON	t0H	IDE	RTR	0	0	DLC3	DLC2	DLC1	DLC0
	t1H	Unused							
IDTX0	t2H	ID standard part							
IDTX1	t3H	ID standard part			0	0	0	0	0
IDTX2	t4H	ID extended part							
IDTX3	t5H	ID extended part							
IDTX4	t6H	ID extended part		0	0	0	0	0	0
	t7H	Unused							
DATA0	t8H	Message data byte 0							
DATA1	t9H	Message data byte 1							
DATA2	tAH	Message data byte 2							
DATA3	tBH	Message data byte 3							
DATA4	tCH	Message data byte 4							
DATA5	tDH	Message data byte 5							
DATA6	tEH	Message data byte 6							
DATA7	tFH	Message data byte 7							

Remark: t = 00, 01 (address index for the 2 Transmit Buffers)

Note: This address is a relative offset to the starting address of the transfer buffer (see 10.6 “Message Buffer Configuration” on page 322).

(1) Transmit Message Definition

The memory location labelled TCON includes the information of the RTR bit and the bits of the control field of a data or remote frame.

TCON is set with an 1-bit or an 8-bit memory manipulation instruction.

Figure 10-4: Transmit Message Definition register (TCON)

Symbol	7	6	5	4	3	2	1	0	Address-offset[Note]	After Reset	R/W
TCON	IDE	RTR	0	0	DLC3	DLC2	DLC1	DLC0	t0H	undefined	R/W

Note: t = 00, 01 (address index for the 2 Transmit Buffers, see 10.6 “Message Buffer Configuration” on page 322).

IDE	Identifier Extension Select
0	Transmit standard frame message; 11 bit identifier
1	Transmit extended frame message; 29 bit identifier

RTR	Remote Transmission Select
0	Transmit data frames
1	Transmit remote frames

DLC3	DLC2	DLC1	DLC0	Data Length Code Selection of Transmit Message
0	0	0	0	0 data bytes
0	0	0	1	1 data bytes
0	0	1	0	2 data bytes
0	0	1	1	3 data bytes
0	1	0	0	4 data bytes
0	1	0	1	5 data bytes
0	1	1	0	6 data bytes
0	1	1	1	7 data bytes
1	0	0	0	8 data bytes
Others than above				Note

Remark: The control field describes the format of frame that is generated and its length. The reserved bits of the CAN protocol are always transferred in dominant state (0).

Note: The data length code selects the number of bytes which have to be transmitted. Valid entries for the data length code (DLC) are 0 to 8. If a value greater than 8 is selected, 8 bytes are transmitted in the data frame. The Data Length Code is specified in DLC3 through DLC0.

(2) Transmit Identifier Definition

These memory locations set the message identifier in the arbitration field of the CAN protocol.

IDTX0 to IDTX4 register can be set with an 1-bit or an 8-bit memory manipulation instruction.

Figure 10-5: Transmit Identifier register (IDTXn) (n = 0 to 4)

Symbol	7	6	5	4	3	2	1	0	Address-offset[Note]	After Reset	R/W
IDTX0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	t2H	undefined	R/W
IDTX1	ID20	ID19	ID18	0	0	0	0	0	t3H	undefined	R/W
IDTX2	ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	t4H	undefined	R/W
IDTX3	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	t5H	undefined	R/W
IDTX4	ID1	ID0	0	0	0	0	0	0	t6H	undefined	R/W

Note: t = 00, 01 (address index for the 2 Transmit Buffers, see 10.6 “Message Buffer Configuration” on page 322).

Remark: If a standard frame is defined by the IDE bit in the TCON byte then only IDTX0 and IDTX1 are used. IDTX2 to IDTX4 are free for use by the CPU for application needs.

(3) Transmit Data Definition

These memory locations set the transmit message data of the data field in the CAN frame.

DATA0 to DATA7 can be set with an 1-bit or an 8-bit memory manipulation instruction.

Figure 10-6: Transmit Data

Symbol	7	6	5	4	3	2	1	0	Address-offset[Note]	After Reset	R/W
DATA0									t8H	undefined	R/W
DATA1									t9H	undefined	R/W
DATA2									tAH	undefined	R/W
DATA3									tBH	undefined	R/W
DATA4									tCH	undefined	R/W
DATA5									tDH	undefined	R/W
DATA6									tEH	undefined	R/W
DATA7									tFH	undefined	R/W

Note: t = 00, 01 (address index for the 2 Transmit Buffers, see 10.6 “Message Buffer Configuration” on page 322).

Remark: Unused data bytes that are not used by the definition in the DLC bits in the TCON byte are free for use by the CPU for application needs.

10.9 Receive Message Buffer Structure

The DCAN has up to 16 receive buffers. The number of used buffers is defined by the MCNTn register. Unused receive buffers can be used as application RAM for the CPU. The received data is stored directly in this RAM area.

The 16 buffers have a 16 byte data structure for standard and extended frames with a capacity of up to 8 data bytes per message. The structure of the receive buffer is similar to the structure of the transmit buffers. The semaphore bits DN and MUC enable a secure reception detection and data handling. For the first 8 receive message buffers the successful reception is mirrored by the DN-flags in the RMESn register.

The receive interrupt request can be enabled/disabled for each used buffer separately.

10.10 Receive Message Buffer Format

Name	Address-offset Note 1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IDCON	r0H	0	0	0	0	0	ENI	RTR	IDE
DSTAT	r1H	DN	MUC	R1	R0	DLC			
IDREC0	r2H	ID standard part							
IDREC1	r3H	ID standard part			0	0	0	0	RTRREC Note 2
IDREC2	r4H	ID extended part							
IDREC3	r5H	ID extended part							
IDREC4	r6H	ID extended part		0	0	0	0	0	0
	r7H	unused							
DATA0	r8H	Message data byte 0							
DATA1	r9H	Message data byte 1							
DATA2	rAH	Message data byte 2							
DATA3	rBH	Message data byte 3							
DATA4	rCH	Message data byte 4							
DATA5	rDH	Message data byte 5							
DATA6	rEH	Message data byte 6							
DATA7	rFH	Message data byte 7							

Notes: 1. r = 02 to 11 (address index for the 14 Receive Buffers, see 10.6 “Message Buffer Configuration” on page 322).
This address is a relative offset to the start address of the receive buffer.

2. RTRREC is the received value of the RTR message bit when this buffer is used together with a mask function.
By using the mask function a successfully received identifier overwrites the bytes IDREC0 and IDREC1 for standard frame format and IDREC0 to IDREC4 for extended frame format. For the RTRREC bit exist two modes:

- a) RTR bit in the MCON byte of the dedicated mask is set to 0. In this case RTRREC will always be written to 0 together with the update of the IDn bits in IDREC1. The received frame type (data or remote) is defined by the RTR bit in IDCON of the buffer.
- b) RTR bit in the MCON byte of the dedicated mask is set to 1 (data and remote frames are accepted). In this case the RTR bit in IDCON has no meaning. The received message type passed the mask is shown in RTRREC.

3. If a buffer is not assigned to a mask function (mask 1, mask 2 or global mask) the bytes IDREC0 to IDREC4 are only read for comparing. During initialization the RTRREC should be defined to 0.

(1) Receive Control Bits Definition

The memory location labelled IDCON defines the kind of frame (data of remote frame with standard or extended format) that should be monitored for this associated buffer and if a successful reception should signalled by the receive interrupt.

IDCON can be set with an 1-bit or an 8-bit memory manipulation instruction.

Figure 10-7: Control Bits for Receive Identifier register (IDCON)

Symbol	7	6	5	4	3	2	1	0	Address-offset[Note]	After Reset	R/W
IDCON	0	0	0	0	0	ENI	RTR	IDE	r0H	undefined	R/W

Note: r = 02 to 11 (address index for the 14 Receive Buffers, see 10.6 “Message Buffer Configuration” on page 322).

Figure 10-8: Control Bits for Receive Identifier register (IDCON)

IDE	Identifier Extension Select
0	Receive standard frame message; 11 bit identifier
1	Receive extended frame message; 29 bit identifier

RTR	Remote Transmission Select
0	Receive data frames
1	Receive remote frames

ENI	Enable Interrupt on Receive[Note]
0	No interrupt generated
1	Generate receive interrupt after reception of valid message

The control bits define the type of message that is transferred in the associated buffer if this type of message appears on the bus.
This byte will never be written by the DCAN. Only CPU can change this byte.

Note: The user has to define with the ENI bit if he wants to set a receive interrupt request when new data is received on this buffer.

(2) Receive Status Bits Definition

The memory location labelled DSTAT sets the receive status bits of the arbitration field of the CAN protocol.
DSTAT can be set with an 1-bit or an 8-bit memory manipulation instruction.

Figure 10-9: Receive Status Bits register (DSTAT) (1/2)

Symbol	7	6	5	4	3	2	1	0	Address-offset[Note]	After Reset	R/W
DSTAT	DN	MUC	R1	R0	DLC3	DLC2	DLC1	DLC0	r1H	undefined	R/W

Note: r = 02 to 11 (address index for the 14 Receive Buffers, see 10.6 “Message Buffer Configuration” on page 322).

The receive status reflects the current status of a message. It signals whether new data is stored or if the DCAN currently transfers data into this buffer.
In addition the data length of the last transferred data and the reserved bits of the protocol are shown.

DN	Data New
0	No change in data
1	Data changed

The DCAN-module sets DNn, when it starts storing a message from the shadow buffer into the receive buffer.

The CPU needs to clear this bit, to signal by itself that it has read the data. During initialisation of the buffers the DNn bit should also be cleared. Otherwise the CPU gets no information of an update of the buffer after a successful reception.

MUC	Memory Update
0	CAN does not access data part
1	CAN is transferring new data to message buffer

The DCAN-module sets MUC when it starts transferring a message into the buffer and clears the MUC bit when the transfer is finished.

Figure 10-9: Receive Status Bits register (DSTAT) (2/2)

R1	Reserved Bit 1
0	Reserved bit 1 of received message was “0”
1	Reserved bit 1 of received message was “1”

R0	Reserved Bit 0
0	Reserved bit 0 of received message was “0”
1	Reserved bit 0 of received message was “1”

DLC3	DLC2	DLC1	DLC0	Data Length Code Selection of Receive Message
0	0	0	0	0 data bytes
0	0	0	1	1 data bytes
0	0	1	0	2 data bytes
0	0	1	1	3 data bytes
0	1	0	0	4 data bytes
0	1	0	1	5 data bytes
0	1	1	0	6 data bytes
0	1	1	1	7 data bytes
1	0	0	0	8 data bytes
Others than above				**Note**

DSTAT is written by the DCAN two times during message storage:
At the first access to this buffer DN = 1, MUC = 1, reserved bits and DLCx (x = 3 to 0) are written.
At the last access to this buffer DN = 1, MUC = 0, reserved bits and DLCx (x = 3 to 0) are written.

Note: Valid entries for the data length code are 0 to 8. If a value higher than 8 is received, 8 bytes are stored in the data frame with the data length code send in the DLC of the message.

(3) Receive Identifier Definition

These memory locations define the receive identifier of the arbitration field of the CAN protocol.

IDREC0 to IDREC4 can be set with an 1-bit or an 8-bit memory manipulation instruction.

Figure 10-10: Receive Identifier

Symbol	7	6	5	4	3	2	1	0	Address-offset[Note]	After Reset	R/W
IDREC0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	r2H	undefined	R/W
IDREC1	ID20	ID19	ID18	0	0	0	0	RTR_{REC}	r3H	undefined	R/W
IDREC2	ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	r4H	undefined	R/W
IDREC3	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	r5H	undefined	R/W
IDREC4	ID1	ID0	0	0	0	0	0	0	r6H	undefined	R/W

Note: r = 02 to 11 (address index for the 14 Receive Buffers, see 10.6 “Message Buffer Configuration” on page 322).

The identifier of the receive message has to be defined during the initialization of the DCAN.
The DCAN uses this data for the comparison with the identifiers received on the CAN bus. For normal message buffers without mask function this data is only read by the DCAN for comparison. In combination with a mask function this data is overwritten by the received ID that has passed the mask.

The identifier of the receive messages should not be changed without being in the initialization phase or setting the receive buffer to redefinition in the RDEF register, because the change of the contents can happen at the same time when the DCAN uses the data for comparison. This can cause inconsistent data storing to this buffer and also of the ID-port in case of using mask function.

Remarks: 1. The unused parts of the identifier may be written by the DCAN to “0”. They are not released for other use by the CPU.

2. RTRREC is the received value of the RTR message bit when this buffer is used together with a mask function.
By using the mask function a successfully received identifier overwrites the bytes IDREC0 and IDREC1 registers for standard frame format and IDREC0 to IDREC4 registers for extended frame format. For the RTRREC bit exists two modes:

a) RTR bit in the MCON register of the dedicated mask is set to “0”. In this case RTRREC register will always be written to “0” together with the update of the IDx bits (x = 18 to 20) in IDREC1. The received frame type (data or remote) is defined by the RTR bit in IDCON of the buffer.

b) RTR bit in the MCON byte of the dedicated mask is set to “1” (data and remote frames are accepted). In this case the RTR bit in IDCON register has no meaning. The received message type passed the mask is shown in RTRREC register.

If a buffer is not dedicate to a mask function (mask 1, mask 2 or global mask) the IDREC0 to IDREC4 registers are only read for comparing. During initialization the RTRREC register should be defined to “0”.

(4) Receive Message Data Part

These memory locations set the receive message data part of the CAN protocol.

DATA0 to DATA7 can be set with an 1-bit or an 8-bit memory manipulation instruction.

Figure 10-11: Receive Data

Symbol	7	6	5	4	3	2	1	0	Address-offset[Note]	After Reset	R/W
DATA0									r8H	undefined	R/W
DATA1									r9H	undefined	R/W
DATA2									rAH	undefined	R/W
DATA3									rBH	undefined	R/W
DATA4									rCH	undefined	R/W
DATA5									rDH	undefined	R/W
DATA6									rEH	undefined	R/W
DATA7									rFH	undefined	R/W

Note: r = 02 to 11 (address index for the 14 Receive Buffers, see 10.6 “Message Buffer Configuration” on page 322).

The DCAN stores received data bytes in this memory area. Only those data bytes which are actually received and match with the identifier are stored in the receive buffer memory area.
If the DLC is less than eight, the DCAN will not write additional bytes exceeding the DLC value up to eight. The DCAN stores a maximum of 8 bytes (according to the CAN protocol rules) even when the received DLC is greater than eight.

10.11 Mask Function

Table 10-1: Mask Function

Name	Address-offset **Note**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCON	m0H							RTR	
	m1H	Unused							
MREC0	m2H	ID standard part							
MREC1	m3H	ID standard part			0	0	0	0	0
MREC2	m4H	ID extended part							
MREC3	m5H	ID extended part							
MREC4	m6H	ID extended part		0	0	0	0	0	0
	m7H	Unused							
	m8H	Unused							
	m9H	Unused							
	mAH	Unused							
	mBH	Unused							
	mCH	Unused							
	mDH	Unused							
	mEH	Unused							
	mFH	Unused							

Note: m = 2, 4 (address index for the 2 Mask Buffers, see 10.6 “Message Buffer Configuration” on page 322).

Receive Message buffer 0 and 2 may be switched for masked operation with the mask control register (MASKCn). In this case the message does not hold message identifier and data of the frame. Instead, it holds identifier and RTR mask information for masked compares of the next higher message buffer number. In case of a global mask select, it keeps mask information for all higher message buffer numbers.

A mask does not store any information about identifier length. The same mask can therefore be used for both types of frames (standard and extended) during global mask operation.

All unused bytes can be used by the CPU for application needs.

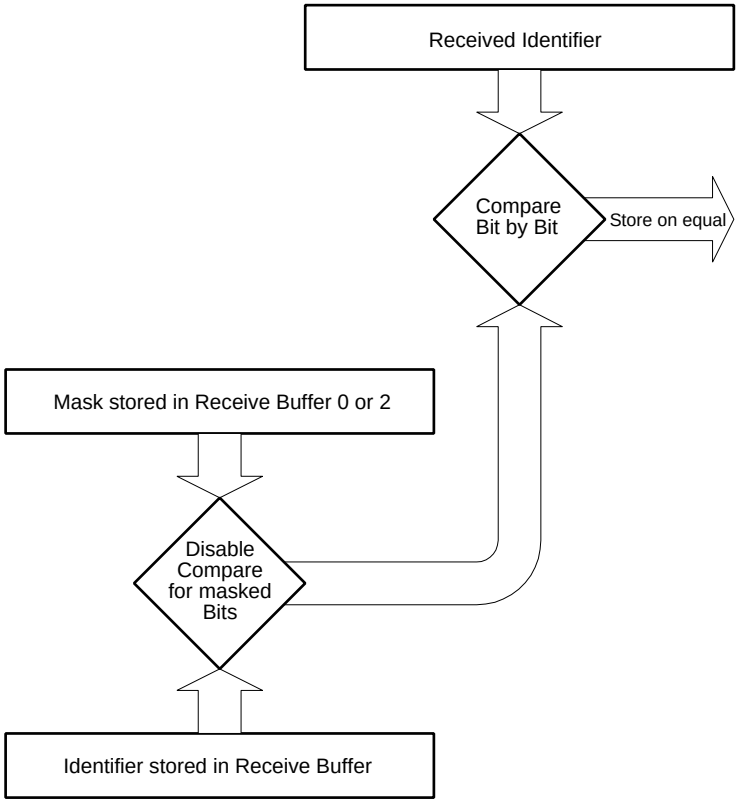
(1) Identifier Compare with Mask

The identifier compare with mask provides the possibility to exclude some bits from the comparison process. That means each bit is ignored when the corresponding bit in the mask definition is set to one.

The setup of the mask control register (MASKCn) defines which receive buffer is used as a mask and which receive buffer uses which mask for comparison.

The mask does not include any information about the identifier type to be masked. This has to be defined within the dedicated receive buffer. Therefore a global mask can serve for standard receive buffers at the same time as for extended receive buffer.

Figure 10-12: Identifier Compare with Mask



This function implements the so called basic-CAN behaviour.

In this case, the type of identifier is fixed to standard or extended by the setup of the IDE bit in the receive buffer. The comparison of the RTR bit can also be masked. It is possible to receive data and remote frames on the same masked receive buffer.

The following information is stored in RAM:

- Identifier (11 or 29 bit as defined by IDE bit)
- Remote bit (RTR)
- Reserved bits
- Data length code (DLC)
- Data bytes as defined by DLC

Caution: All writes into the DCAN memory are byte accesses. Unused bits in the same byte will be written zero. Unused bytes will not be written and are free for application use by the CPU.

(2) Mask Identifier Control Bit Definition register (MCON)

The memory location labelled MCON sets the mask identifier control bit of the CAN protocol.

MCON can be set with an 1-bit or an 8-bit memory manipulation instruction.

Figure 10-13: Control Bits for Mask Identifier

Symbol	7	6	5	4	3	2	1	0	Address-offset[Note]	After Reset	R/W
MCON	0	0	0	0	0	0	RTR	0	m0H	undefined	R/W

Note: m = 2, 4 (address index for the 2 Mask Buffers, see 10.6 “Message Buffer Configuration” on page 322).

RTR	Remote Transmission Select
0	Check RTR bit of received message **Note 1**
1	Receive message independent from RTR bit **Note 2**

Notes: 1. For RTR = 0 the received frame type (data or remote) is defined by the RTR bit in IDCON of the dedicated buffer. In this case RTRREC will always be written to “0” together with the update of the IDx bits (x = 18 to 20) in IDREC1.

2. In case RTR is set to “1” RTR bit in IDCON of the dedicated receive buffer has no meaning. The received message type passed the mask is shown in the RTRREC register.

(3) Mask Identifier Definition

These memory locations set the mask identifier definition of the DCAN.

MREC0 to MREC4 can be set with an 1-bit or an 8-bit memory manipulation instruction.

Figure 10-14: Mask Identifier register (MRESC0) (n = 0 to 4)

Symbol	7	6	5	4	3	2	1	0	Address-offset[Note]	After Reset	R/W
MREC0	MID28	MID27	MID26	MID25	MID24	MID23	MID22	MID21	m2H	undefined	R/W
MREC1	MID20	MID19	MID18	0	0	0	0	0	m3H	undefined	R/W
MREC2	MID17	MID16	MID15	MID14	MID13	MID12	MID11	MID10	m4H	undefined	R/W
MREC3	MID9	MID8	MID7	MID6	MID5	MID4	MID3	MID2	m5H	undefined	R/W
MREC4	MID1	MID0	0	0	0	0	0	0	m6H	undefined	R/W

Note: m = 2, 4 (address index for the 2 Mask Buffers, see 10.6 “Message Buffer Configuration” on page 322).

MIDn	Mask Identifier Bit (n = 0...28)
0	Check IDn bit in IDREC0 through IDREC4 of received message
1	Receive message independent from IDn bit

10.12 Operation of the DCAN Controller

10.12.1 DCAN Control Register (DCANCn) (n = 0, 1)

With the DCANEN bit of the DCAN Control register (DCANCn) the DCAN channel n is enable/disable.

DANCn can be set with an 1-bit or an 8-bit memory manipulation instruction.

Figure 10-15: DCAN Control Register (DCANCn) (n = 0, 1)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset
DCANC0	0	0	0	0	0	0	0	DCANEN	XXnF FF20H	00H
	R	R	R	R	R	R	R	R/W	n = 3, 7, B	
DCANC1	0	0	0	0	0	0	0	DCANEN	XXnF FD20H	00H
	R	R	R	R	R	R	R	R/W	n = 3, 7, B	

DCANENn	accessing data RAM operation
0	Disable accessing data RAM operation
1	Enable accessing data RAM operation

10.12.2 CAN Control Register (CANCn)

The main functions are defined in the CAN control register CANCn.

CANCn can be set with an 8-bit memory manipulation instruction.

Remark: The bits 0, 2 and 4 are 1-bit accessible.

Figure 10-16: CAN Control Register (CANCn) (n = 0, 1) (1/2)

Symbol	7	6	5	<4>	3	<2>	1	<0>	Address	After Reset
CANC0	RxF	TxF	0	SOFE	SOFSEL	SLEEP	STOP	INIT	XXnF FF21H	01H
	R	R	R	R/W	R/W	R/W	R/W	R/W	n = 3, 7, B	
CANC1	RxF	TxF	0	SOFE	SOFSEL	SLEEP	STOP	INIT	XXnF FD21H	01H
	R	R	R	R/W	R/W	R/W	R/W	R/W	n = 3, 7, B	

INIT	Initialize
0	Normal operation
1	Initialization mode

The INIT is the request bit to control the DCAN. INIT starts and stops the CAN protocol activities. Due to bus activities disabling the DCAN is not allowed any time. Therefore changing the INIT bit must not have an immediate effect to the CAN protocol activities. Setting the INIT bit is a request only.
The INITSTAT bit in the CANESn register reflects if the request has been granted. The registers MCNTn, SYNC0n, SYNC1n, and MASKCn are write protected while INIT is cleared independently of INITSTAT. Any write to these registers when INIT is set and the initialisation mode is not confirmed by the INITSTAT bit can have unexpected behaviour to the CAN bus.

SLEEP	Sleep/Stop Request for CAN protocol
0	Normal operation
1	CAN protocol goes to sleep or stop mode depending on STOP bit

Caution: The DCAN Sleep or DCAN Stop mode can not be requested as long as the WAKE bit in CANES is set.

Figure 10-16: CAN Control Register (CANCn) (n = 0, 1) (2/2)

STOP	Stop Mode Selection
0	Normal sleep operation / Sleep mode is released when a transition on the CAN bus is detected
1	Stop operation / Sleep mode is cancelled only by CPU access. No wake up from CAN bus

The clock supply to the DCAN is switched off during initialisation, DCAN Sleep, and DCAN Stop mode. All modes are only accepted while CAN protocol is in idle state, whereby the CRXD pin must be recessive (= high level). A sleep or stop request out of idle state is rejected and the WAKE bit in CANES is set. DCAN Sleep and DCAN Stop mode can be requested in the same manner. The only difference is that the DCAN Stop mode prevents the wake up by CAN bus activity.

The DCAN Sleep mode condition is cancelled under following conditions:

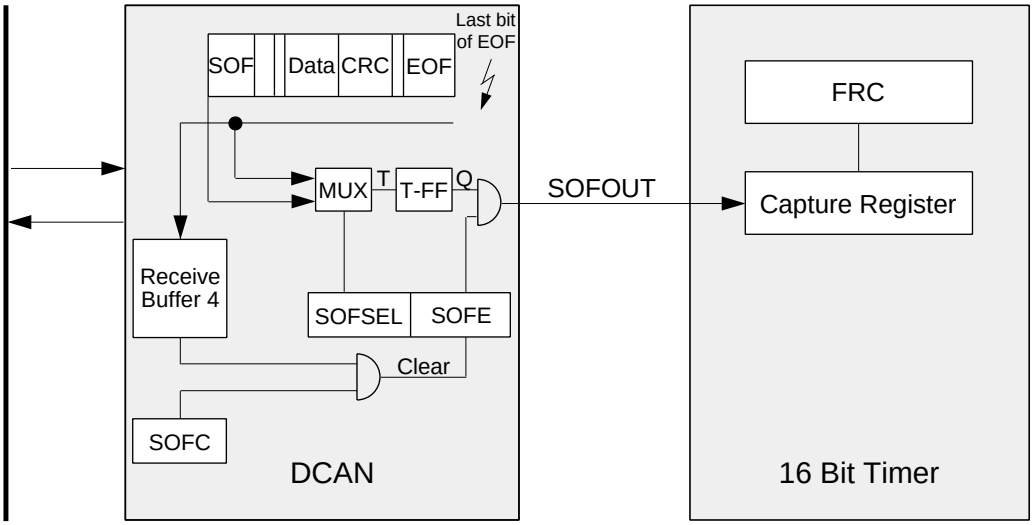
a) CPU clears the SLEEP bit.
b) Any transition while idle state on CAN bus (STOP bit = 0)
c) CPU sets SLEEP, but CAN protocol is active due to bus activity.

The WAKE bit in CANESn is set under condition b) and c).

SOFE	Start of Frame Enable
0	SOFOUT does not change
1	SOFOUT toggles depending on the selected mode

SOFSEL	Start of Frame Output Function Select
0	Last bit of EOF is used to generate the time stamp
1	SOF is used to generate the time stamp

Figure 10-17: DCAN Time Stamp Support



The generation of an SOFOUT signal can be used for time measurements and for global time base synchronisation of different CAN nodes as a prerequisite for time triggered communication.

SOFSEL	SOFC	SOFE	SOFOUT Function
x	x	0	Time stamp function disabled
0	x	1	Toggles with each EOF
1	0	1	Toggles with each start of frame on the CAN Bus
1	1	1	Toggles with each start of frame on the CAN bus. Clears the SOFE bit when DCAN starts to store a message in receive buffer 4

SOFCn is located in the synchronisation register SYNC1n.
$\overline{\text{RESET}}$ and setting of the INIT bit of CANCn register clears the SOFOUT to 0.

Figure 10-18: Time Stamp Function

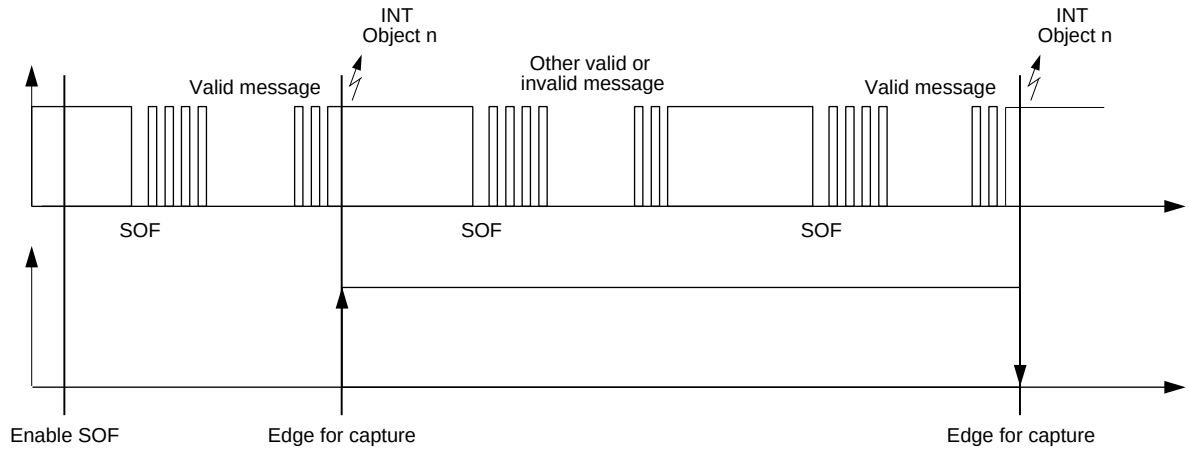


Figure 10-19: SOFOUT Toggle Function

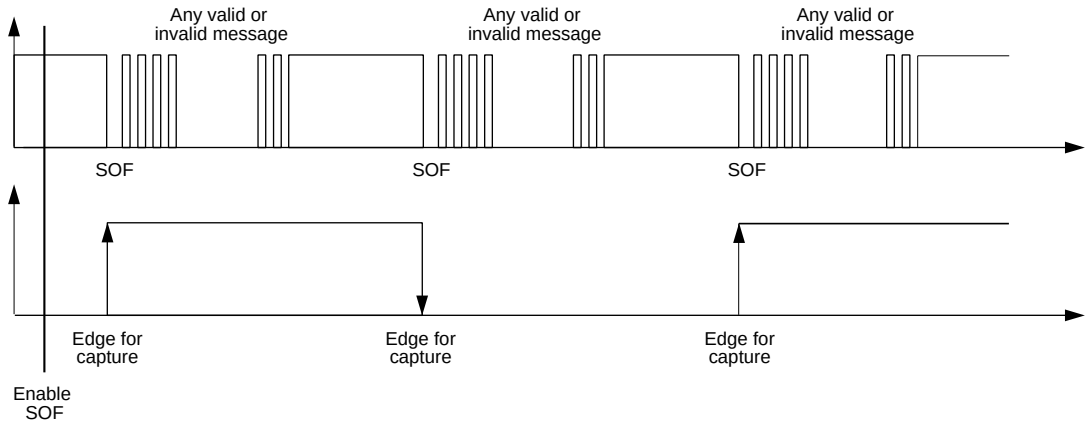


Figure 10-20: Global Time System Function

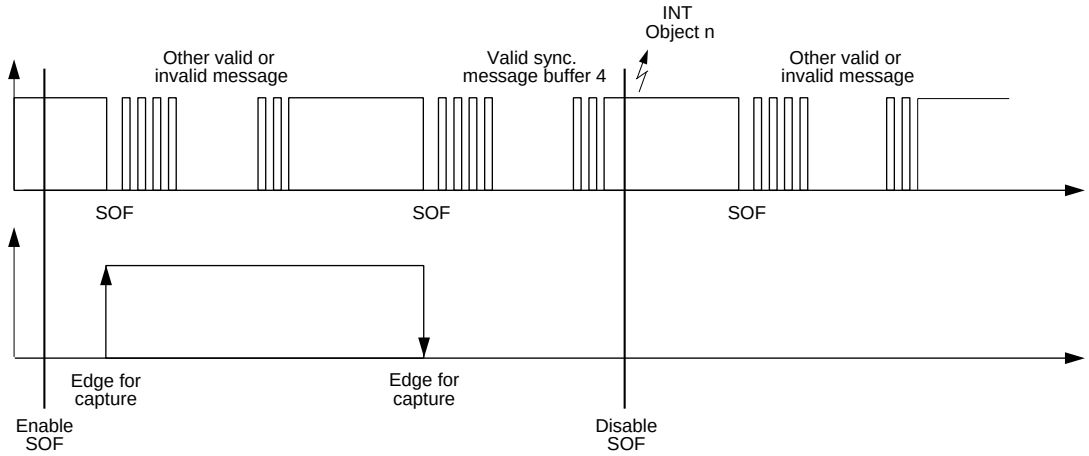


Figure 10-21: Transmission/Reception Flag

TXF	Transmission Flag
0	No transmission
1	Transmission active on CAN bus **Note**

Note: Transmission is active until intermission is completed.

RXF	Reception Flag
0	No data on the CAN bus
1	Reception active on the CAN bus

The TXF bit and RXF bit of CANCn register show the present status of the DCAN on the bus. If both bits are cleared, the bus is in idle state. RXFn and TXFn are read only bits. During initialisation mode both bits do not reflect the bus status.

10.12.3 DCAN Error Status Register

This register shows the status of the DCAN.

CANESn has to be set with an 8-bit memory manipulation instruction.

Figure 10-22: DCAN Error Status Register (CANESn) (n = 0, 1) (1/3)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset
CANES0	BOFF	RECS	TECS	0	INIT-STATE	VALID	WAKE	OVER	XXnF FF25H	00H[Note]
	R	R	R	R	R	R/W	R/W	R/W	n = 3, 7, B	
CANES1	BOFF	RECS	TECS	0	INIT-STATE	VALID	WAKE	OVER	XXnF FD25H	00H[Note]
	R	R	R	R	R	R/W	R/W	R/W	n = 3, 7, B	

Note: $\overline{\text{RESET}}$ input sets CANCn to 00H. The $\overline{\text{RESET}}$ sets the INIT bit in CANCn register, therefore CANESn will be read as 08h after $\overline{\text{RESET}}$ release.

Remark: BOFF, RECS, TECS and INITSTATE are read only bits.

Caution: Don't use bit operations on this SFR.
The VALID, WAKE and OVER bits have a special behavior during CPU write operations.

- **Writing a "0" to them do not change them.**
- **Writing a "1" clears the associated bit.**

This avoids any timing conflicts between CPU access and internal activities. An internal set condition of a bit overrides a CPU clear request at the same time.

BOFF	Bus Off Flag
0	Transmission error counter ≤ 255
1	Transmission error counter > 255

BOFFn is cleared after receiving 128 x 11 bits recessive state (Bus idle) or by issuing a hard DCAN reset with the TLRES bit in the MCNTn register [Note]. An interrupt is generated when the BOFF bit changes its value.

Note: Issuing TLRES bit may violate the minimum recovery time as defined in ISO-11898.

RECS	Reception error counter status
0	Reception error counter < 96
1	Reception error counter ≥ 96 / Warning level for error passive reached

RECS is updated after each reception. An interrupt is generated when RECS changes its value.

Figure 10-22: DCAN Error Status Register (CANESn) (n = 0, 1) (2/3)

TECS	Transmission error counter status
0	Transmission error counter < 96
1	Transmission error counter ≥ 96 / Warning level for error passive reached

TECS is updated after each reception. An interrupt is generated when TECS changes its value.

INITSTATE	INIT accepted
0	CAN is in normal operation
1	CAN is stopped and ready to accept new configuration data

INITSTATEn changes with a delay to the INIT bit in CANCn register. The delay depends on the current bus activity and the time to set all internal activities to inactive state. This time can be several bit times long. While BOFF bit is set, a request to go into the initialisation mode by setting the INIT bit is ignored. In this case the INITSTATE bit won't be set until the Bus off state is left.

VALID	Valid protocol activity detected
0	No valid message detected by the CAN protocol
1	Error free message reception from CAN bus

This bit shows valid protocol activities independent from the message definitions and the RXONLY bit setting in SYNC1n register. VALID is updated after each reception. After detecting successfully a complete protocol the VALID bit will be set at the end of the frame.

Cautions: 1. The VALID bit is cleared if CPU writes “1” to it, or when the INIT bit in CANCn register is set.

2. Writing a “0” to the valid bit has no influence.

WAKE	Wake up Condition
0	Normal operation
1	Sleep mode has been cancelled or sleep/stop mode request was not granted

This bit is set and an error interrupt is generated under the following circumstances:

a) A CAN bus activity occurs during DCAN Sleep mode.

b) Any attempt to set the SLEEP bit in the CAN control register during receive or transmit operation will immediately set the WAKE bit.

The CPU must clear this bit after recognition in order to receive further error interrupts, because the error interrupt line is kept active as long as this bit is set.

Cautions: 1. The WAKE bit is cleared to “0” if CPU writes “1” to it, or when the INIT bit in CANCn register is set.

2. Writing a “0” to the WAKE bit has no influence.

Figure 10-22: DCAN Error Status Register (CANESn) (n = 0, 1) (3/3)

OVER	Overrun Condition
0	Normal operation
1	Overrun occurred during access to RAM

The overrun condition is set whenever the CAN can not perform all RAM accesses that are necessary for comparing and storing received data or fetching transmitted data. Typically, the overrun condition is encountered when the frequency for the macro is too low compared to the programmed baud rate. An error interrupt is generated at the same time.

The DCAN interface will work properly (i. e. no overrun condition will occur) with the following settings: The DCAN clock as defined with the PRM bits in the BRPRSn register is set to a minimum of 16 times of the CAN baudrate **and** the selected CPU clock (defined in the PCC register) is set to a minimum of 16 times of the baudrate.

Possible reasons for an overrun condition are:

- Too many messages are defined.
- DMA access to RAM area is too slow compared to the CAN Baudrate.

The possible reactions of the DCAN differ depending on the situation, when the overrun occurs.

Table 10-2: Possible Reactions of the DCAN

Overrun Situation	When detected	DCAN Behavior
Cannot get transmit data.	Next data byte request from protocol. Immediate during the frame.	The frame itself conforms to the CAN specification, but its content is faulty. Corrupted data or ID in the frame. TXRQx bit (x = 0, 1) is not cleared. DCAN will retransmit the correct frame after synchronization to the bus.
Cannot store receive data.	Data storage is ongoing during the six bit of the next frame.	Data in RAM is inconsistent. No receive flags. DN and MUC bit may be set in message.
Cannot get data for ID comparison	ID compare is ongoing during six bits of next frame.	Message is not received and its data is lost.

10.12.4 CAN Transmit Error Counter

This register shows the transmit error counter.

TECn register can be read with an 8-bit memory manipulation instruction.

Figure 10-23: Transmit Error Counter register (TECn) (n = 0, 1)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset
TEC0	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	XXnF FF26H	00H
	R	R	R	R	R	R	R	R	n = 3, 7, B	
TEC1	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	XXnF FD26H	00H
	R	R	R	R	R	R	R	R	n = 3, 7, B	

The transmit error counters reflects the status of the error counter for transmission errors as it is defined in the CAN Protocol.

10.12.5 CAN Receive Error Counter

This register shows the receive error counter.

RECn can be read with an 8-bit memory manipulation instruction.

Figure 10-24: Receive Error Counter Register (RECn) (n = 0, 1)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset
REC0	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	XXnF FF27H	00H
	R	R	R	R	R	R	R	R	n = 3, 7, B	
REC1	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	XXnF FD27H	00H
	R	R	R	R	R	R	R	R	n = 3, 7, B	

The receive error counters reflects the status of the error counter for reception errors as it is defined in the CAN Protocol (ISO 11898).

10.12.6 Message Count Register

This register sets the number of receive message buffers and allocates the RAM area of the receive message buffers which are handled by the DCAN-module.

MCNTn can be read with an 8-bit memory manipulation instruction.

Figure 10-25: Message Count Register (MCNTn) (n = 0, 1)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset
MCNT0	1	1	TLRES	MCNT4	MCNT3	MCNT2	MCNT1	MCNT0	XXnF FF28H	C0H
	R	R	R/W	R/W	R/W	R/W	R/W	R/W	n = 3, 7, B	
MCNT1	1	1	TLRES	MCNT4	MCNT3	MCNT2	MCNT1	MCNT0	XXnF FD28H	C0H
	R	R	R/W	R/W	R/W	R/W	R/W	R/W	n = 3, 7, B	

TLRES	Reset function for CAN Protocol Machine[Note]
0	No Reset is issued
1	Reset of CAN protocol machine is issued if DCAN is in bus off state, DCAN will enter INIT state (CANC.0 = 1)

Note: Issuing TLRES bit may violate the minimum recovery time as defined in ISO-11898.

MCNT4	MCNT3	MCNT2	MCNT1	MCNT0	Receive Message Count
0	0	0	0	0	Setting prohibited
0	0	0	0	1	1 receive buffer
0	0	0	1	0	2 receive buffer
0	0	0	1	1	3 receive buffer
0	0	1	0	0	4 receive buffer
0	0	1	0	1	5 receive buffer
0	0	1	1	0	6 receive buffer
0	0	1	1	1	7 receive buffer
0	1	0	0	0	8 receive buffer
0	1	0	0	1	9 receive buffer
0	1	0	1	0	10 receive buffer
0	1	0	1	1	11 receive buffer
0	1	1	0	0	12 receive buffer
0	1	1	0	1	13 receive buffer
0	1	1	1	0	14 receive buffer
0	1	1	1	1	15 receive buffer
1	0	0	0	0	16 receive buffer
1	x	x	x	x	Setting prohibited, will be automatically changed to 16

10.13 Baudrate Generation

(1) Bit rate Prescaler Register

This register sets the clock for the DCAN (internal DCAN clock) and the number of clocks per time quantum (TQ).

BRPRSn can be set with an 8-bit memory manipulation instruction.

Figure 10-26: Bit Rate Prescaler Register (BRPRSn) (n = 0, 1) (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset
BRPRS0	PRM1	PRM0	BRPRS5	BRPRS4	BRPRS3	BRPRS2	BRPRS1	BRPRS0	XXnF FF29H	00H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	n = 3, 7, B	
BRPRS1	PRM1	PRM0	BRPRS5	BRPRS4	BRPRS3	BRPRS2	BRPRS1	BRPRS0	XXnF FD29H	00H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	n = 3, 7, B	

The PRMx (x = 0, 1) bits define the clock source for the DCAN operation.
The PRM selector defines the input clock to the DCAN Macro and influences therefore all DCAN activities.
Writing to the BRPRSn register is only allowed during initialization mode. Any write to this register when INIT bit is set in CANCn register and the initialization mode is not confirmed by the INITSTATE bit of CANESn register can have unexpected behaviour to the CAN bus.

PRM1	PRM0	Input Clock Selector for DCAN Clock
0	0	f_{XX} is input for DCAN
0	1	$f_{XX}/2$ is input for DCAN
1	0	$f_{XX}/4$ is input for DCAN
1	1	$f_{XX}/8$ is input for DCAN (at RESET)

Figure 10-26: Bit Rate Prescaler Register (BRPRSn) (n = 0, 1) (2/2)

BRPRSn defines the number of DCAN clocks applied for one TQ.
For BRPRSn two modes are available depending on the TLMODE bit in the SYNC1n register.

Setting of BRPRSx (x = 5 to 0) for TLMODE=0:

BRPRS5	BRPRS4	BRPRS3	BRPRS2	BRPRS1	BRPRS0	Bit Rate Prescaler[Note]
0	0	0	0	0	0	2
0	0	0	0	0	1	4
0	0	0	0	1	0	6
0	0	0	0	1	1	8
.	.	.	.	.	.	.
.	.	.	.	.	.	2 x BRPRSn[5-0] + 2
.	.	.	.	.	.	.
1	1	1	0	1	0	118
1	1	1	0	1	1	120
1	1	1	1	0	0	122
1	1	1	1	0	1	124
1	1	1	1	1	0	126
1	1	1	1	1	1	128

Note: The bit rate prescaler value represents the DCAN clocks per TQ.

(2) Synchronization Control Registers 0 and 1

These registers define the CAN bit timing. They define the length of one data bit on the CAN bus, the position of the sample point during the bit timing, and the synchronization jump width. The range of resynchronization can be adapted to different CAN bus speeds. Additionally, some modes can be selected in SYNC1n register.

SYNC0n and SYNC1n can be read or written with an 8-bit memory manipulation instruction.

Figure 10-27: Synchronization Control Register 0 (SYNC0n) (n = 3, 7, B)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
SYNC00	SPT2	SPT1	SPT0	DBT4	DBT3	DBT2	DBT1	DBT0	XXnF FF2AH	18H	R/W

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
SYNC01	SPT2	SPT1	SPT0	DBT4	DBT3	DBT2	DBT1	DBT0	XXnF FD2AH	18H	R/W

Figure 10-28: Synchronization Control Register 1 (SYNC1n) (n = 3, 7, B)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset	R/W
SYNC10	0	SOFC	SAMP	RXONLY	SJW1	SJW0	SPT4	SPT3	XXnF FF2BH	0EH	R/W
SYNC11	0	SOFC	SAMP	RXONLY	SJW1	SJW0	SPT4	SPT3	XXnF FD2BH	0EH	R/W

The length of a data bit time is programmable via DBT[4-0].

DBT4	DBT3	DBT2	DBT1	DBT0	Data Bit Time
Other than under					Setting prohibited
0	0	1	1	1	8 x TQ
0	1	0	0	0	9 x TQ
0	1	0	0	1	10 x TQ
0	1	0	1	0	11 x TQ
0	1	0	1	1	12 x TQ
0	1	1	0	0	13 x TQ
0	1	1	0	1	14 x TQ
0	1	1	1	0	15 x TQ
0	1	1	1	1	16 x TQ
1	0	0	0	0	17 x TQ
1	0	0	0	1	18 x TQ
1	0	0	1	0	19 x TQ
1	0	0	1	1	20 x TQ
1	0	1	0	0	21 x TQ
1	0	1	0	1	22 x TQ
1	0	1	1	0	23 x TQ
1	0	1	1	1	24 x TQ
1	1	0	0	0	25 x TQ
Other than above					Setting prohibited

The position of the sample point within the bit timing is defined by SPT0n through SPT4n.

SPT4	SPT3	SPT2	SPT1	SPT0	Sample Point Position
Other than under					Setting prohibited
0	0	0	0	1	2 x TQ
0	0	0	1	0	3 x TQ
0	0	0	1	1	4 x TQ
0	0	1	0	0	5 x TQ
0	0	1	0	1	6 x TQ
0	0	1	1	0	7 x TQ
0	0	1	1	1	8 x TQ
0	1	0	0	0	9 x TQ
0	1	0	0	1	10 x TQ
0	1	0	1	0	11 x TQ
0	1	0	1	1	12 x TQ
0	1	1	0	0	13 x TQ
0	1	1	0	1	14 x TQ
0	1	1	1	0	15 x TQ
0	1	1	1	1	16 x TQ
1	0	0	0	0	17 x TQ
Other than above					Setting prohibited

SJW0 and SJW1 define the synchronization jump width as specified in ISO 11898.

SJW1	SJW0	Synchronisation Jump Width
0	0	1 x TQ
0	1	2 x TQ
1	0	3 x TQ
1	1	4 x TQ

Limits on defining the bit timing

The sample point position needs to be programmed between 3TQ[Note] and 17TQ, which equals a register value of **2** ≤ **SPTxn** ≤ **16** (n = 0, 1; x = 4 to 0).

The number of TQ per bit is restricted to the range from 8TQ to 25TQ, which equals a register value of **7** ≤ **DBTxn** ≤ **24** (n = 0, 1; x = 4 to 0).

The length of phase segment 2 (TSEG2) in TQ is given by the difference of TQ per bit (DBTxn) and the sample point position (SPTxn). Converted to register values the following condition applies:
2 ≤ **DBTxn - SPTxn** ≤ **8** (n = 0, 1; x = 4 to 0).

The number of TQ allocated for soft synchronization must not exceed the number of TQ for phase segment 2, but SJWyn may have as many TQ as phase segment 2:
SJWyn ≤ **DBTxn - SPTxn - 1** (n = 0, 1; x = 4 to 0; y = 0, 1).

Note: Sample point positions of 3 TQ or 4 TQ are for test purposes only. For the minimum number of TQ per bit time, 8TQ, the minimum sample point position is 5 TQ.

Example:

System clock:	fx	8 MHz
CAN parameter:	Baud rate	500 kBaud
	Sample Point	75%
	SJW	25%

At first, calculate the overall prescaler value:

$$\frac{f_X}{Baudrate} = \frac{8\ MHz}{500\ KBaud} = 16$$

16 can be split as 1 x 16 or 2 x 8. Other factors can not be mapped to the registers. Only 8 and 16 are valid values for TQ per bit. Therefore the overall prescaler value realized by BRPRSn is 2 or 1 respectively.

The following register settings apply:

Register value	Description	Bit fields
BRPRSn = 00h	Clock selector = fx	PRMn = 00b
		BRPRSx = 000000b
SYNC0n = A7h	CAN Bit in TQ = 8	DBTx = 00111b
	7 < (fx/Baudrate/bit rate prescaler) < 25]	
SYNC1n = 0zzz0100b	sample point 75% = 6 TQ	SPTx = 00101b
	SJW 25% = 2 TQ	SJWy = 01b
	1 TQ equals 2 clocks & BRPRS6, 7 are disabled	TLMODE = 0
	z depends on the setting of: - Number of sampling points - Receive only function - Use of time stamp or global time system	

The receive-only mode can be used for baudrate detection. Different baudrate configurations can be tested without disturbing other CAN nodes on the bus.

RXONLY	Receive Only Operation
0	Normal operation
1	Only receive operation, CAN does not activate transmit line

Differences to CAN protocol in the receive-only mode:

- The mode never sends an acknowledge, error frames or transmit messages.
- The error counters do not count.

The VALID bit in CANESn register reports if the DCAN interface receives any valid message.

SAMPn defines the number of sample points per bit as specified in the ISO-11898.

SAMP	Bit Sampling
0	Sample receive data one time at receive point
1	Sample receive data three times and take majority decision at sample point

SOFC works in conjunction with the SOFE and SOFSEL bits in the CAN Control Register CANCn. For detailed information please refer to the bit description of the SFR register and the time function mode.

SOFC	Start of Frame Control
0	SOFE bit is independent from CAN bus activities
1	SOFE bit will be cleared when a message for receive message 4 is received and SOF mode is selected

Caution: CPU can read SYNC0n/SYNC1n register at any time. Writing to the SYNC0n/SYNC1n registers is only allowed during initialization mode. Any write to this register when INITn is set and the initialization mode is not confirmed by the INITSTATE bit can have unexpected behavior to the CAN bus.

10.14 Function Control

10.14.1 Transmit Control

(1) Transmit Control Register

This register controls the transmission of the DCAN-module. The transmit control register (TCRn) provides complete control over the two transmit buffers and their status. It is possible to request and abort transmission of both buffers independently.

TCRn can be set with a an 8-bit memory manipulation instruction.

Caution: Don't use bit operations on this register. Also logical operations (read-modify-write) via software may lead to unexpected transmissions. Initiating a transmit request for buffer 1 while TxRQ0 ia already set, is simply achieved by writing 02h or 82h. The status of the bits for buffer 0 is not affected by this write operation.

Figure 10-29: Transmit Control Register (TCRn) (n = 0, 1) (1/3)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset
TCR0	TXP	0	TXC1	TXC0	TXA1	TXA0	TXRQ1	TXRQ0	XXnF FF22H	00H
	R/W	R	R	R	R/W	R/W	R/W	R/W	n = 3, 7, B	
TCR1	TXP	0	TXC1	TXC0	TXA1	TXA0	TXRQ1	TXRQ0	XXnF FD22H	00H
	R/W	R	R	R	R/W	R/W	R/W	R/W	n = 3, 7, B	

TXP	Transmission Priority
0	Buffer 0 has priority over buffer 1
1	Buffer 1 has priority over buffer 0

The user defines which buffer has to be send first in the case of both request bits are set. If only one buffer is requested by the TXRQx bit (x = 0, 1) bits, TXP bit has no influence.
TXCx (x = 0, 1) shows the status of the first transmission. It is updated when TXRQx (x = 0, 1) is cleared.

Figure 10-29: Transmit Control Register (TCRn) (n = 0, 1) (2/3)

TXAx	Transmission Abort Flag
0	Write: normal operation
	Read: no abort pending
1	Write: aborts current transmission request for this buffer x
	Read: abort is pending

TXCn	Transmission Complete Flag
0	Transmit was aborted / no data sent
1	Transmit was complete / abort had no effect

The TXAx bits allow to free a transmit buffer with a pending transmit request. Setting the TXAx bit (x = 0, 1) by the CPU requests the DCAN to empty its buffer by clearing the respective TXRQx bit.

The TXAx bits (x = 0, 1) have a dual function:

1. The CPU can request an abort by writing a “1” into the bit.

2. The DCAN signals whether such an request is still pending. The bit is cleared at the same time when the TXRQx bit (x = 0, 1) is cleared.

This abort does not affect any rules of the CAN protocol. A frame already started will continue to its end.

An abort operation can cause different results dependent on the time it is issued.

a) Abort is received before the start of the arbitration for transmit.
The TXCx bit (x = 0, 1) is reset showing that the buffer was not send to other nodes.

b) Abort is received during the arbitration, but arbitration is lost.
The TXCx bit (x = 0, 1) is reset showing that the buffer was not send to other nodes.

c) Abort is received during frame transmission, but transmission ends with an error.
The TXCx bit (x = 0, 1) is reset showing that the buffer was not send to other nodes.

d) Abort is received during the frame transmission and transmission ends without error.
The TXCx bit (x = 0, 1) is set showing a successful transfer of the data before the abort gets active.

In all cases the TXRQx bit (x = 0, 1) and the TXAx bit (x = 0, 1) bit will be cleared at the end of the abort operation, when the receive buffer is available again.

Cautions: 1. The bits are cleared when the INIT bit in CANCn register is set.

2. Writing a 0 to TXAx (x = 0, 1) bit has no influence

3. Do not perform read-modify-write operations on TCRn.

The TXCx bit (x = 0, 1) are updated at the end of every frame transmission or abort.

Figure 10-29: Transmit Control Register (TCRn) (n = 0, 1) (3/3)

TXRQx	Transmission Request Flag
0	Write: no influence
	Read: transmit buffer is free
1	Write: request transmission for buffer n
	Read: transmit buffer is occupied by former transmit request

The bits are checked by the DCAN immediately before the frame is started. The order in which the TXRQx bit (x = 0, 1) will be set does not matter as long as the first requested frame is not started on the bus.

The TXRQx bit (x = 0, 1) have dual function:

- 1. Request the transmission of a transmit buffer.
- 2. Inform the CPU whether a buffer is available or if it is still occupied by a former transmit request.

Setting the transmission request bit requests the DCAN to sent the buffer contents onto the bus. The DCAN clears the bit after completion of the transmission. Completion is either a normal transfer without error or an abort request.

An error during the transmission does not influence the transmit request status. The DCAN will automatically retry the transfer.

Cautions: 1. The bits are cleared when the INIT bit in CANCn is set. An already started transmission will be finished but not retransmitted in case of an error.

2. Writing a 0 to TXRQ0 bit has no influence

3. Do not use bit operations on this register.

4. Do not change data in transmit buffer n when TXRQmm bit (m = 0, 1) is set.

10.14.2 Receive Control

The receive message register mirrors the current status of the first 8 receive buffers. Each buffer has one status bit in this register. This bit is always set when a new message is completely stored out of the shadow buffer into the associated buffer. The CPU can easily find the last received message during receive interrupt handling. The bits in this register always correspond to the DNn bit in the data buffers. They are cleared when the CPU clears the DNn bit in the data buffer. The register itself is read only.

(1) Receive Message Register

This register shows receptions of messages of the DCAN-module. More than one bit set is possible.

RMESn can be read with an 1-bit or an 8-bit memory manipulation instruction.

Figure 10-30: Receive Message Register (RMESn) (n = 0, 1)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset
RMES0	DN7	DN6	DN5	DN4	DN3	DN2	DN1	DN0	XXnF FF23H	00H
	R	R	R	R	R	R	R	R	n = 3, 7, B	
RMES1	DN7	DN6	DN5	DN4	DN3	DN2	DN1	DN0	XXnF FD23H	00H
	R	R	R	R	R	R	R	R	n = 3, 7, B	

This register is read only and it is cleared when the INIT bit in CANCn register is set.

DNn	Data New Bit for Message n (n = 0...7)
0	No message received on message n, CPU has cleared DN bit in message n
1	Data received in message n that was not acknowledged by the CPU

DN0 bit has no meaning when receive buffer 0 is configured for mask operation in the mask control register.
DN2 bit has no meaning when receive buffer 2 is configured for mask operation in the mask control register.

10.14.3 Mask Control

The mask control register defines whether the DCAN compares all identifier bits or if some bits are not used for comparison.
This functionality is provided by the use of the mask information. The mask information defines for each bit of the identifier whether it is used for comparison or not.

The DCAN uses a receive buffer for this information, when it is enabled by this register. In this case, this buffer is not used for normal message storage. Unused bytes can be used for application needs.

(1) Mask Control Register

This register controls the mask function of the receive messages of the DCAN-module.

MASKCn can be written with an 8-bit memory manipulation instruction.

Figure 10-31: Mask Control Register (MASKCn) (n = 0, 1)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset
MASKC0	0	0	SSHT	AL	0	GLOBAL	MSK1	MSK0	XXnF FF2CH	00H
	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	n = 3, 7, B	
MASKC1	0	0	SSHT	AL	0	GLOBAL	MSK1	MSK0	XXnF FD2CH	00H

MSK0	Mask 0 Enable
0	Receive buffer 0 and 1 in normal operation
1	Receive buffer 0 is mask for buffer 1

MSK1	Mask 1 Enable
0	Receive buffer 2 and 3 in normal operation
1	Receive buffer 2 is mask for buffer 3

GLOBAL	Enable Global Mask
0	Normal operation
1	Highest defined mask is active for all following buffers

SSHT	AL	Function
0	x	Single shot mode disabled
1	0	Single shot mode enabled; no re-transmission when an error occurs. Transmit message will not be queued for a second transmit request when the arbitration was lost
1	1	Single shot mode enabled; no re-transmission when an error occurs. Transmit message will be queued for a second transmit request when the arbitration was lost.

Caution: This register is readable at any time. Writing to the MASKCn register is only allowed during initialization mode. Any write to this register when INITn is set and the initialization mode is not confirmed by the INITSTATE bit can have unexpected behavior to the CAN bus.

The following table shows which compare takes place for the different receive buffers. The ID in this table always represents the ID stored in the mentioned receive buffer. The table also shows which buffers are used as mask information and do not receive messages. A global mask can be used for standard and extended frames at the same time. The frame type is only controlled by the IDE bit of the receiving buffer.

Table 10-3: Mask Operation Buffers

GLOBAL	MSK1	MSK0	Receive Buffer					Operation
			0	1	2	3	4-15	
X	0	0	Compare ID	Compare ID	Compare ID	Compare ID	Compare ID	Normal
0	0	1	Mask0	Compare ID & mask0	Compare ID	Compare ID	Compare ID	One mask
0	1	0	Compare ID	Compare ID	Mask1	Compare ID & mask1	Compare ID	One mask
0	1	1	Mask0	Compare ID & mask0	Mask1	Compare ID & mask1	Compare ID	Two masks
1	0	1	Mask0	Compare ID & mask0	Compare ID & mask0	Compare ID	& mask0	Global mask
1	1	0	Compare ID	Compare ID	Mask1	Compare ID	& mask1	Two normal, rest global mask
1	1	1	Mask0	Compare ID & mask0	Mask1	Compare ID	& mask1	One mask, rest global mask

Priority of receive buffers during compare
It is possible that more than one receive buffer is configured to receive an incoming message. In this case it is necessary to priory the receive buffers.

The priority of the 16 receive buffers depends on the setup of the mask register.

The rules for priority are:

- Lower numbered receive buffers have higher priority
- A masked receive buffer has a lower priority than all non-masked receive buffers.

Examples:

1. All RX buffers are enabled to receive the same standard identifier 0x7FFH. Result: the message with identifier 0x7FFh is stored in RX0.

2. In difference to the previous set up, the mask option is set for RX2. Again the message 0x7FFH is stored in buffer in RX0.

3. If additionally RX0 is configured as a mask, the message will be stored in RX4.

10.14.4 Special Functions

(1) Redefinition Control Register

This register controls the redefinition of an identifier of a received buffer.

REDEFn can be written with an 1-bit or an 8-bit memory manipulation instruction.

Figure 10-32: Redefinition Control Register (REDEFn) (n = 0, 1) (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After Reset
REDEF0	DEF	0	0	0	SEL3	SEL2	SEL1	SEL0	XXnF FF24H	00H
	R/W	R	R	R	R/W	R/W	R/W	R/W	n = 3, 7, B	

Symbol	7	6	5	4	3	2	1	0	Address	After Reset
REDEF1	DEF	0	0	0	SEL3	SEL2	SEL1	SEL0	XXnF FD24H	00H
	R/W	R	R	R	R/W	R/W	R/W	R/W	n = 3, 7, B	

The redefinition register provides a way to change identifiers and other control information for one receive buffer, without disturbing the operation of the other buffers.

DEF	Redefine Permission Bit
0	Normal operation
1	Receive operation for selected message is disabled CPU can change definition data for this message

This bit is cleared when INIT bit in CANC is set.

Figure 10-32: Redefinition Control Register (2/2)

SEL3	SEL2	SEL1	SEL0	Buffer selection (n =0...15)
0	0	0	0	Buffer 0 is selected for redefinition
0	0	0	1	Buffer 1 is selected for redefinition
0	0	1	0	Buffer 2 is selected for redefinition
0	0	1	1	Buffer 3 is selected for redefinition
0	1	0	0	Buffer 4 is selected for redefinition
0	1	0	1	Buffer 5 is selected for redefinition
0	1	1	0	Buffer 6 is selected for redefinition
0	1	1	1	Buffer 7 is selected for redefinition
1	0	0	0	Buffer 8 is selected for redefinition
1	0	0	1	Buffer 9 is selected for redefinition
1	0	1	0	Buffer 10 is selected for redefinition
1	0	1	1	Buffer 11 is selected for redefinition
1	1	0	0	Buffer 12 is selected for redefinition
1	1	0	1	Buffer 13 is selected for redefinition
1	1	1	0	Buffer 14 is selected for redefinition
1	1	1	1	Buffer 15 is selected for redefinition
Other than above				Setting prohibited

Cautions: 1. Keep special programming sequence. Falling to do so can cause inconsistent data or loss of receive data:

2. Do not change DEF bit and SELx bit (x = 3 to 0) at the same time. Change SELx bit (x = 3 to 0) only when DEF bit is cleared.

3. Write first SELxn (x = 3 to 0) with DEFn cleared. Write than SELxn (x = 3 to 0) with DEF bit, or use bit manipulation instruction. Only clear DEF bit by keeping SELx bit (x = 3 to 0) or use bit manipulation instruction.

Setting the redefinition bit removes the selected receive buffer from the list of possible ID hits during identifier comparisons.

Setting the DEF bit will not have immediate effect, if DCAN is preparing to store or is already in progress of storing a received message into the particular buffer. In this case the redefinition request is ignored for the currently processed message.

The application should monitor the DN flag before requesting the redefinition state for a particular buffer. A DN flag set indicates a new message that arrived or a new message that is in progress of being stored to that buffer. The application should be prepared to receive a message immediately after redefinition state was set. The user can identify this situation because the data new bit (DN) in the receive buffer will be set. This is of special importance if it is used together with a mask function because in this case the DCAN also writes the identifier part of the message to the receive buffer. Then the application needs to re-write the configuration of the message buffer.

10.14.5 Performance of the DCAN

(1) Data wait control register (DWC)

For the access to the DCAN SFRs, up to 3 data wait states can be inserted in a bus cycle that starts every two memory blocks.
The number of wait states can be programmed by using data wait control register (DWC). Immediately after the system has been reset, three data wait insertion states are automatically programmed for all memory blocks.

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

Figure 10-33: Data Wait Control Register (DWC)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After Reset
DWC	DW71	DW70	DW61	DW60	DW51	DW50	DW41	DW40	DW31	DW30	DW21	DW20	DW11	DW10	DW01	DW00	XXFF F060H	FFH

DWn1	DWn0	Number of Wait States to Be Inserted
0	0	0
0	1	1
1	0	2
1	1	3 (RESET Value)

Remark: n = 0 to 7

n	Blocks into Which Wait States Are Inserted
0	Blocks 0/1 (XX00 0000H to XX1F FFFFH)
1	Blocks 2/3 (XX20 0000H to XX3F FFFFH)
2	Blocks 4/5 (XX40 0000H to XX5F FFFFH)
3	Blocks 6/7 (XX60 0000H to XX7F FFFFH)
4	Blocks 8/9 (XX80 0000H to XX9F FFFFH)
5	Blocks 10/11 (XXA0 0000H to XXBF FFFFH)
6	Blocks 12/13 (XXC0 0000H to XXDF FFFFH)
7	Blocks 14/15 (XXE0 0000H to XXFF FFFFH)

Remark: X = don’t care

Depending of the selected base address of the DCAN, the DCAN SFRs of the DCAN are in block 3, 7 or 11. To insure the best DCAN-performance the DWC register has to be set to 00H. Therefore no wait states are used for the access of the DCAN SFRs.

(Block 0 is reserved for the internal ROM area. It is not subject to programmable wait control, regardless of the setting of DWC, and is always accessed without wait states.
The internal RAM area of block 15 is not subject to programmable wait control and is always accessed without wait states. The on-chip peripheral I/O area of this block is not subject to programmable wait control, either. The wait control is dependent upon the execution of each peripheral function.)

10.15 Interrupt Information

10.15.1 Interrupt Vectors

The DCAN peripheral supports four interrupt sources as shown in the following table.

Table 10-4: Interrupt Sources

Function	Source	Interrupt Flag	
		Channel 0	Channel 1
Error	Error counter Overrun error Wake up	CEIC0	CEIC1
Receive	Received frame is valid	CRIC0	CRIC1
Transmit buffer 0	TXRQ0 is cleared	CTIC00	CTIC10
Transmit buffer 1	TXRQ1 is cleared	CTIC01	CTIC11

10.15.2 Transmit Interrupt

The transmit interrupt is generated when all following conditions are fulfilled:

- The transmit interrupt 0 is generated when TXRQ0 bit is cleared.
- The transmit interrupt 1 is generated when TXRQ1 bit is cleared.

Clearing of these bits releases the buffer for writing a new message into it. This event can occur due to a successful transmission or due to an abort of a transmission. Only the DCAN can clear this bit. The CPU can only request to clear the TXRQx bit by setting the TXAx bit (x = 0, 1).

10.15.3 Receive Interrupt

The receive interrupt is generated when all following conditions are fulfilled:

- CAN protocol part marks received frame valid
- Memory access engine finds a message buffer with an identifier/mask combination that fits to the received frame
- The memory access engine successfully stored data in the message buffer
- The message buffer is marked for interrupt generation with ENI bit set

The memory access engine can delay the interrupt up to the 7th bit of the next frame because of its compare and store operations.

10.15.4 Error Interrupt

The error interrupt is generated when all following conditions are fulfilled:

- Transmission error counter (BOFF bit of CANESn register) changes its state.
- Transmission error counter status (TECS bit of CANESn register) changes its state.
- Reception error status (RECS bit of CANESn register) changes its state.
- Overrun during RAM access (OVER bit of CANESn register) becomes active.
- The wake-up condition (WAKE bit of CANESn register) becomes active.

The internal wake-up conditions sets the interrupt high. The interrupt is kept high until the CPU cleared the WAKE bit. No further interrupt can be detected by the CPU as long as the WAKE bit is set.

10.16 Power Saving Modes

10.16.1 CPU Halt Mode

The CPU halt mode is possible in conjunction with DCAN Sleep mode.

10.16.2 CPU WATCH Mode

The CPU watch mode is possible in conjunction with DCAN Sleep mode. Not all host CPU cores feature a watch mode.

10.16.3 CPU Stop Mode

The DCAN stops any activity when its clock supply stops due to a CPU Stop mode issued. This may cause an erroneous behaviour on the CAN bus. Entering the CPU Stop Mode is not allowed when the DCAN is in normal mode, i.e. online to the CAN bus.

The DCAN will reach an overrun condition, when it receives clock supply again.

CPU Stop mode is possible when the DCAN was set to initialization state, sleep mode or stop mode beforehand. Note that the CPU will not be started again if the DCAN Stop mode was entered previously. The DCAN has to be set to initialization state or stop mode when the host CPU operates on subclock (i.e. feature of some 78K0 products).

10.16.4 DCAN Sleep Mode

The DCAN Sleep mode is intended to lower the power consumption during phases where no communication is required.
The CPU requests the DCAN Sleep mode. The DCAN will signal with the WAKE bit, if the request was granted or if it is not possible to enter the sleep mode due to ongoing bus activities.
After a successful switch to the DCAN Sleep mode, the CPU can safely go into halt, watch or stop mode. However, the application needs to be prepared that the DCAN cancels the sleep mode any time due to bus activities. If the wake-up interrupt is serviced, the CPU Stop mode has not to be issued. Otherwise the CPU will not be released from CPU Stop mode even when there is ongoing bus activity.
The wake-up is independent from the clock. The release time for the CPU Stop mode of the device is of no concern because the DCAN synchronizes again to the CAN bus after clock supply has started.

The following example sketches the general approach on how to enter the DCAN Sleep mode. Note that the function may not return for infinite time when the CAN bus is busy. The user may apply time out controls to avoid excessive run-times.

Code example:

```
DCAN_Sleep_Mode(void){
CANES = 0x02;                  // clear Wake bit
CANC = 0x04                    // request DCAN Sleep mode
while (CANES & 0x02)           // check if DCAN Sleep mode was accepted
       {
       CANES = 0x02;           // try again to get DCAN asleep
       CANC = 0x04;
       }
}
```

The following code example assures a safe transition into CPU Stop mode for all timing scenarios of a suddenly occurring bus activity. The code prevents that the CPU gets stuck with its oscillator stopped despite CAN bus activity.

Code example:

```
........                       //any application code

DCAN_Sleep_Mode;               //request and enter DCAN sleep mode

........                       //any application code

DI();                          //disable interrupts
NOP;Note
NOP;
if (wakeup_interrupt_occurred == FALSE)
                               // the variable wakeup_interrupt occurred
                               // needs to be initialized at system reset
                               // and it needs to be set TRUE when servicing
                               // the wake-up interrupt.
       {
       CPU_STOP;               //enter CPU Stop mode
       }
NOP;Note
NOP:
NOP;
EI();                          // enable interrupts
.........                      // resume with application code
```

Note: The interrupt acknowledge needs some clock cycles (depends on host core). In order to prevent that the variable wakeup_interrupt_occurred is already read before DI(); becomes effective some NOP-instruction have to be inserted. As well the number of NOP-instructions after the CPU Stop instruction is dependent on the host core. The given example is tailored for 78K0.

10.17 DCAN Stop Mode

The CPU requests this mode from DCAN. The procedure equals the request for DCAN Sleep mode. The DCAN will signal with the WAKE bit, if the request was granted or if it is not possible to enter the DCAN Stop mode due to ongoing bus activities.

After a successful switch to the DCAN Stop mode, the CPU can safely go into halt, watch or stop mode without any precautions. The DCAN can only be woken up by the CPU. Therefore the CPU needs to clear the SLEEP bit in the CANCn register.

This mode reduces the power consumption of the DCAN to a minimum.

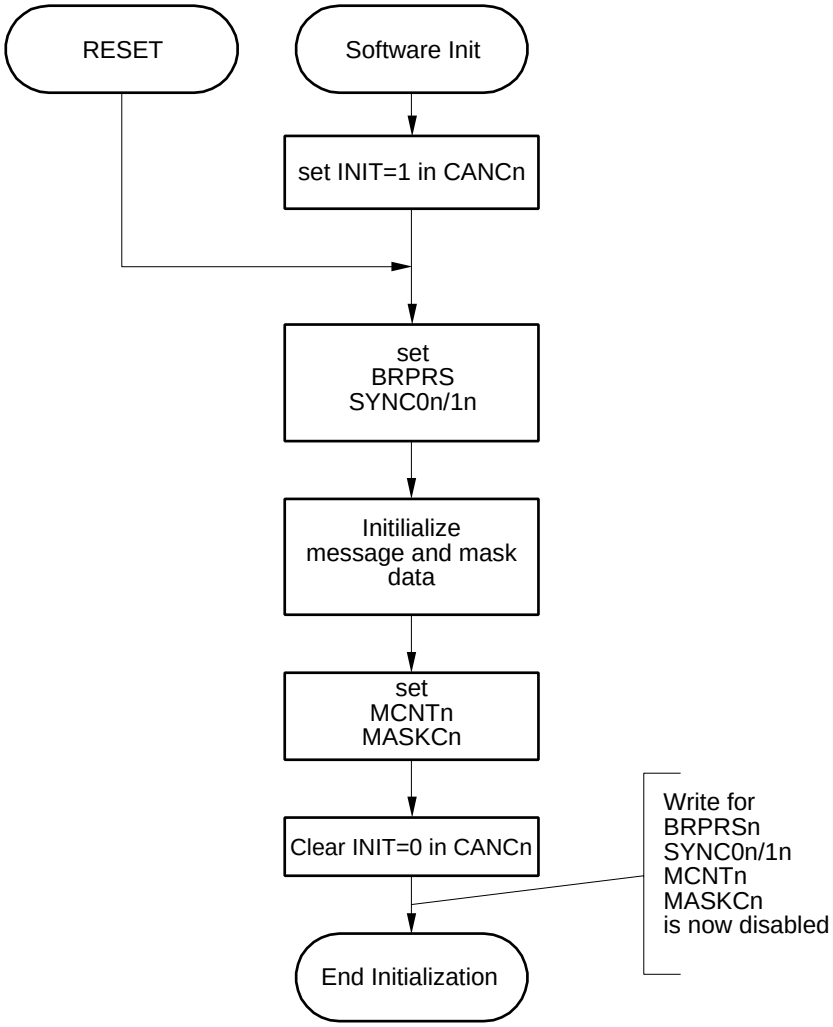
Code example:

```
DCAN_Stop_Mode(void){
CANES = 0x02;                    // clear Wake bit
CANC = 0x06                      // request DCAN Stop mode
while (CANES & 0x02)             // check if DCAN Stop mode was accepted
        {
        CANES = 0x02;            // try again to get DCAN into stop mode
        CANC = 0x06;
        }
}
```

10.18 Functional Description by Flowcharts

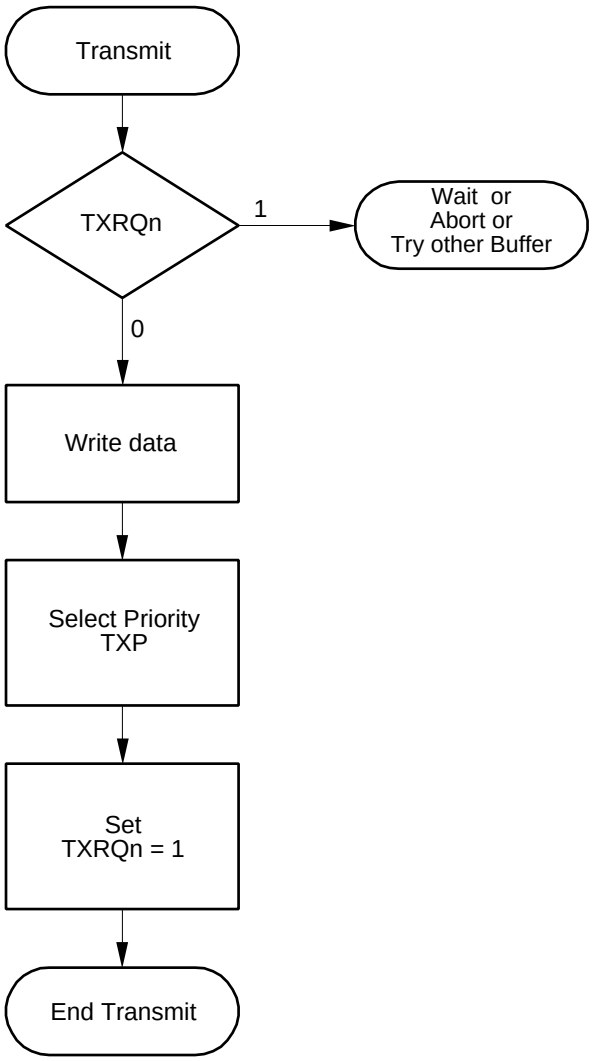
10.18.1 Initialization

Figure 10-34: Initialization Flow Chart



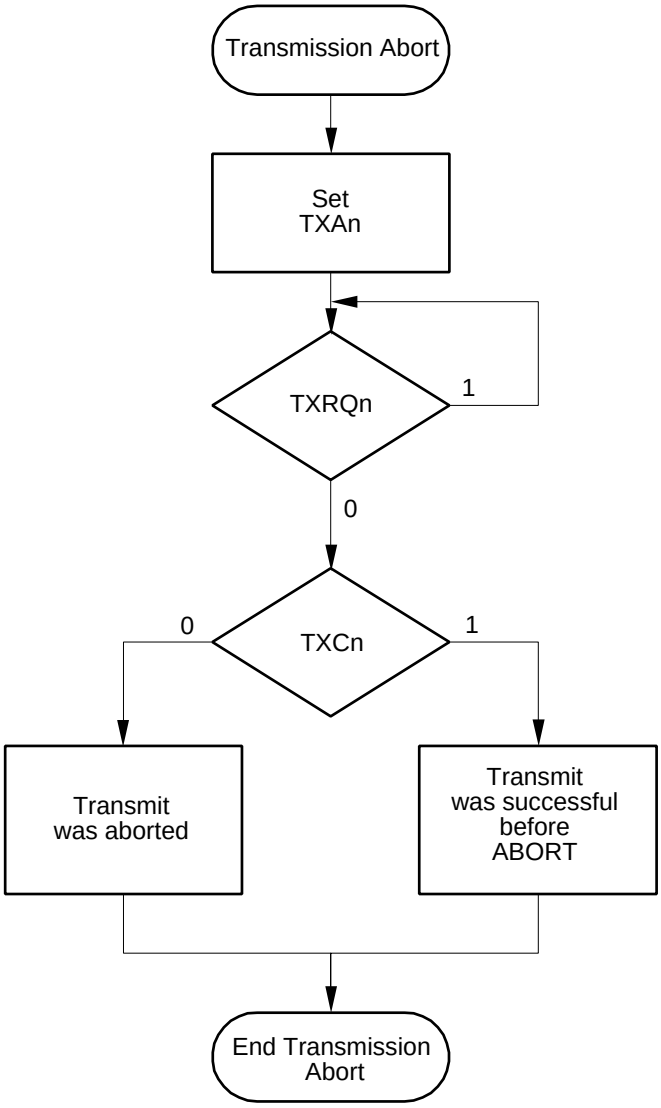
10.18.2 Transmit Preparation

Figure 10-35: Transmit Preparation



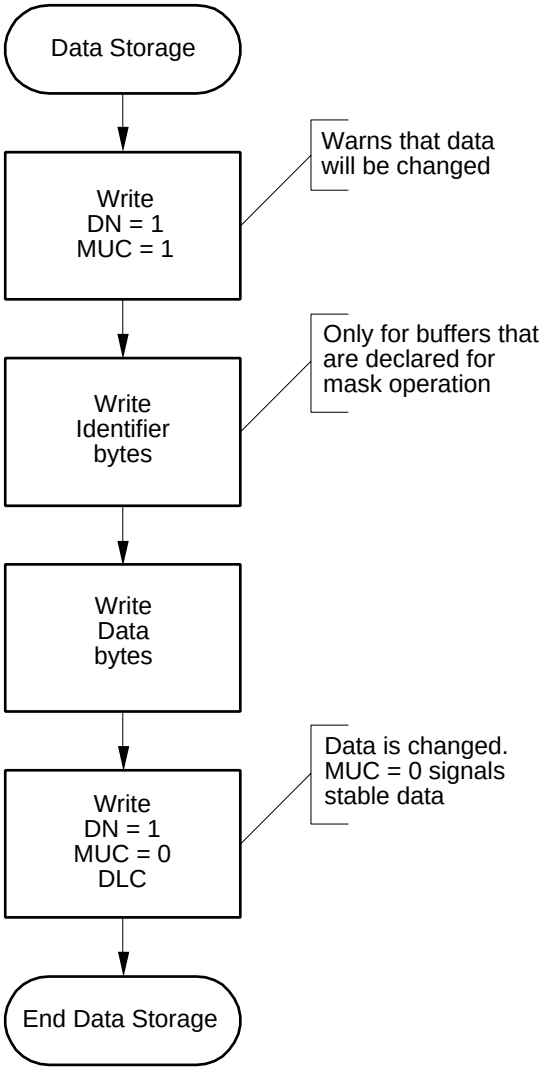
10.18.3 Abort Transmit

Figure 10-36: Transmit Abort



10.18.4 Handling by the DCAN

Figure 10-37: Handling of Semaphore Bits by DCAN-Module



10.18.5 Data New Flag behavior

The DCAN implementations have a special behavior on the function of the DN flag. The CPU may inadvertently reset the MUC bit when resetting the DN flag (Clear DN bit).

A particular waiting time (Wait loop) needs to be inserted before checking if consistent data was read beforehand.

The waiting time depends on macro frequency, CPU frequency and the number of data bytes read. Refer to the application report EACT-BR-5001-1.0 for details on the exact waiting time.

Figure 10-38: Receive with Interrupt in case of Data New Flag behavior

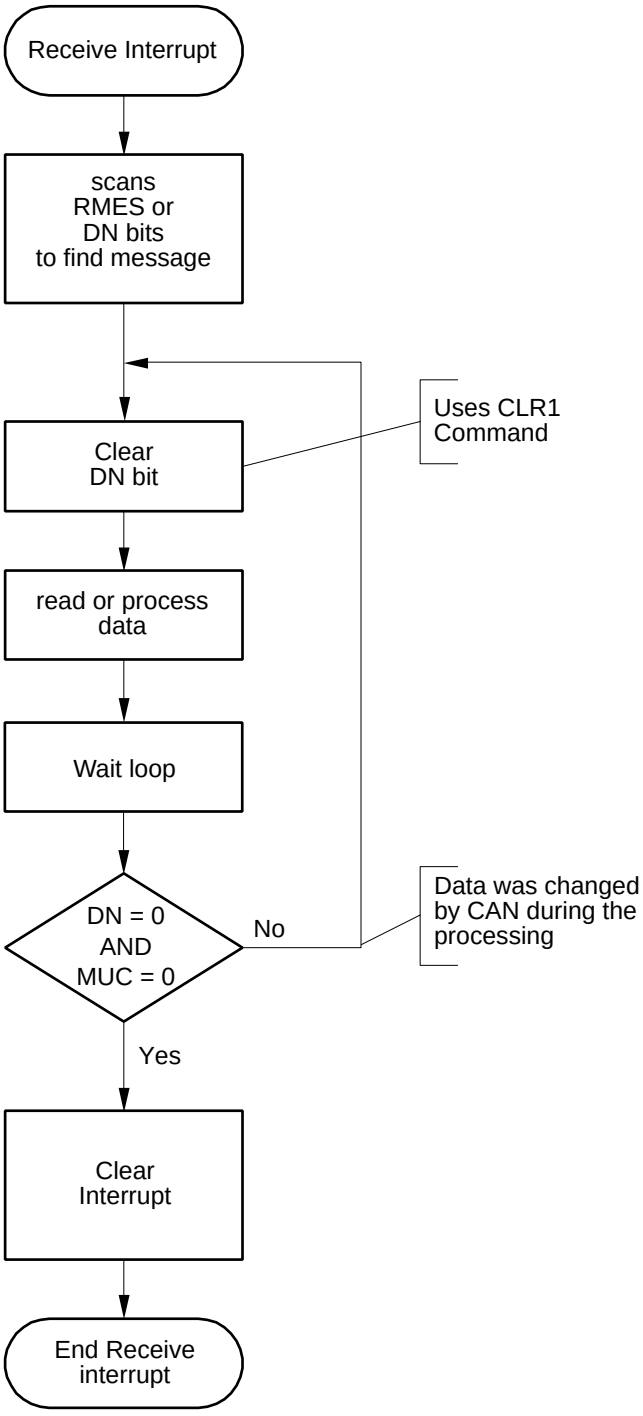
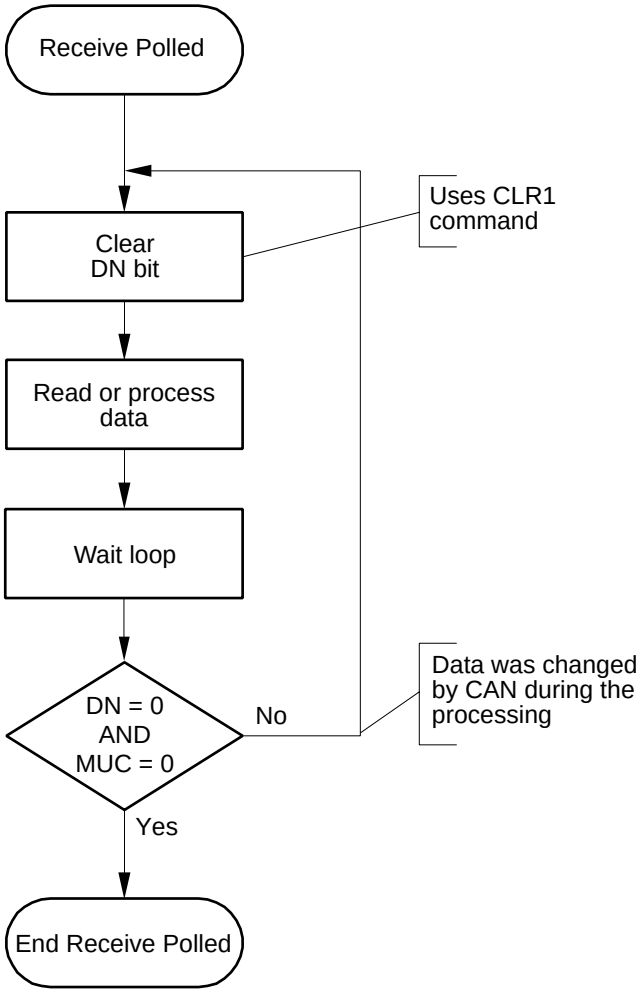


Figure 10-39: Receive, Software Polling in case of Data New Flag behavior



Chapter 11 A/D Converter

11.1 A/D Converter Functions

The A/D converter is an 10-bit resolution converter that converts analog inputs into digital values. It can control up to 8 analog input channels (ANI0 to ANI7).
This A/D converter has the following functions:

(1) A/D conversion with 10-bit resolution

One channel of analog input is selected from ANI0 to ANI7, and A/D conversion is repeatedly executed with a resolution of 10 bits. Each time the conversion has been completed, an interrupt request (INTAD) is generated.

(2) Power-fail detection function

This function is to detect a voltage drop in the battery of an automobile. The result of A/D conversion (value of the ADCR1 register) and the value of PFT register (PFT: power-fail compare threshold value register) are compared. If the condition for comparison is satisfied, the INTAD is generated.

Figure 11-1: A/D Converter Block Diagram

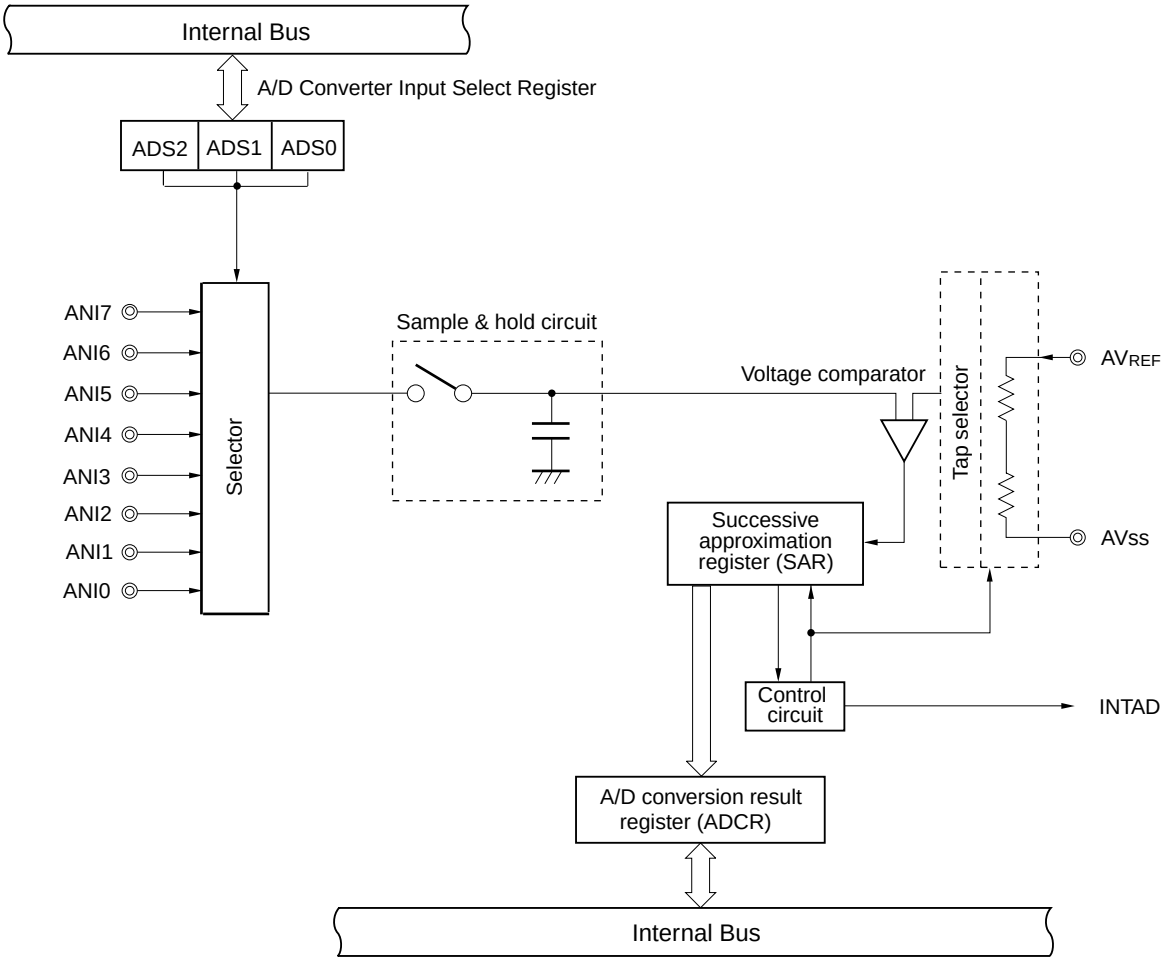
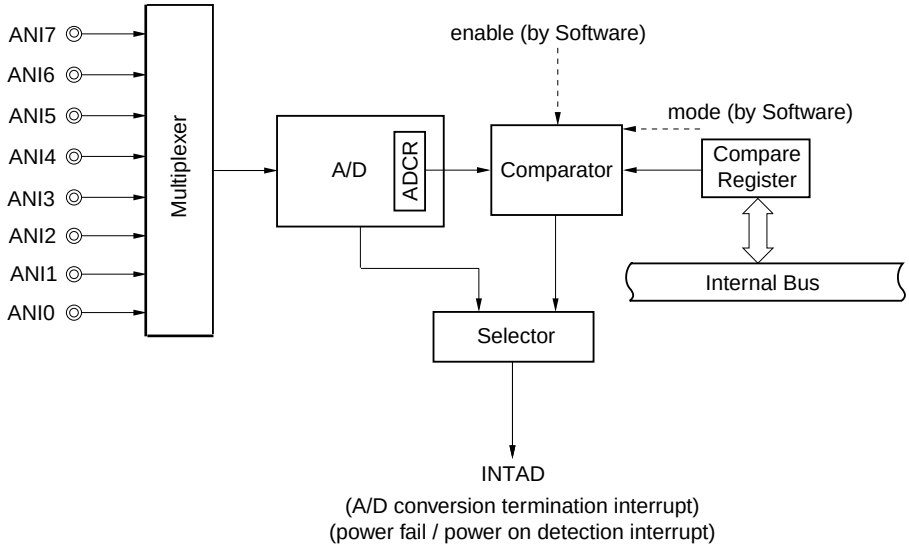


Figure 11-2: Power-Fail Detection Function Block Diagram



11.2 A/D Converter Configuration

A/D converter consists of the following hardware.

Table 11-1: A/D Converter Configuration

Item	Configuration
Analog input	8 channels (ANI0 to ANI7)
Register	Successive approximation register (SAR) A/D converter conversion result register (ADCR)
Control register	A/D converter mode register (ADM) Analog input channel specification register (ADS) Power-fail compare mode register (PFM) Power fail threshold value register (PFT)

(1) Successive approximation register (SAR)

This register compares the analog input voltage value to the voltage tap (compare voltage) value applied from the series resistor string, and holds the result from the most significant bit (MSB). When up to the least significant bit (LSB) is set (end of A/D conversion), the SAR contents are transferred to the A/D conversion result register.

(2) A/D conversion result register (ADCR)

This register holds the A/D conversion result. Each time when the A/D conversion ends, the conversion result is loaded from the successive approximation register.

ADCR is read with an 10-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ADCR to 00H.

Caution: If a write operation is executed to the A/D converter mode register (ADM) and the analog input channel specification register (ADS) the contents of ADCR are undefined. Read the conversion result before a write operation is executed to ADM and ADS. If a timing other than the above is used, the correct conversion result may not be read.

(3) Sample & hold circuit

The sample & hold circuit samples each analog input sequential applied from the input circuit, and sends it to the voltage comparator. This circuit holds the sampled analog input voltage value during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the analog input to the series resistor string output voltage.

(5) Series resistor string

The series resistor string is in AV_{DD} to AV_{SS}, and generates a voltage to be compared to the analog input.

(6) ANI0 to ANI7 pins

These are eight analog input pins to input analog signals to the A/D converter. ANI0 to ANI7 are alternate-function pins that can also be used for digital input.

Caution: Use ANI0 to ANI7 input voltages within the specification range. If a voltage higher than AV_{DD} or lower than AV_{SS} is applied (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

(7) AV_{DD} pin

This pin is used as the power supply pin of the A/D converter. The supply power has to be connected when the A/D converter is used.

(8) AV_{REF} pin

This pin inputs the A/D converter reference voltage of the A/D converter.
It converts signals input to ANI0 to ANI7 into digital signals according to the voltage applied between AV_{REF} and AV_{SS}.
The current flowing into series resistor string can be reduced by setting the A/D converter in the standby mode via setting the ADCS bit of the ADM register to “0”. A transistor will disconnect the AV_{REF} from the resistor string.

(9) AV_{SS} pin

This is the GND potential pin of the A/D converter. Always keep it at the same potential as the V_{SS} pin even when not using the A/D converter.

11.3 A/D Converter Control Registers

The following 4 types of registers are used to control A/D converter:

- A/D converter mode register (ADM)
- Analog input channel specification register (ADS)
- Power-fail compare mode register (PFM)
- Power-fail compare threshold value register (PFT)

11.3.1 Register format of A/D Converter Control Register

Table 11-2: Register format of A/D Converter Control Register

SFR name	Symbol	R/W	Manipulable Bit Unit			After RESET
			1-bit	8-bit	16-bit	
A/D converter mode register	ADM	R/W	×	×	---	00H
Analog input channel specification register	ADS	R/W	---	×	---	00H
Power-fail compare mode register	PFM	R/W	×	×	---	00H
Power-fail compare threshold value register	PFT	R/W	---	×	---	00H
A/D conversion result register	ADCR	R	---	---	×	00H

(1) A/D converter mode register (ADM)

This register sets the conversion time for analog input to be A/D converted and conversion start/stop.

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

Figure 11-3: A/D Converter Mode Register (ADM) Format

Symbol	7	6	5	4	3	2	1	0	Address	Initial value
ADM	ADCS	0	FR3	FR2	FR1	FR0	0	0	FFFF F370H	00H

Caution: Bits 0, 1 and 6 must be set to 0.

ADCS	A/D Conversion Operation Control
0	Stop conversion operation **Note**
1	Enable conversion operation

Note: While the ADCS bit is 0, the series resistor string is disabled, in order to reduce the power consumption.

FR3	FR2	FR1	FR0	Conversion clock	Conversion time (μs)
0	1	0	1	$f_{XX} \times 84$	5.25
0	1	1	0	$f_{XX} \times 96$	6.00
0	1	1	1	$f_{XX} \times 108$	6.75
1	0	0	0	$f_{XX} \times 120$	7.50
1	0	0	1	$f_{XX} \times 144$	9.00
1	0	1	0	$f_{XX} \times 168$	10.5
1	0	1	1	$f_{XX} \times 192$	12.0
Other than above				Setting Prohibited	

Remark: f_{XX} : Main system clock oscillation frequency (16 MHz).

(2) Analog input channel specification register (ADS)

This register specifies the analog voltage input port for A/D conversion.

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

Figure 11-4: Analog Input Channel Specification Register (ADS) Format

Symbol	7	6	5	4	3	2	1	0	Address	Initial value
ADS	0	0	0	0	0	ADS2	ADS1	ADS0	FFFF F372H	00H

ADS2	ADS1	ADS0	Analog Input Channel Specification
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

(3) Power-fail compare mode register (PFM)

The power-fail compare mode register (PFM) controls a comparison operation.

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

Figure 11-5: Power-Fail Compare Mode Register (PFM) Format

Symbol	7	6	5	4	3	2	1	0	Address	Initial value
PFM	PFEN	PFCM	0	0	0	0	0	0	FFFF F374H	00H

Caution: Bits 0 to 5 must be set to 0.

PFEN	Enables Power-Fail Comparison
0	Disables power-fail comparison (used as normal A/D converter)
1	Enables power-fail comparison (used to detect power failure)

PFCM	Power-Fail Compare Mode Selection
0	ADCR ≥ PFT Generates interrupt request signal INTAD
1	ADCR < PFT Generates interrupt request signal INTAD

(4) Power-fail compare threshold value register (PFT)

The power-fail compare threshold value register (PFT) sets a threshold value against which the result of A/D conversion is to be compared.
This 8-bit value will be compared with ADCR (Only high-order 8-bits in 10-bit Comparison result).

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

Figure 11-6: Power-fail compare threshold value register (PFT)

Symbol	7	6	5	4	3	2	1	0	Address	Initial value
PFT	PFT7	PFT6	PFT5	PFT4	PFT3	PFT2	PFT1	PFT0	FFFF F376H	00H

(5) A/D conversion result register (ADCR)

The ADCR register is a 10-bit registers that hold the results of A/D conversion.
When reading 16 bits of data of an A/D conversion result from an ADCR register, only the upper 10 bits are valid and the lower 6 bits are always read as 0.

This register can be read in 16 bit units.

Figure 11-7: A/D Conversion Result Registers (ADCR)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Initial value
ADCR	ADCR9	ADCR8	ADCR7	ADCR6	ADCR5	ADCR4	ADCR3	ADCR2	ADCR1	ADCR0	0	0	0	0	0	0	FFFF F378H	unde-fined

Caution: If a write operation is performed to a SFR of the A/D before the ADCR register is read, the content of the ADCR register is not guaranteed.
Therefore, no other SFR of the A/D should be written before the ADCR register is read after conversion is completed.

11.4 A/D Converter Operations

11.4.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion with the analog input channel specification register (ADS).

<2> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<3> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation is ended.

<4> Set bit 7 of the successive approximation register (SAR) so that the tap selector sets the series resistor string voltage tap to (1/2) AV_{DD}.

<5> The voltage difference between the series resistor string voltage tap and analog input is compared with the voltage comparator. If the analog input is greater than (1/2) AV_{DD}, the MSB of SAR remains set. If the analog input is smaller than (1/2) AV_{DD}, the MSB is reset.

<6> Next, bit 6 of SAR is automatically set, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 7, as described below.

- Bit 7 = 1: (3/4) AV_{DD}
- Bit 7 = 0: (1/4) AV_{DD}

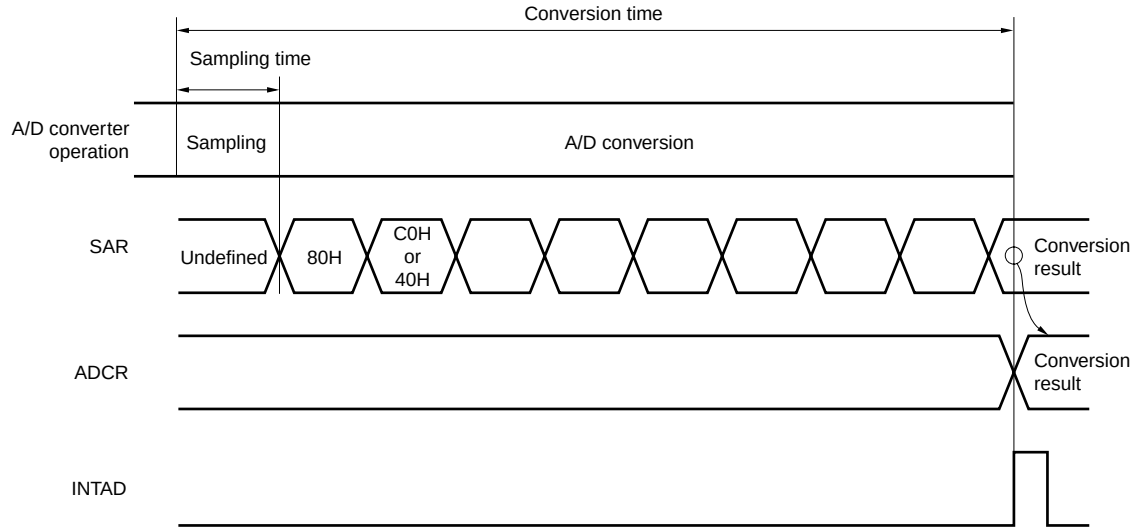
The voltage tap and analog input voltage are compared and bit 6 of SAR is manipulated as follows.

- Analog input voltage ≥ Voltage tap: Bit 6 = 1
- Analog input voltage < Voltage tap: Bit 6 = 0

<7> Comparison is continued in this way up to bit 0 of SAR.

<8> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to and latched in the A/D conversion result register (ADCR).
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

Figure 11-8: Basic Operation of 10-bit A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register (ADM) is reset (to 0) by software.
If a write operation to the ADM and analog input channel specification register (ADS) is performed during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (to 1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input sets the A/D conversion result register (ADCR) to 00H.

11.4.2 Input voltage and conversion results

The relation between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the A/D conversion result (stored in the A/D conversion result register (ADCR)) is shown by the following expression:

$$ADCR = INT\left[\left(\frac{(V_{IN} \times 1024)}{AV_{REF}} \pm 0{,}5 \right) \right] \times 64$$

or the following expression:

$$\frac{(ADCR/64 - 0.5) \times AV_{REF}}{1024} < V_{IN} < \frac{(ADCR/64 + 0.5) \times AV_{REF}}{1024}$$

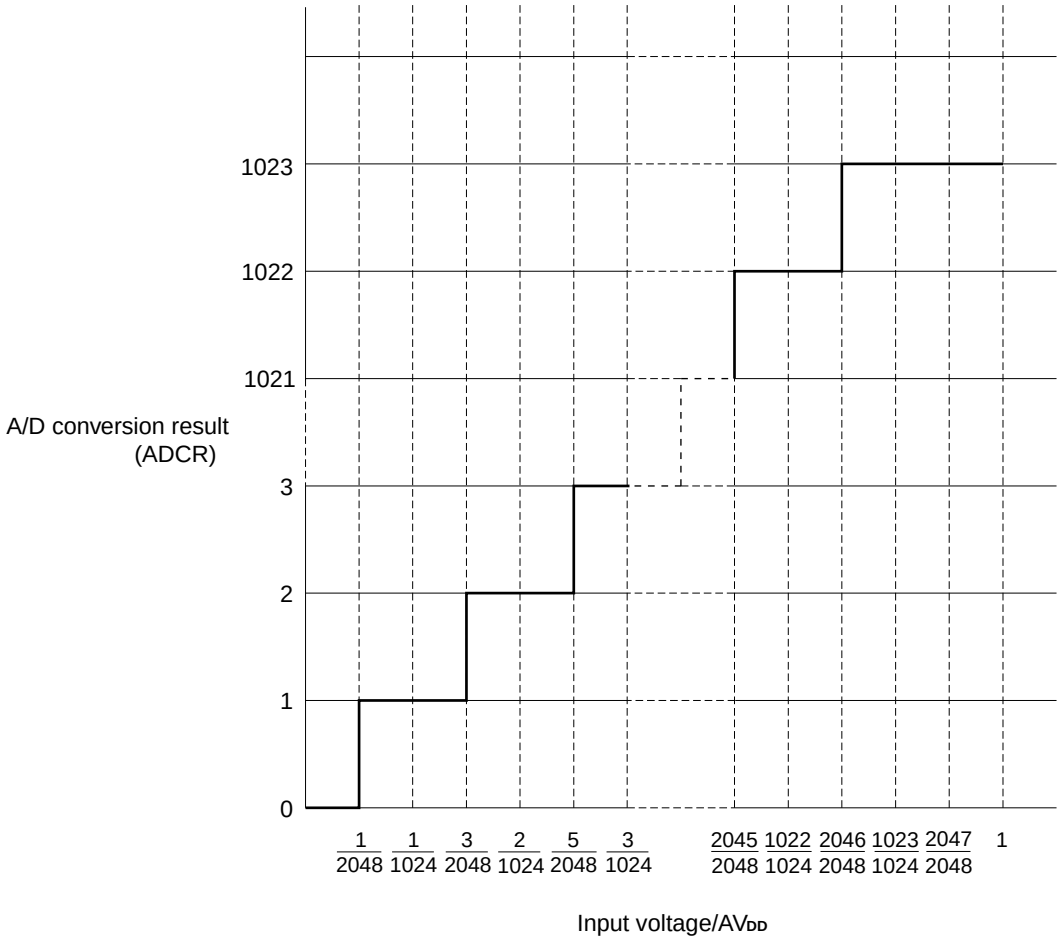
Remark: The factor 64 is used to shift the 10 higher AD-bits to the right calculation position. It shifts this bits for 6 digits to right.

where,

INT():: Function which returns integer part of value in parentheses
V_{IN}: Analog input voltage
AV_{DD}:: AV_{DD} pin voltage and A/D converter power supply
ADCR:: A/D conversion result register (ADCR) value

Figure 11-9, “Relation between Analog Input Voltage and A/D Conversion Result,” on page 389 shows the relation between the analog input voltage and the A/D conversion result.

Figure 11-9: Relation between Analog Input Voltage and A/D Conversion Result



11.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One analog input channel is selected from among ANI0 to ANI7 with the analog input channel specification register (ADS) and A/D conversion is performed.
The following two types of functions can be selected by setting the PFEN flag of the PFM register.

(1) Normal 10-bit A/D converter (PFEN = 0)
(2) Power-fail detection function (PFEN = 1)

(1) A/D conversion (when PFEN = 0)

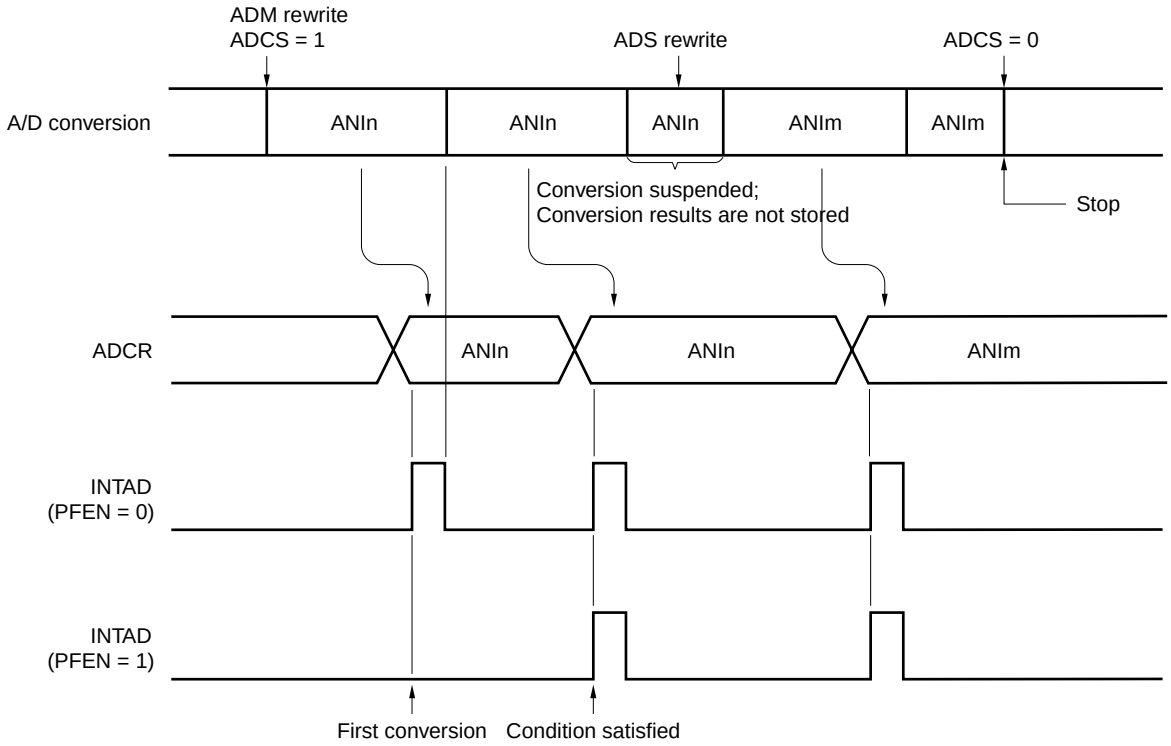
When the ADCS bit of the A/D converter mode register (ADM) is set to “1” and bit 7 of the power-fail compare mode register (PFM) is set to “0”, A/D conversion of the voltage applied to the analog input pin specified with the analog input channel specification register (ADS) starts.
Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR), and the interrupt request signal (INTAD) is generated. After one A/D conversion operation is started and ended, the next conversion operation is immediately started. A/D conversion operations are repeated until new data is written to ADS.
If ADS is rewritten during A/D conversion operation, the A/D conversion operation under execution is stopped, and A/D conversion of a newly selected analog input channel is started.
If data with ADCS set to “0” is written to ADM during A/D conversion operation, the A/D conversion operation stops immediately.

(2) Power-fail detection function (when PFEN = 1)

When ADCS bit of the A/D converter mode register (ADM) and PFEN bit of the power-fail compare mode register (PFM) are set to “1”, A/D conversion of the voltage applied to the analog input pin specified with the analog input channel specification register (ADS) starts.
Upon the end of the A/D conversion, the conversion result is stored in the A/D conversion result register (ADCR), compared with the value of the power-fail compare threshold value register (PFT), and INTAD is generated under the condition specified by the PFCM flag of the PFM register.

Caution: When executing power-fail comparison, the interrupt request signal (INTAD) is not generated on completion of the first conversion after ADCS bit has been set to “1”. INTAD is valid from completion of the second conversion.

Figure 11-10: A/D Conversion



Remarks: 1. n = 0 to 7

2. m = 0 to 7

11.5 A/D Converter Precautions

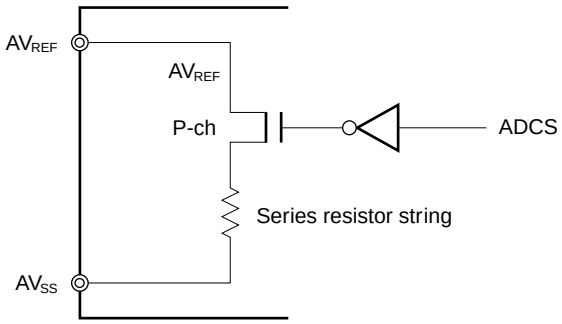
(1) Current consumption in standby mode

A/D converter stops operating in the standby mode. At this time, current consumption is reduced by setting the ADCS bit of the A/D converter mode register (ADM) to “0” to stop conversion.

Caution: This has to be done before an other power save mode of the AVALON (WATCH or STOP mode) is set.

Figure 11-11, “Example Method of Reducing Current Consumption in Standby Mode,” on page 392 shows how to reduce the current consumption in the standby mode.

Figure 11-11: Example Method of Reducing Current Consumption in Standby Mode



(2) Input range of ANI0 to ANI7

The input voltages of ANI0 to ANI7 should be within the specification range. In particular, if a voltage higher than AV_{DD} or lower than AV_{SS} is input (even if within the absolute maximum rating range), the conversion value of that channel will be undefined and the conversion values of other channels may also be affected.

(3) Contending operations

<1> Contention between A/D conversion result register (ADCR) write and ADCR read by instruction upon the end of conversion

ADCR read is given priority. After the read operation, the new conversion result is written to ADCR.

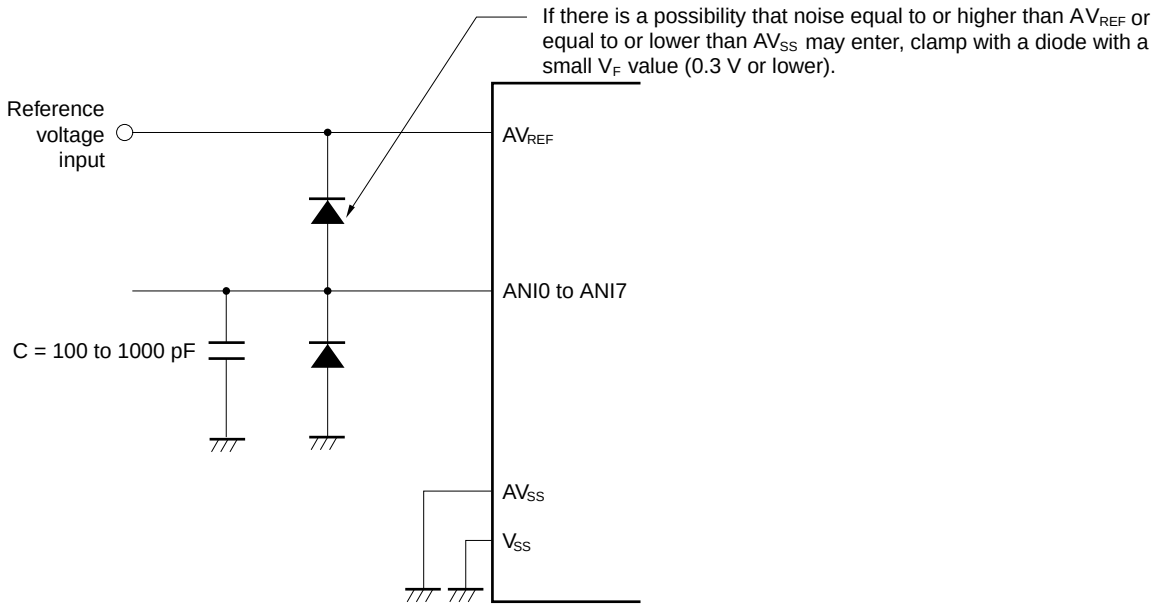
<2> Contention between ADCR write and A/D converter mode register (ADM) write or analog input channel specification register (ADS) write upon the end of conversion

ADM or ADS write is given priority. ADCR write is not performed, nor is the conversion end interrupt request signal (INTAD) generated.

(4) Noise counter measures

To maintain 10-bit resolution, attention must be paid to noise input to pin AV_{DD} and pins ANI0 to ANI7. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally as shown in Figure 11-12, “Analog Input Pin Handling,” on page 393 to reduce noise.

Figure 11-12: Analog Input Pin Handling



(5) ANI0 to ANI7

The analog input pins (ANI0 to ANI7) also function as input port pins (P70 to P77).
When A/D conversion is performed with any of pins ANI0 to ANI7 selected, do not execute a port input instruction while conversion is in progress, as this may reduce the conversion resolution.
Also, if digital pulses are applied to a pin adjacent to the pin in the process of A/D conversion, the expected A/D conversion value may not be obtainable due to coupling noise. Therefore, avoid applying pulses to pins adjacent to the pin undergoing A/D conversion.

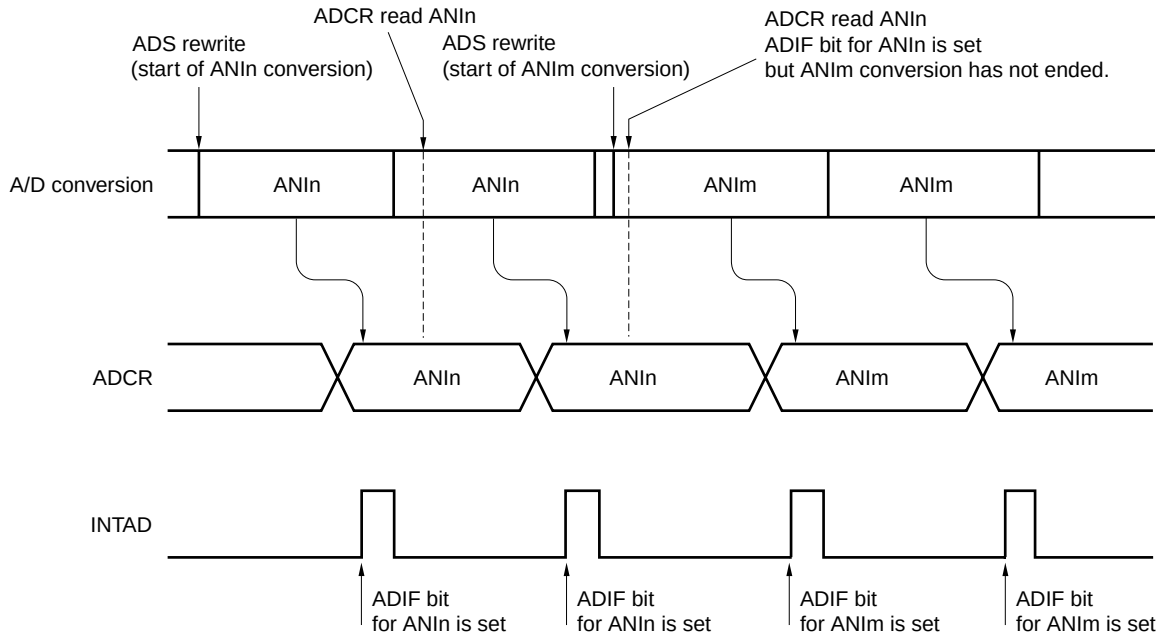
(6) AV_{REF} pin input impedance

A series resistor string is connected between the AV_{REF} pin and the AV_{SS} pin.
If the output impedance of the reference voltage supply is high, this will result in parallel connection to the series resistor string between the AV_{REF} pin and the AV_{SS} pin, and there will be a large reference voltage error. This input impedance is not constant, due to the switching process at the conversion time at this resistor string.
Therefore a low impedance source is require to supply this AV_{REF} pin.

(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF bit) is not cleared even if the analog input channel specification register (ADS) is changed.
Caution is therefore required if a change of analog input pin is performed during A/D conversion. The A/D conversion result and conversion end interrupt request flag for the pre-change analog input may be set just before the ADS rewrite, if the ADIF is read immediately after the ADS rewrite, the ADIF may be set despite to the fact that the A/D conversion for the post-change analog input has not ended.
When the A/D conversion is stopped and then resumed, clear ADIF before the A/D conversion operation is resumed.

Figure 11-13: A/D Conversion End Interrupt Request Generation Timing



Remarks: 1. n = 0 to 7

2. m = 0 to 7

(8) Read of A/D conversion result register (ADCR)

When a write operation is executed to A/D converter mode register (ADM) and analog input channel specification register (ADS), the contents of ADCR are undefined. Read the conversion result before write operation is executed to ADM or ADS register. If a timing other than the above is used, the correct conversion result may not be read.

Chapter 12 Meter Controller/Driver

12.1 Meter Controller/Driver Functions

The meter controller/driver is a function to drive a stepping motor for external meter control or cross coil:

- the pulse width can be set with 8 bits precision
- the pulse width can be set with a precision of 8 + 1 bits with 1-bit addition function
- it can drive up to six 360° type meters

Figure 12-1, “Meter Controller/Driver 0 Block Diagram,” on page 395 and Figure 12-2, “Meter Controller/Driver 1 Block Diagram,” on page 396 show the block diagram of the meter controller/driver.

Figure 12-3, “1-bit Addition Circuit Block Diagram,” on page 396 shows 1-bit addition circuit block diagram.

Figure 12-1: Meter Controller/Driver 0 Block Diagram

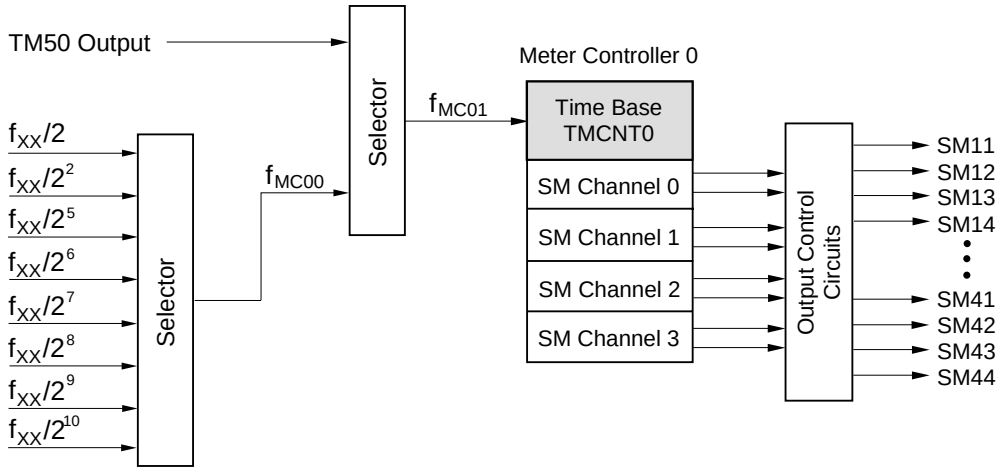
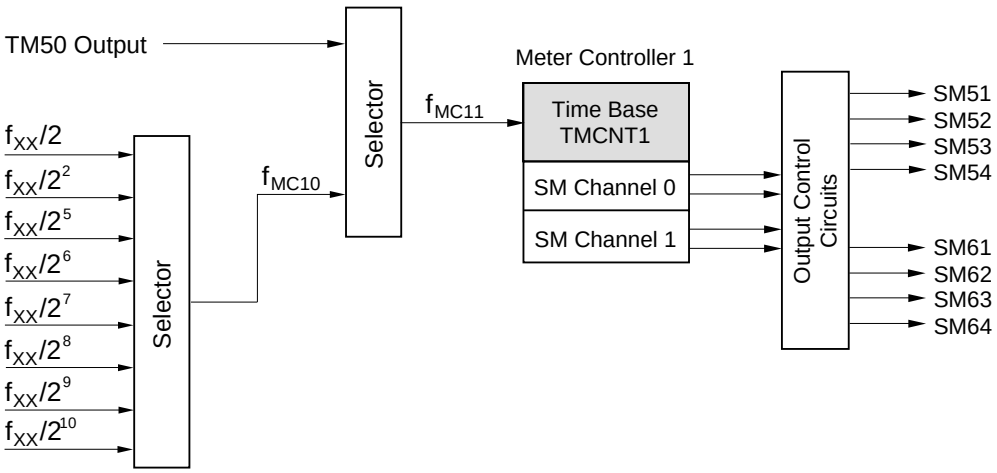
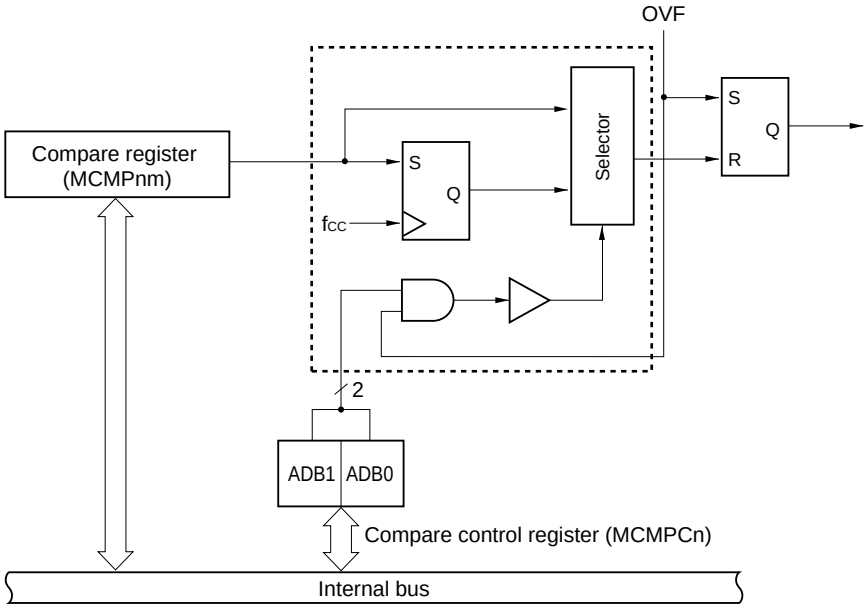


Figure 12-2: Meter Controller/Driver 1 Block Diagram



Remark: n = 1 to 4

Figure 12-3: 1-bit Addition Circuit Block Diagram



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

12.2 Meter Controller/Driver Configuration

The meter controller/driver consists of the following hardware.

Item	Configuration
Timer	Free-running up counter (TMCNT0, TMCNT1): 2 channels
Register	Compare register (MCMPn1, MCMPn0): 12 channels
Control Registers	Timer mode control register (MCNTC0, MCNTC1) Compare control register n (MCMPCn)
Pulse control circuit	1-bit addition circuit/output control circuit

Remark: n = 1 to 6

(1) Free running up counter (TMCNTn, n = 0, 1)

TMCNTn is a 8-bit free running counter. It is also a register that executes an increment at the rising edge of the input clock (f_{MCn1}).

A PWM pulse with a resolution of 8 bits can be created. The duty factor can be set in a range of 0 to 100%.

The count value is cleared in the following cases.

- When $\overline{\text{RESET}}$ signal input
- When counter stops (PCE = 0)

(2) Compare register for sin side (MCMPn0, n = 1 to 6)

MCMPn0 is an 8-bit register that can rewrite a complete value according to the specification by bit 4 (TENn) of the compare control register n (MCMPCn).

The values of these registers are cleared to 00H at $\overline{\text{RESET}}$. The hardware is cleared to 0 by $\overline{\text{RESET}}$.

MCMPn0 is a register that supports read/write only for 8-bit access instructions.

MCMPn0 continuously compares its value with the TMCNT0, TMCNT1 value. When the two values match, a match signal on the sin side of the meter n is generated.

(3) Compare register for cos side (MCMPn1, n = 1 to 6)

MCMPn1 is an 8-bit register that can rewrite compare values through specification of bit 4 (TENn) of Compare control register n (MCMPCn).

$\overline{\text{RESET}}$ input sets this register to 00H and clears hardware to 0. MCMPn1 is a register that supports read/write only for 8-bit access instructions. MCMPn1 compares its value with the TMCNT0, TMCNT1 value. When the two values match, a match signal on cos side of the meter n is generated.

(4) 1-bit addition circuit

The 1-bit addition circuit repeats 1-bit addition/non-addition to the PWM output alternately upon TMCNT0/TMCNT1 overflow, and enables the state of the PWM output between the current compare value and the next compare value. This circuit is controlled by bits 2 and 3 (ADBn0, ADBn1) of the MCMPCn register.

(5) Output control circuit

This circuit consists of P-channel and N-channel drivers and can drive a meter in H bridge configuration by connecting a coil. When a meter is driven in half bridge configuration, the unused pins can be used as normal output port pins.

The relation of the duty factor of the PWM signal output from the SMnm pin is indicated by the following expression (n = 1 to 6, m = 0, 1):

$$PWM_{duty} = \frac{\text{Set value of MCMPnm} \times \text{cycle of MCNT count clock}}{255 \times \text{cycle of MCNT count clock}} \times 100\% =$$

$$= \frac{\text{Set value of MCMPnm}}{255} \times 100\%$$

Cautions: 1. MCMPn0 and MCMPn1 cannot be read or written by a 16-bit access instruction.

2. MCMPn0 and MCMPn1 are in master-slave configuration, and TMCNT0/TMCNT1 is compared with the slave register. The PWM pulse is not generated until the first overflow occurs after the counting operation has been started because the compare data is not transferred to the slave.

12.3 Meter Controller/Driver Control Registers

The meter controller/driver is controlled by the following three registers.

- Timer mode control register (MCNTC0, MCNTC1)
- Compare control register n (MCMPCn)

Remark: n = 1 to 6

(1) Timer mode control register (MCNTCn) (n = 0, 1)

MCNTCn is an 8-bit register that controls the operation of the free-running up counter (TMCNT0/ TMCNT1).

MCNTCn is set with an 1-bit or an 8-bit memory manipulation instruction.

Figure 12-4: Timer Mode Control Register (MCNTC0, MCNTC1) Format (1/2)

	7	6	5	4	3	2	1	0	Address	Initial value
MCNTC0	MCAE	0	FULL	PCE	PCS	SMCL2	SMCL1	SMCL0	F180H	00H
MCNTC1	0	0	FULL	PCE	PCS	SMCL2	SMCL1	SMCL0	F1A8H	00H

Cautions: 1. When rewriting MCNTC0n to other data, stop the timer operation (PCE=0) beforehand.

2. Bit 6 of MCNTC0/MCNTC1 and Bit 7 of MCNTC1 must be set to 0.

Figure 12-4: Timer Mode Control Register (MCNTC0, MCNTC1) Format (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7</td><td>MCAENote</td><td>Meter Controller/Driver control
0: Meter C/D operation stopped (MCNTC0/MCNTC1 is reset)
1: Meter C/D operation enabled</td></tr>
<tr><td>5</td><td>FULL</td><td>Full Count Function Control:
0: TMCNTn counts from 01H to FFH (Overflow occurs at TMCNTn = FFH)Note
1: TMCNTn counts from 00H to FFH (Overflow occurs at TMCNTn = 00H)
First count starts at 00H</td></tr>
<tr><td>4</td><td>PCE</td><td>Timer Operation Control:
0: Operation is stopped (timer value is cleared)
1: Operation is enabled</td></tr>
<tr><td>3</td><td>PCS</td><td>Timer Count Clock (f_{MCO01}) Selection:
for MCNTC0
0: Selection of $f_{MC01}=f_{MC00}$
1: Selection of external clock input from TM50

for MCNTC1
0: Selection of $f_{MC11}=f_{MC10}$
1: Selection of external clock input from TM50</td></tr>
<tr><td>2 to 0</td><td>SMCLm
(m= 2 to 0)</td><td>Timer counter clock selection (f_{MCn0})
<table>
<tr><th>SMCL2</th><th>SMCL1</th><th>SMCL0</th><th>Timer Counter Clock (f_{MCn0})</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>$f_{XX}/2$</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>$f_{XX}/4$</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>$f_{XX}/8$</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>$f_{XX}/16$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>$f_{XX}/32$</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>$f_{XX}/64$</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>$f_{XX}/128$</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>$f_{XX}/256$</td></tr>
</table></td></tr>
</table>

Note: Only available in MCNTC0. This bit switches both Meter Controller/Driver sources.

Remarks: 1. n = 0,1

2. m = 2 to 0

(2) Compare control register (MCMPCn) (n = 1 to 6)

MCMPCn is an 8-bit register that controls the operation of the compare register and output direction of the PWM pin.

MCMPCn is set with an 1-bit or an 8-bit memory manipulation instruction.

Cautions: 1. Bits 5 to 7 must be set to 0.

2. TEN bit have to be “0” before rewriting of MCMPCn register. Therefore the TEN bit have to check if its “0” right before rewriting the MCMPCn register.

Figure 12-5: Compare Control Register (MCMPCn) Format (1/2)

	7	6	5	4	3	2	1	0	Address	Initial value
MCMPC1	AOUT	0	0	TEN	ADB1	ADB0	DIR1	DIR0	F194H	00H
MCMPC2	AOUT	0	0	TEN	ADB1	ADB0	DIR1	DIR0	F198H	00H
MCMPC3	AOUT	0	0	TEN	ADB1	ADB0	DIR1	DIR0	F19CH	00H
MCMPC4	AOUT	0	0	TEN	ADB1	ADB0	DIR1	DIR0	F1A0H	00H
MCMPC5	AOUT	0	0	TEN	ADB1	ADB0	DIR1	DIR0	F1B4H	00H
MCMPC6	AOUT	0	0	TEN	ADB1	ADB0	DIR1	DIR0	F1B8H	00H

Note: TEN functions as a control bit and status flag. As soon as the timer overflows and PWM data is output, TEN is cleared to “0” by hardware.

The relation (sin, cos) among the DIR1 and DIR0 bits of the MCMPCn register and output pin is shown below.

Figure 12-5: Compare Control Register (MCMPCn) Format (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7</td><td>AOUT</td><td>Selection of the PWM modes:
0: Stepper Motor PWM output signal:
PWM output at the selected pins by DIRn1, DIRn0
no PWM output (SMVss level) at the other pins
1: PWM output signal:
SMn1 = SMn2 (same PWM output signal)
SMn3 = SMn4 (same PWM output signal)</td></tr>
<tr><td>4</td><td>TEN</td><td>Transfer enable control bit from Master to Slave register:
for TMCNT0
0: Disables data transfer, new data can be written
1: Enables data transfer when TMCNT0 overflows, new data can not be written
for TMCNT1
0: Disables data transfer, new data can be written
1: Enables data transfer when TMCNT1 overflows, new data can not be written
Caution: The TEN bit functions as a control bit and status flag. As soon as the timer overflows and PWM data is output, TEN is cleared to “0” by hardware.</td></tr>
<tr><td>3</td><td>ADB1</td><td>1 Bit addition circuit control (cosine side of meter n)
0: No 1-bit addition to PWM output
1: 1-bit addition to PWM output</td></tr>
<tr><td>2</td><td>ADB0</td><td>1 Bit addition circuit control (sine side of meter n)
0: No 1-bit addition to PWM output
1: 1-bit addition to PWM output</td></tr>
<tr><td>2 to 0</td><td>DIR1, DIR0</td><td>Direction control bits (n = 1 to 6):
if AOUT=0
<table>
<tr><th>DIR1</th><th>DIR0</th><th>SMn1 (sin+)</th><th>SMn1 (sin-)</th><th>SMn1 (cos+)</th><th>SMn1 (cos-)</th></tr>
<tr><td>0</td><td>0</td><td>PWM</td><td>0</td><td>PWM</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>PWM</td><td>0</td><td>0</td><td>PWM</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>PWM</td><td>0</td><td>PWM</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>PWM</td><td>PWM</td><td>0</td></tr>
</table></td></tr>
</table>

DIR1 and DIR0 address the quadrant of sin and cos. DIR1 and DIR0 = 00 through 11 correspond to quadrants 1 through 4, respectively. The PWM signal is routed to the specific pin with respect to the sin/cos of each quadrant.

When AOUT = 1, all the output pins are used as port pins regardless of DIR1 and DIR0.
When AOUT = 0, the full bridge mode is set, and 0 a pin that does not output a PWM signal is “0”.

Cautions: 1. The output polarity of the PWM output changes when TMCNT0, TMCNT1 overflows.

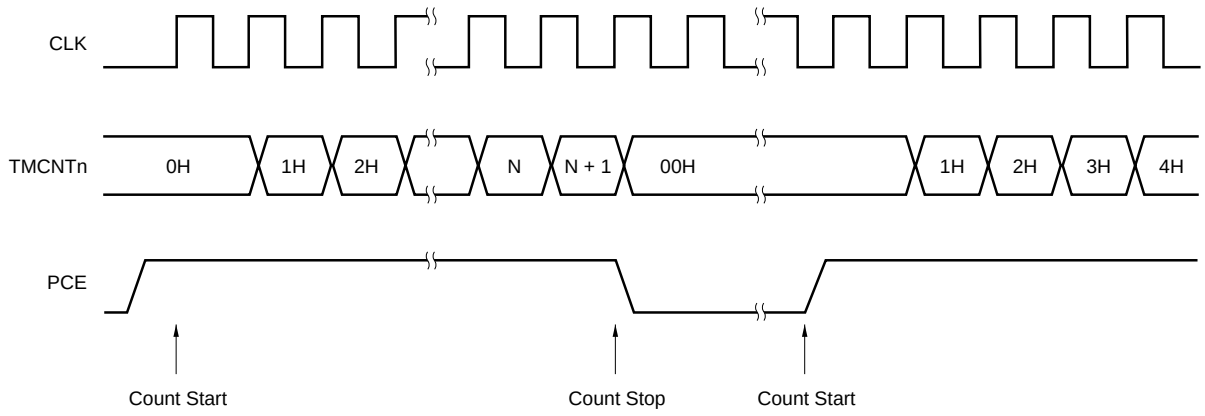
2. The MCMPCn register (n = 1 to 6) can only be changed when the TEN bit of this register is “0”. Therefore the TEN bit have to check before the MCMPC register is rewritten.

12.4 Meter Controller/Driver Operations

12.4.1 Basic operation of free-running up counter (TMCNT0, TMCNT1)

The free-running up counter is clocked by the count clock selected by the PCS bit of the timer mode control register (MCNTC0, MCNTC1).
The value of TMCNT0/TMCNT1 is cleared by $\overline{\text{RESET}}$ input.
The counting operation is enabled or disabled by the PCE bit of the timer mode control register (MCNTC0, MCNTC1).

Figure 12-6: Restart Timing after Count Stop (Count Start → Count Stop → Count Start)



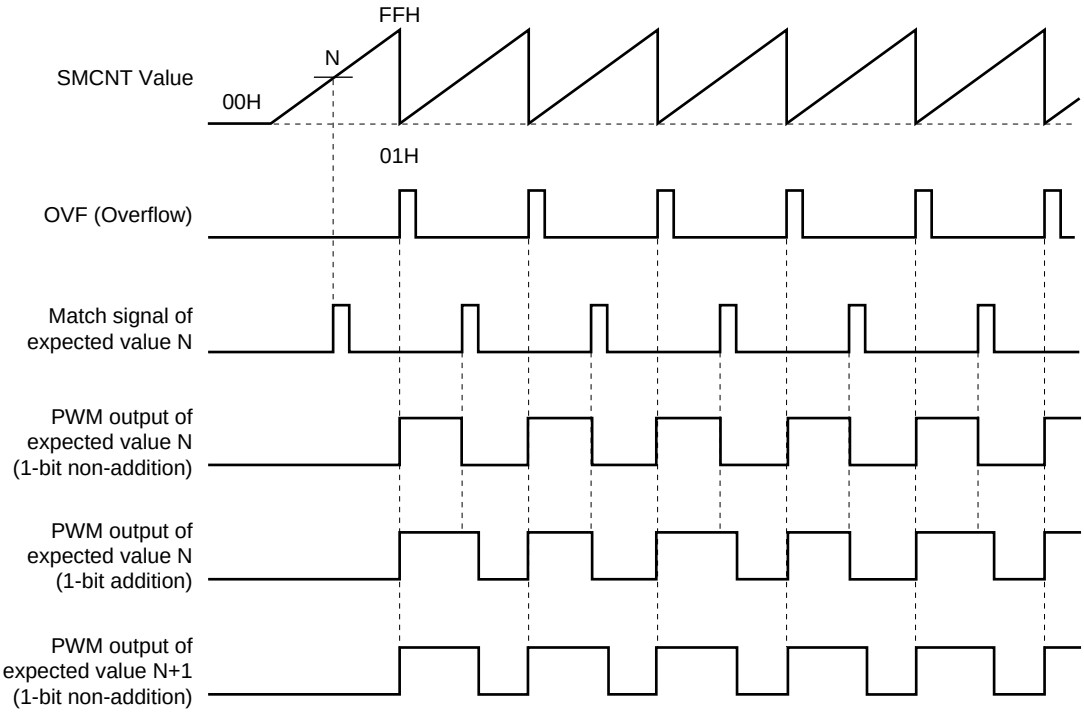
Remark: N = 00H to FFH.

12.4.2 Update of PWM data

Confirm that TEN bit of MCMPCn (n= 1 to 6) is “0”, and then write 8-bit PWM data to MCMPn1, MCMPn0, and ADB1 and ADB0 bit of MCMPCn. At the same time, set TEN bit to “1”.
The data will be automatically transferred to the slave latch when the timer overflows, and the PWM data becomes valid. At the same time, TEN bit is automatically cleared to “0”.

12.4.3 Operation of 1-bit addition circuit

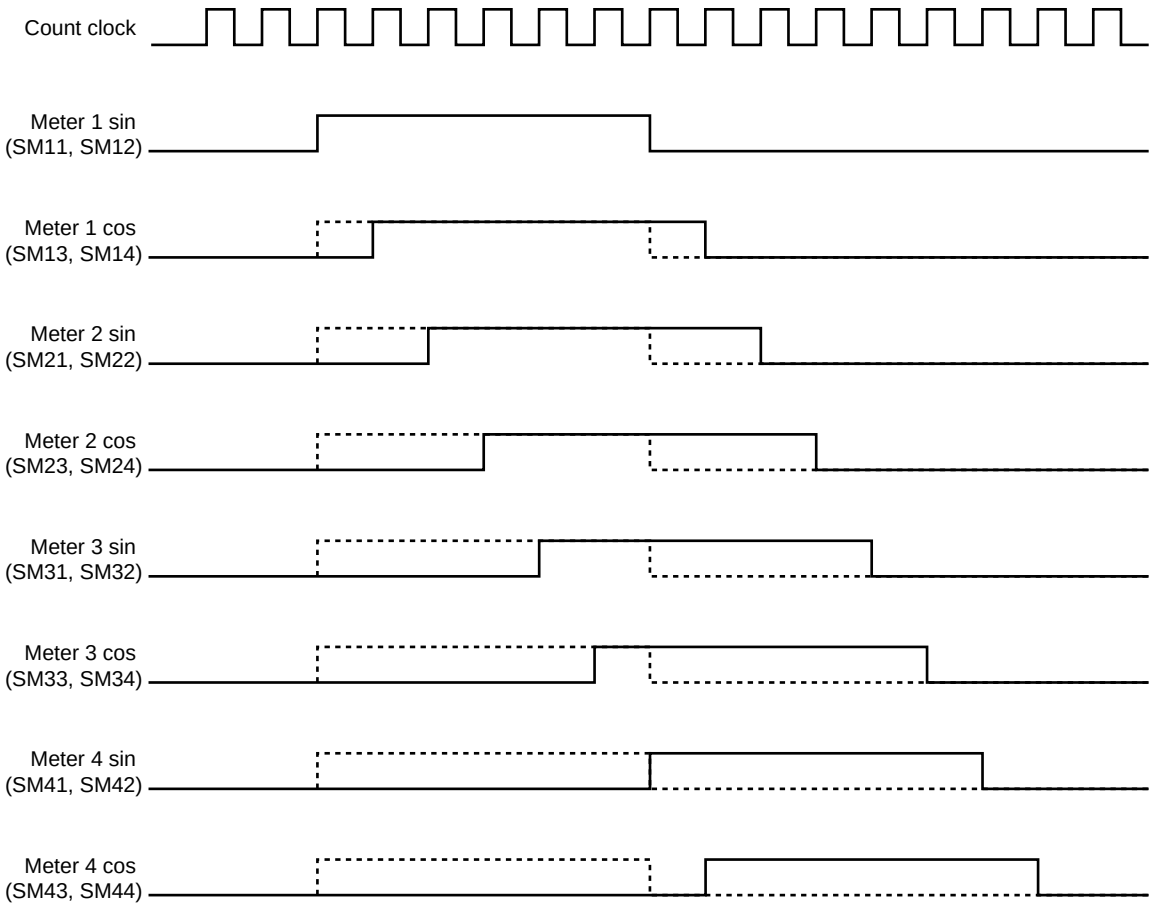
Figure 12-7: Timing in 1-bit Addition Circuit Operation



- The 1-bit addition mode repeats 1-bit addition/non-addition to the PWM output every **second** TMCNT0, TMCNT1 overflow. Therefore, the falling edge of the PWM output signal will occur at compare value N and compare value N+1 alternately.
- A 1-bit addition to the PWM output is applied by setting ADB-bit of the MCMPCn (n= 1 to 6) register to 1.
- In 1-bit non-addition mode the falling edge of the PWM output signal will always occur at compare value N+1 of TMCNT0, TMCNT1. A 1-bit non-addition (normal output) is applied by setting ADB1, ADB0 bit to “0”.

12.4.4 PWM output operation (output with 1 clock shifted)

Figure 12-8: Timing of Output with 1 Clock Shifted



If the wave of sin and cos of meters 1 to 4 rises and falls internally as indicated by the broken line, the SM11 to SM44 pins always shift the count clock by 1 clock of the timebase of controller 0.
The output delays are generated in order to prevent the SMV_{DDn}/SMV_{SSn} from fluctuating.
The count clock is set by the MCNTC0 register.

For the SM51 to SM64 pins the behaviour is similar to the procedure above.
Due to the timebase of controller 1 the count clock could be different to the timebase of controller 0.
If the wave of sin and cos of meters 5 to 6 rises and falls internally, the SM51 to SM64 pins always shift the count clock by 1 clock of the timebase of controller 1.
The count clock is set by the MCNTC1 register.

[MEMO]

Chapter 13 LCD Controller/Driver

13.1 LCD Controller/Driver Functions

The functions of the LCD controller/driver incorporated in the V850/DB1 are shown below.

(1) Automatic output of segment signals and common signals is possible by automatic writing of the display data memory.
(2) Any of two display modes can be selected.

- 1/3 duty (1/3 bias)
- 1/4 duty (1/3 bias)

(3) Any of two frame frequencies can be selected in each display mode.

(4) Maximum of 36 segment signal outputs (SEG0 to SEG35); 4 common signal outputs (COM0 to COM3).
The port mode control register (PMC) has to be set to LCD mode to allow the segment signal output. This LCD mode can be set bit-wise.

The maximum number of displayable pixels in each display mode is shown in Table 13-1, “Maximum Number of Display Pixels,” on page 407.

Table 13-1: Maximum Number of Display Pixels

Bias Method	Time division	Common Signals Used	Maximum Number of Pixels
1/3	3	COM0 - COM2	108 (36 segments x 3 commons)
	4	COM0 - COM3	144 (36 segments x 4 commons)

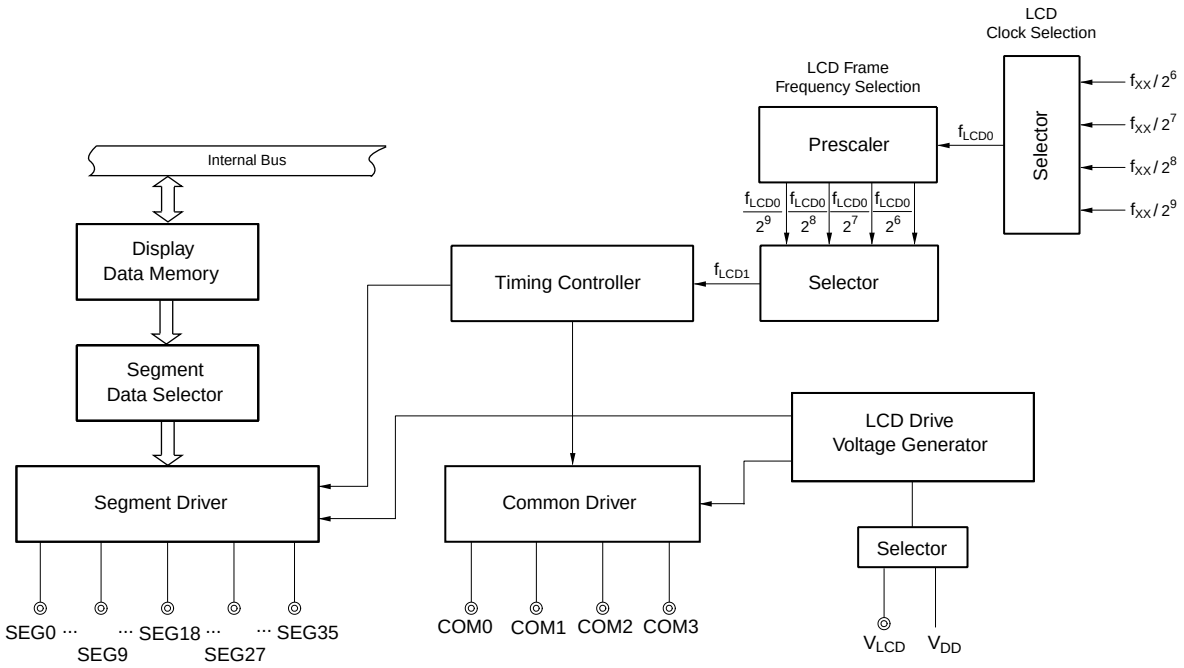
13.2 LCD Controller/Driver Configuration

The LCD controller/driver is composed of the following hardware.

Table 13-2: LCD Controller/Driver Configuration

Item	Configuration
Display outputs	Segment signals: 36 Segment signal input/output port dual function: 36 Common signals: 4 (COM0 to COM3)
Control registers	LCD display mode register (LCDM) LCD display control register (LCDC)

Figure 13-1: LCD Controller/Driver Block Diagram



13.3 LCD Controller/Driver Control Registers

The LCD controller/driver is controlled by the following two registers.

- LCD display mode register (LCDM)
- LCD display control register (LCDC)

(1) LCD display mode register (LCDM)

This register sets display operation enabling/ disabling, the LCD driving power and the LCD display mode.

LCDM is set with an 1-bit or an 8-bit memory manipulation instruction.

Figure 13-2: LCD Display Mode Register Format

	7	6	5	4	3	2	1	0	Address	Initial value
LCDM	LCDON	0	SCOC	LIPS	0	0	0	LCDM0	FFFF F392H	00H

Caution: Bits 1, 2, 3 and 6 must be set to 0.

LCDON	LCD Display Enable/Disable
0	Display OFF
1	Display ON

SCOC	Segment/ Common Output Control
0	All segment/Common outputs are VSS level (independent of LCD RAM Data)
1	All segment/Common outputs are select signal output or non-select signal output

LIPS	LCD Driving Power Supply Selection
0	Supplies external power to the LCD driver from V_{LCD} pin.
1	Supplies internal power to the LCD driver from V_{DD} pin.

LCDM0	Display Mode Selection Option
0	1/4 Duty, 1/3 bias
1	1/3 Duty, 1/3 bias

(2) LCD display clock control register (LCDC)

This register sets the LCD clock.

LCDC is set with an 8-bit memory manipulation instruction.

Figure 13-3: LCD Display Clock Control Register (LCDC)

Symbol	7	6	5	4	3	2	1	0	Address	Initial value
LCDC	0	0	0	0	LCDC3	LCDC2	LCDC1	LCDC0	FFFF F390H	00H

Caution: Bits 4, 5, 6 and 7 must be set to 0.

LCDC3	LCDC2	LCD clock selection (f_{LCD0})
0	0	$f_X / 2^6$
0	1	$f_X / 2^7$
1	0	$f_X / 2^8$
1	1	$f_X / 2^9$

LCDC1	LCDC0	LCD frame frequency selection (f_{LCD1})**Note**
0	0	$f_{LCD0}/2^6$ (488 Hz)
0	1	$f_{LCD0}/2^7$ (244 Hz)
1	0	$f_{LCD0}/2^8$ (122 Hz)
1	1	$f_{LCD0}/2^9$ (61 Hz)

Note: f_X = 4 MHz, $f_{LCD0} = f_X / 2^7$

Table 13-3: Frame Frequencies [Hz] (fx = 4 MHz)

LCD3	LCD2	LCD1	LCD0	Divider	duty-cycle	
					1/3	1/4
0	0	0	0	2^{12}	325.3	244
0	0	0	1	2^{13}	162.7	122
0	0	1	0	2^{14}	81.3	61
0	0	1	1	2^{15}	40.7	30.5
0	1	0	0	2^{13}	162.7	122
0	1	0	1	2^{14}	81.3	61
0	1	1	0	2^{15}	40.7	30.5
0	1	1	1	2^{16}	20.3	15.3
1	0	0	0	2^{14}	81.3	61
1	0	0	1	2^{15}	40.7	30.5
1	0	1	0	2^{16}	20.3	15.3
1	0	1	1	2^{17}	10.2	7.6
1	1	0	0	2^{15}	40.7	30.5
1	1	0	1	2^{16}	20.3	15.3
1	1	1	0	2^{17}	10.2	7.6
1	1	1	1	2^{18}	5.1	3.8

13.4 LCD Controller/Driver Settings

LCD controller/driver settings should be performed as shown below. When the LCD controller/driver is used, the watch timer should be set to the operational state beforehand.

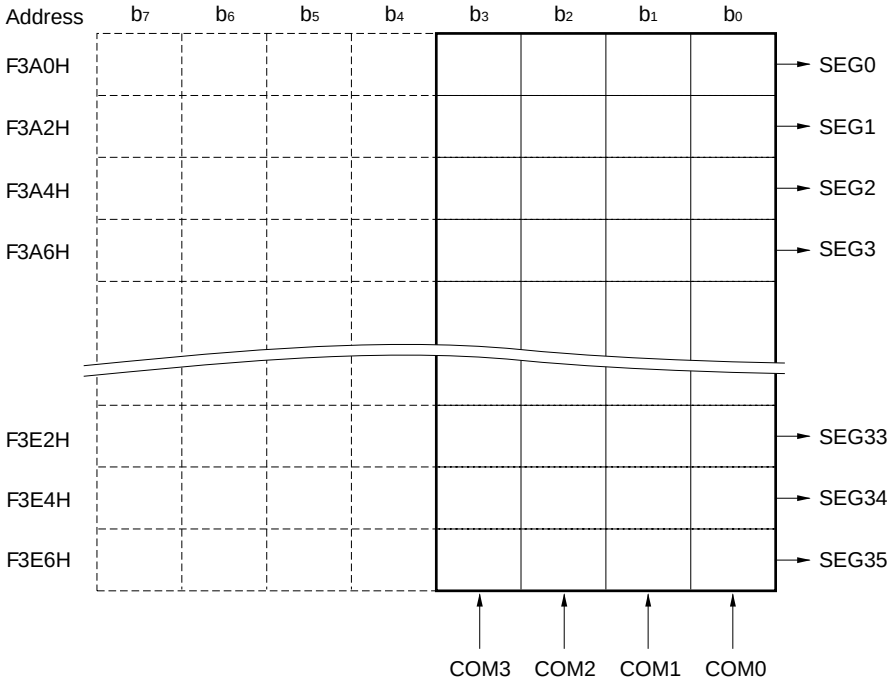
<1> Set the initial value in the display data memory (FFFF F3A0H to FFFF F3E6H).
<2> Set the pins to be used as segment outputs in the port function registers (P4, P5, P6, P9, P13, P14).
<3> Set the display mode, operating mode in the LCD display mode register (LCDM), and the LCD clock in the LCD clock control register (LCDC).

Next, set data in the display data memory according to the display contents

13.5 LCD Display Data Memory

The LCD display data memory is mapped onto addresses FFFF F3A0H to FFFF F3E6H. The data stored in the LCD display data memory can be displayed on an LCD panel by the LCD controller/driver. Figure 13-4, “Relationship between LCD Display Data Memory Contents and Segment/Common Outputs,” on page 413 shows the relationship between the LCD display data memory contents and the segment outputs/common outputs.
Any area not used for the display can be used as normal RAM.

Figure 13-4: Relationship between LCD Display Data Memory Contents and Segment/Common Outputs



Caution: The higher 4 bits of the LCD display data memory do not incorporate memory. Be sure to set them to 0.

13.6 Common Signals and Segment Signals

An individual pixel on an LCD panel lights when the potential difference of the corresponding common signal and segment signal reaches or exceeds a given voltage (the LCD drive voltage V_{LCD}).

As an LCD panel deteriorates if a DC voltage is applied in the common signals and segment signals, it is driven by AC voltage.

(1) Common signals

For common signals, the selection timing order is as shown in Table 13-4, “COM Signals,” on page 414 according to the number of time divisions set, and operations are repeated with these as the cycle. With 3-time-division operation, the COM3 pin is left open.

Table 13-4: COM Signals

COM signal / Time division	COM0	COM1	COM2	COM3
3-time division				Open
4-time division				

(2) Segment signals

Segment signals correspond to a 36-byte LCD display data memory. Each display data memory bit 0, bit 1, bit 2, and bit 3 is read in synchronization with the COM0, COM1, COM2 and COM3 timings respectively, and if the value of the bit is 1, it is converted to the selection voltage. If the value of the bit is 0, it is converted to the non-selection voltage and output to a segment pin (SEG0 to SEG35).

Consequently, it is necessary to check what combination of front surface electrodes (corresponding to the segment signals) and rear surface electrodes (corresponding to the common signals) of the LCD display to be used form the display pattern, and then write bit data corresponding on a one-to-one basis with the pattern to be displayed.

In addition, because LCD display data memory bit 3 is not used with the 3-time-division method, these can be used for other than display purposes.

Bits 4 to 7 are fixed at 0.

(3) Common signal and segment signal output waveforms

The voltages shown in Table 13-4, “COM Signals,” on page 414 are output in the common signals and segment signals.

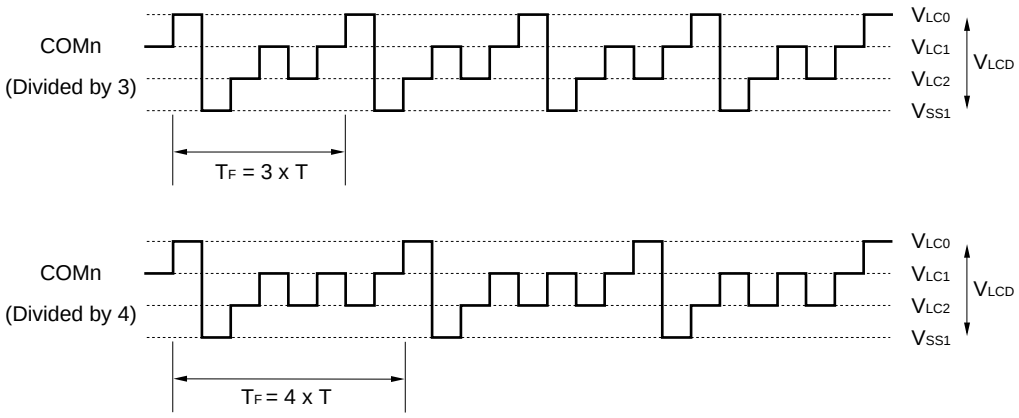
The $\pm V_{LCD}$ ON voltage is only produced when the common signal and segment signal are both at the selection voltage; other combinations produce the OFF voltage.

Table 13-5: LCD Drive Voltages 1/3 bias method

Common \ Segment		Select	Non-Select
		V_{SS1}, V_{LC0}	V_{LC1}, V_{LC2}
Select level	V_{LC0}, V_{SS1}	$-V_{LCD}$, $+V_{LCD}$	-1/3 V_{LCD}, +1/3 V_{LCD}
Non-select level	V_{LC2}, V_{LC1}	-1/3 V_{LCD}, +1/3 V_{LCD}	-1/3 V_{LCD}, +1/3 V_{LCD}

Figure 13-5, “Common Signal Waveform 1/3 bias method,” on page 415 shows the common signal waveform, and Figure 13-6, “Common Signal and Static Signal Voltages and Phases 1/3 bias method,” on page 416 shows the common signal and segment signal voltages and phases.

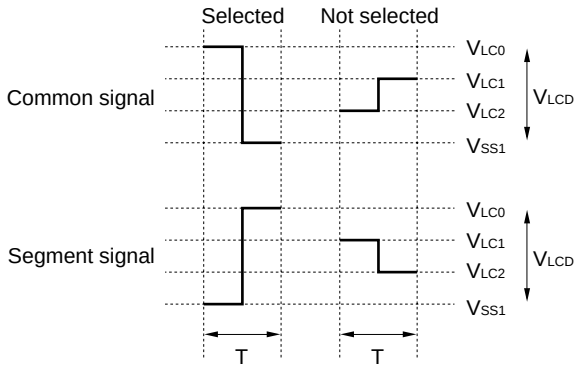
Figure 13-5: Common Signal Waveform 1/3 bias method



Remarks: 1. T: One LCDCL cycle

2. TF: Frame frequency

Figure 13-6: Common Signal and Static Signal Voltages and Phases 1/3 bias method



Remark: T: One LCDCL cycle

13.7 Display Modes

13.7.1 3-time-division display example

Figure 13-8, “3-Time-Division LCD Panel Connection Example,” on page 418 shows the connection of a 3-time-division type 12-digit LCD panel with the display pattern shown in Figure 13-7, “3-Time-Division LCD Display Pattern and Electrode Connections,” on page 417 with segment signals (SEG0 to SEG35) and common signals (COM0 to COM2). The display example is “23456.7890123,” and the display data memory contents correspond to this.
An explanation is given here taking the example of the eighth digit “8.”. In accordance with the display pattern in Figure 13-7, “3-Time-Division LCD Display Pattern and Electrode Connections,” on page 417, selection and non-selection voltages must be output to pins SEG21 through SEG23 as shown in Table 13-6, “Selection and Non-Selection Voltages (COM0 to COM2),” on page 417 at the COM0 to COM2 common signal timings.

Table 13-6: Selection and Non-Selection Voltages (COM0 to COM2)

Common	Segment		
	SEG21	SEG22	SEG23
COM0	NS	S	S
COM1	S	S	S
COM2	S	S	-

Remark: S: Selection, NS: Non-selection

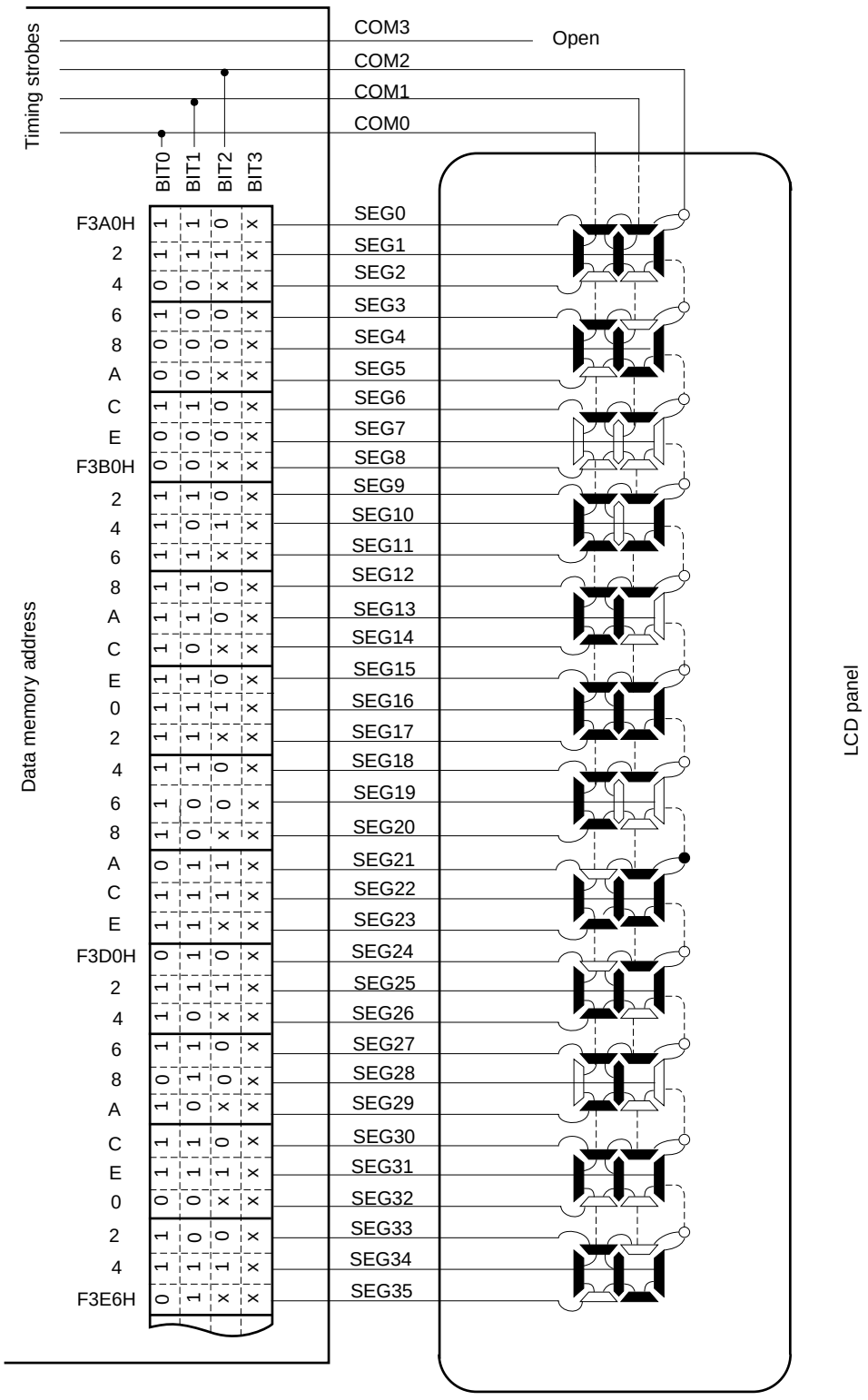
From this, it can be seen that X110B must be prepared in the display data memory (address FFFF F3CAH) corresponding to SEG21.

Examples of the LCD drive waveforms between SEG21 and the common signals are shown in Figure 13-9, “3-Time-Division LCD Drive Waveform Examples (1/3 Bias Method),” on page 419.
When SEG21 is at the selection voltage at the COM1 selection timing, and SEG21 is at the selection voltage at the COM2 selection timing, it can be seen that the $+V_{LCD}/-V_{LCD}$ AC square wave, which is the LCD illumination (ON) level, is generated.

Figure 13-7: 3-Time-Division LCD Display Pattern and Electrode Connections

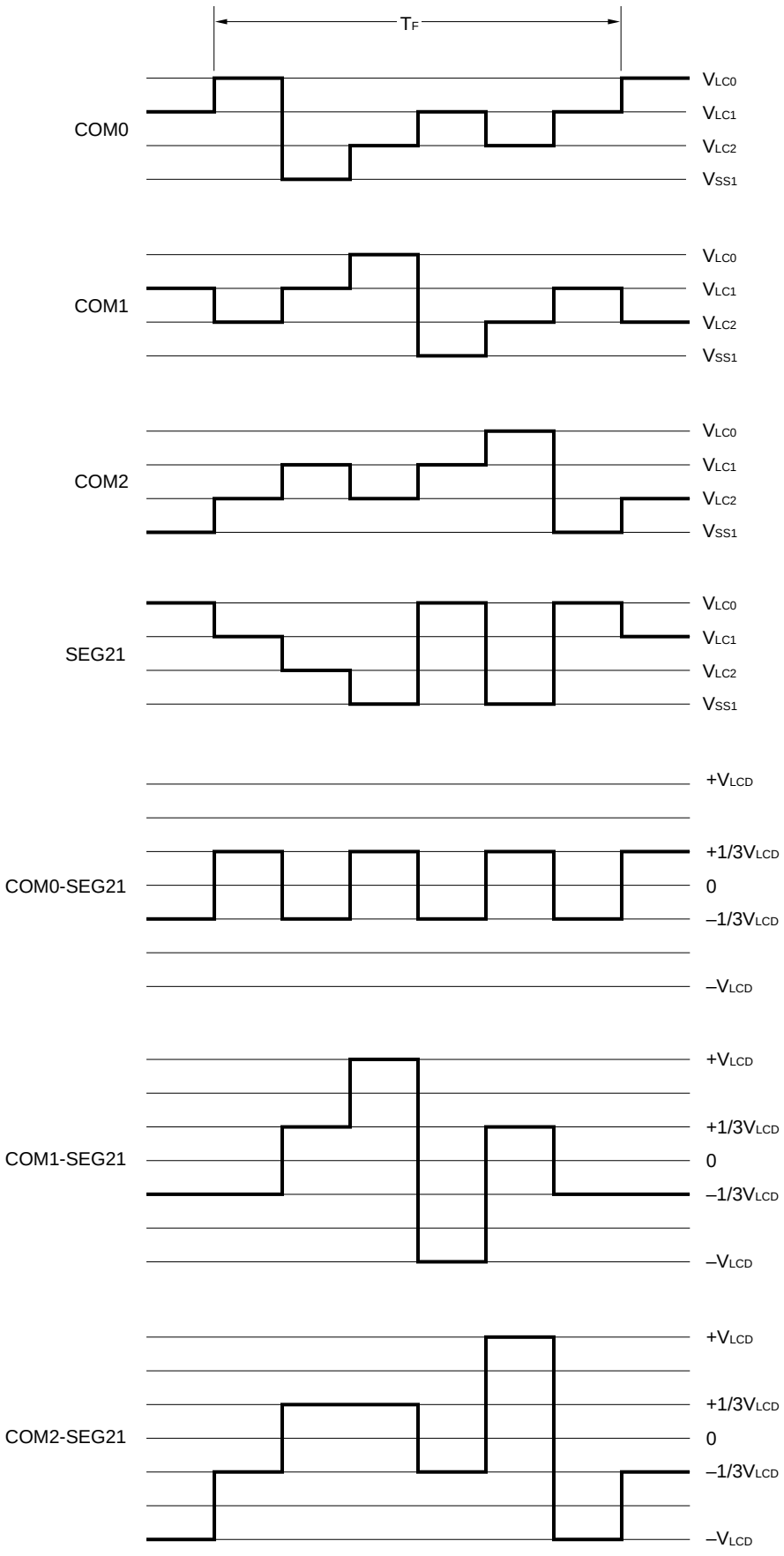
Remark: n = 0 to 12

Figure 13-8: 3-Time-Division LCD Panel Connection Example



Remark: X: Irrelevant bits because this is a 3-time-division display

Figure 13-9: 3-Time-Division LCD Drive Waveform Examples (1/3 Bias Method)



13.7.2 4-time-division display example

Figure 13-11, “4-Time-Division LCD Panel Connection Example,” on page 421 shows the connection of a 4-time-division type 18-digit LCD panel with the display pattern shown in Figure 13-10, “4-Time-Division LCD Display Pattern and Electrode Connections,” on page 420 with segment signals (SEG0 to SEG35) and common signals (COM0 to COM3). The display example is “123456.789012345678,” and the display data memory contents correspond to this.
An explanation is given here taking the example of the 15th digit “5.”. In accordance with the display pattern in Figure 13-10, “4-Time-Division LCD Display Pattern and Electrode Connections,” on page 420, selection and non-selection voltages must be output to pins SEG28 and SEG29 as shown in Table 13-7, “Selection and Non-Selection Voltages (COM0 to COM3),” on page 420 at the COM0 to COM3 common signal timings.

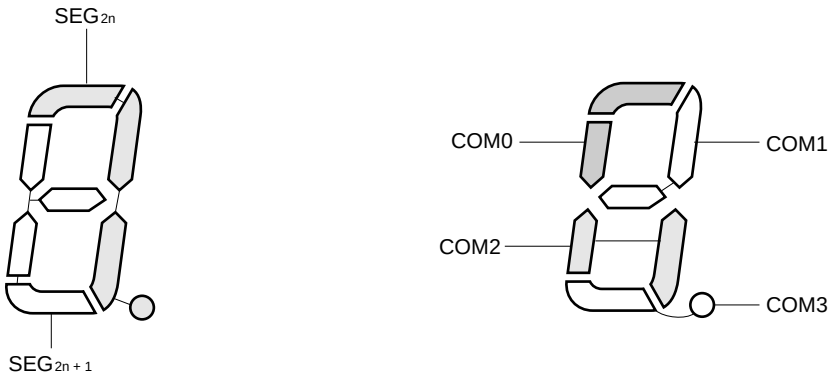
Table 13-7: Selection and Non-Selection Voltages (COM0 to COM3)

Common	Segment	
	SEG28	SEG29
COM0	S	S
COM1	NS	S
COM2	S	S
COM3	S	S

Remark: S: Selection, NS: Non-selection

From this, it can be seen that 1101B must be prepared in the display data memory
(address FFFF F3D8H) corresponding to SEG28.
Examples of the LCD drive waveforms between SEG28 and the COM0 and COM1 signals are shown in Figure 13-12, “4-Time-Division LCD Drive Waveform Examples (1/3 Bias Method),” on page 422 (for the sake of simplicity, waveforms for COM2 and COM3 have been omitted). When SEG28 is at the selection voltage at the COM0 selection timing, it can be seen that the $+V_{LCD}/-V_{LCD}$ AC square wave, which is the LCD illumination (ON) level, is generated.

Figure 13-10: 4-Time-Division LCD Display Pattern and Electrode Connections



Remark: n = 0 to 18

Figure 13-11: 4-Time-Division LCD Panel Connection Example

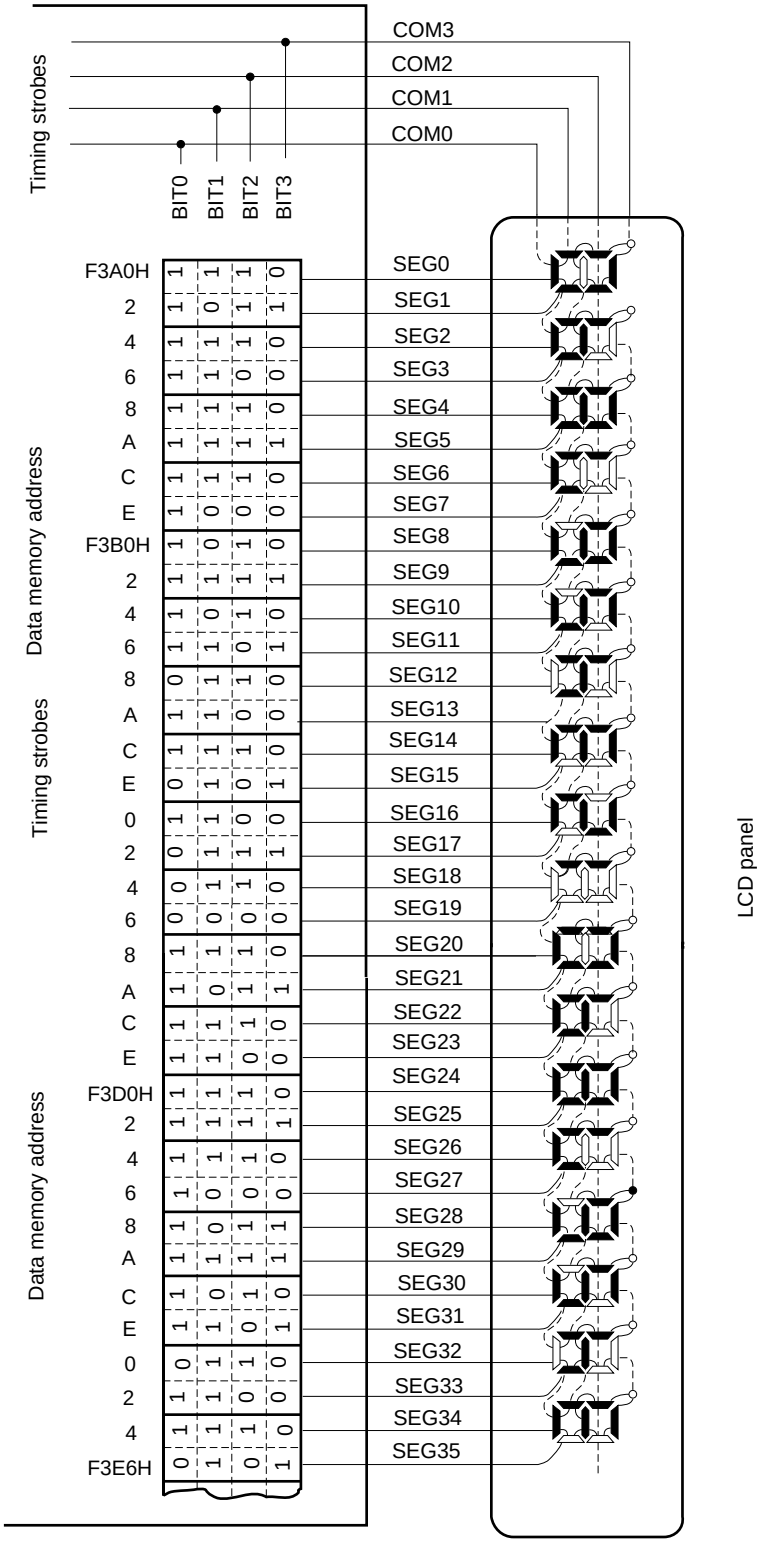
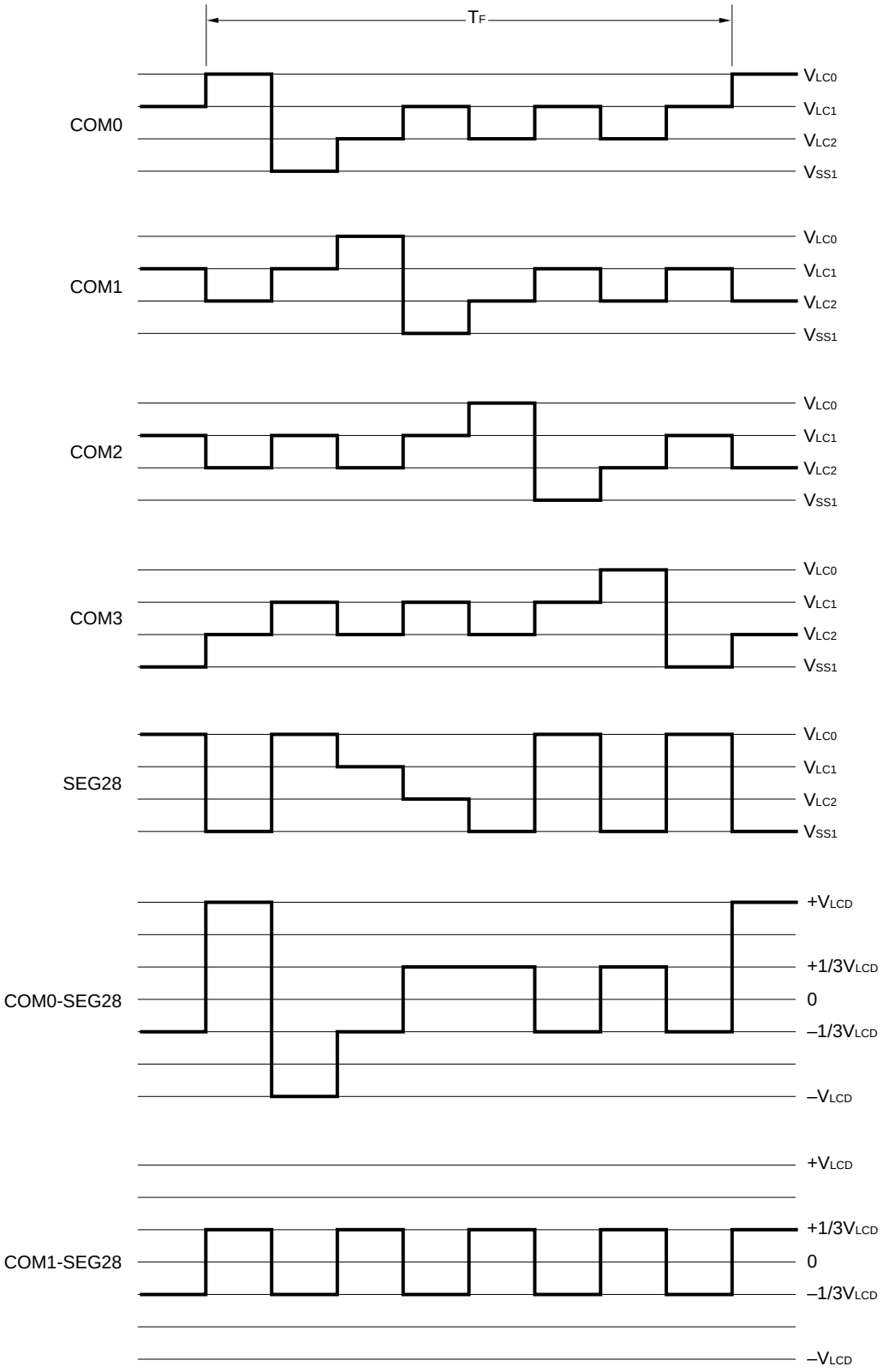


Figure 13-12: 4-Time-Division LCD Drive Waveform Examples (1/3 Bias Method)



Chapter 14 Port Functions

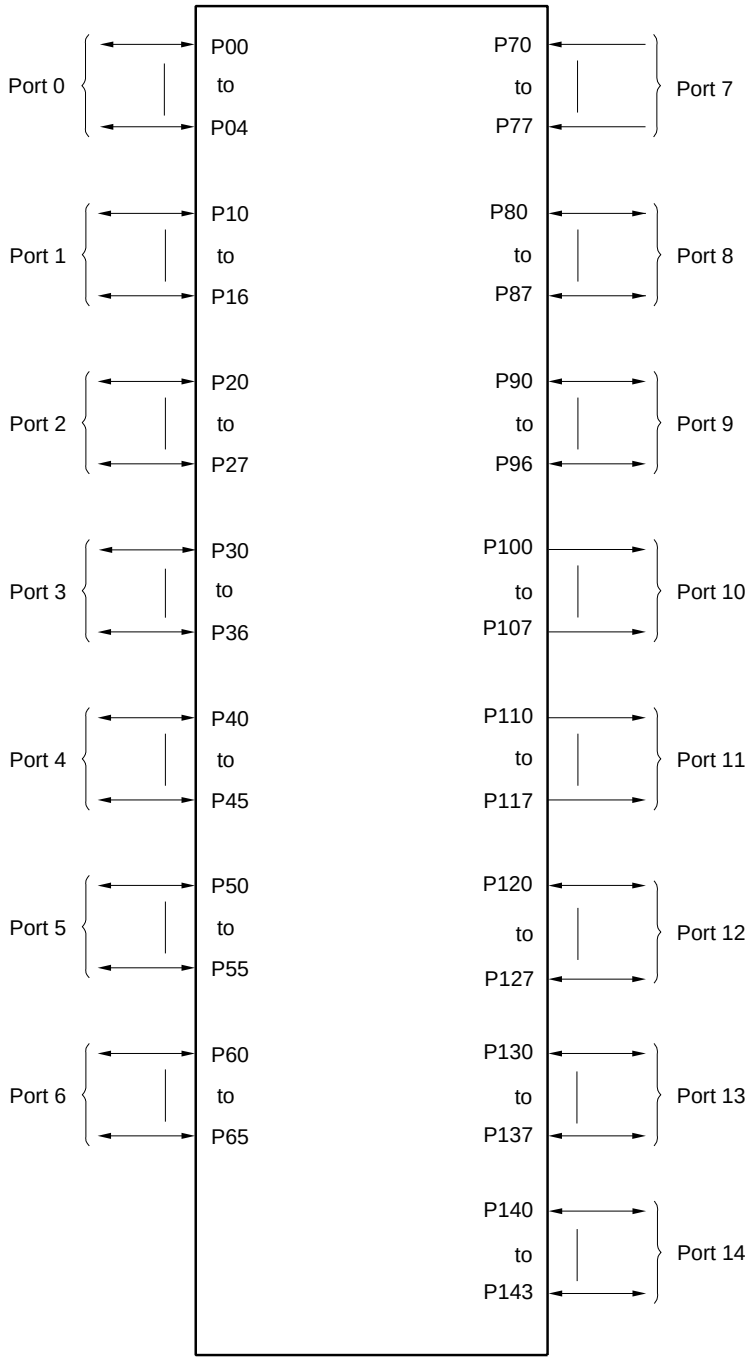
14.1 Features

- Input/Output ports: 15
- ports pins: 107
- Ports alternate as input/output pins of other peripheral functions for ports 0 to 6, 8, 9, and 12 to 14
- Input or output can be specified in bit units
- software switchable pull-up resistor for ports 0 to 3 and 8.

14.2 Port Configuration

The V850/DB1 incorporates a total of 107 input/output ports, named ports P0 through P14. The configuration is shown below.

Figure 14-1: Port Configuration



(1) Functions of each port

The V850/DB1 has the ports shown below.
Any port can operate in 8-bit or 1-bit units and can provide a variety of controls.
Moreover, besides its function as a port, the most of them has functions as the input/output pins of on-chip peripheral I/O.
The selection of the port function or alternate peripheral function is mentioned in Table 14-2 on page 426 in the Mode Logic column.
Refer to Chapter 14.2.1 “Port block diagram” on page 430 for a block diagram of the block type of each port.

Table 14-1: Functions of Each Port (1/2)

Port Name	Pin Name	Block Type	Port Function	Function In Alternate Mode
Port 0	P00 to P04	A	5-bit input/output port	External interrupt input
Port 1	P10	A	7-bit input/output port	Serial interface (CSI00, UART50, UART51) input/output External interrupt input
	P11	B		
	P12	C		
	P13	A		
	P14	C		
	P15	A		
	P16	C		
Port 2	P20	A	8-bit input/output port	16 Bit Timer G inputs/outputs 8 Bit Timer 50/51 inputs/outputs
	P21	B		
	P22	B		
	P23	B		
	P24	B		
	P25	A		
	P26	B		
	P27	B		
Port 3	P30	B	6-bit input/output port	Processor Clock output 16 Bit Timer C0/C1 inputs/outputs
	P31	B		
	P32	A		
	P33	B		
	P34	A		
	P35	A		
Port 4	P40 to P47	E	8-bit input/output port	LCD Segment output
Port 5	P50 to P57	E	8-bit input/output port	LCD Segment output
Port 6	P60 to P65	E	6-bit input/output port	LCD Segment output
Port 7	P70 to P77	D	8-bit input port	Analog/Digital Converter inputs
Port 8	P80	A	8-bit input/output port	Serial interface (CSI02, DCAN0, DCAN1) input/output
	P81	B		
	P82	C		
	P83	A		
	P84	C		
	P85	A		
	P86	C		
	P87	A		

Table 14-1: Functions of Each Port (2/2)

Port Name	Pin Name	Block Type	Port Function	Function In Alternate Mode
Port 9	P90 to P96	E	7-bit input/output port	LCD Segment output
Port 10	P100 to P107	F	8-bit output port	Meter-PWM output
Port 11	P110 to P117	F	8-bit output port	Meter-PWM output
Port 12	P120 to P127	E	8-bit input/output port	Meter-PWM output
Port 13	P130 to P137	E	8-bit input/output port	LCD Segment output
Port 14	P140	E	4-bit input/output port	Serial interface input/output (CSI02) LCD Segment output
	P141	G		
	P142	H		
	P143	I		

(2) Functions of each port pin on reset and registers that set port or control mode

Table 14-2: Functions of each port pin on reset and registers that set port or control mode (1/4)

Port Name	Pin Name	Pin Function after Reset Single-Chip Mode	Mode Logic **Note**	Mode-Setting Register
P0	P00/NMI	Port Mode (input mode)	SIM	-
	P01/INTP0			
	P02/INTP1			
	P03/INTP2			
	P04/INPT3			
P1	P10/SI00	Port Mode (input mode)	SIM	-
	P11/SO00		OR	
	P12/$\overline{\text{SCK00}}$		AND	
	P13/INTP4/RXD50		SIM	
	P14/TXD50		AND	
	P15/INTP5/RXD51		SIM	
	P16/TXD51		AND	
P2	P20/TIG0	Port Mode (input mode)	SIM	-
	P21/TIG1/TOG1		OR	
	P22/TIG2/TOG2		OR	
	P23/TIG3/TOG3		OR	
	P24/TIG4/TOG4		OR	
	P25/TIG5		SIM	
	P26/TI50/TO50		OR	
	P27/TI50/TO51		OR	

Table 14-2: Functions of each port pin on reset and registers that set port or control mode (2/4)

Port Name	Pin Name	Pin Function after Reset Single-Chip Mode	Mode Logic **Note**	Mode-Setting Register
P3	P30/PCL	Port Mode (input mode)	OR	-
	P31/TIC00/TOC0		OR	
	P32/TIC10		SIM	
	P33/TIC01/TOC1		OR	
	P34/TIC11		SIM	
	P35		-	
P4	P40/COM0	Port Mode (input mode)	Selector	PMC4
	P41/COM1			
	P42/COM2			
	P43/COM3			
	P44/SEG0			
	P45/SEG1			
	P46/SEG2			
	P47/SEG3			
P5	P50/SEG4	Port Mode (input mode)	Selector	PMC5
	P51/SEG5			
	P52/SEG6			
	P53/SEG7			
	P54/SEG8			
	P55/SEG9			
	P56/SEG10			
	P57/SEG11			
P6	P60/SEG12	Port Mode (input mode)	Selector	PMC6
	P61/SEG13			
	P62/SEG14			
	P63/SEG15			
	P64/SEG16			
	P65/SEG17			
P7	P70/ANI0	Port Mode (input mode)	SIM	-
	P71/ANI1			
	P72/ANI2			
	P73/ANI3			
	P74/ANI4			
	P75/ANI5			
	P76/ANI6			
	P77/ANI7			

Table 14-2: Functions of each port pin on reset and registers that set port or control mode (3/4)

Port Name	Pin Name	Pin Function after Reset Single-Chip Mode	Mode Logic **Note**	Mode-Setting Register
P8	P80/SI01	Port Mode (input mode)	SIM	-
	P81/SO01		OR	
	P82/$\overline{\text{SCK01}}$		AND	
	P83/CRXD10		SIM	
	P84/CTXD10		AND	
	P85/CRXD11		SIM	
	P86/CTXD11		AND	
	P87		-	
P9	P90/SEG18	Port Mode (input mode)	Selector	PMC9
	P91/SEG19			
	P92/SEG20			
	P93/SEG21			
	P94/SEG22			
	P95/SEG23			
	P96/SEG24			
P10	P100/SM11	Port Mode (output disable mode, high Z)	Selector	PMC10
	P101/SM12			
	P102/SM13			
	P103/SM14			
	P104/SM21			
	P105/SM22			
	P106/SM23			
	P107/SM24			
P11	P110/SM31	Port Mode (output disable mode, high Z)	Selector	PMC11
	P111/SM32			
	P112/SM33			
	P113/SM34			
	P114/SM41			
	P115/SM42			
	P116/SM43			
	P117/SM44			
P12	P120/SM51	Port Mode (input mode)	Selector	PMC12
	P121/SM52			
	P122/SM53			
	P123/SM54			
	P124/SM61			
	P125/SM62			
	P126/SM63			
	P127/SM64			

Table 14-2: Functions of each port pin on reset and registers that set port or control mode (4/4)

Port Name	Pin Name	Pin Function after Reset Single-Chip Mode	Mode Logic **Note**	Mode-Setting Register
P13	P130/SEG25	Port Mode (input mode)	Selector	PMC13
	P131/SEG26			
	P132/SEG27			
	P133/SEG28			
	P134/SEG29			
	P135/SEG30			
	P136/SEG31			
	P137/SEG32			
P14	P140/SI02/SEG33	Port Mode (input mode)	SIM/ Selector	PMC14
	P141/SO02/SEG34		OR/ Selector	
	P142/$\overline{\text{SCK02}}$/SEG35		AND/ Selector	
	P143		-	-

Note: The port pins of the V850/DB1 has different properties for the output/input mode configuration. The following list shows the meaning of the abbreviation:

- SIM: the port function and alternate function are **simultaneously** available, this is only possible for alternate input functions
- AND: the port function and alternate function are **AND** connected (i.e. if only the alternate function should output then the port has to be set to “1”)
- OR: the port function and alternate function are **OR** connected, (i.e. if only the alternate function should output then the port has to be set to “0”)
- Selector: the port function and alternate function are connected via a hardware **selector**, that is software controlled.

14.2.1 Port block diagram

Figure 14-2: Type A Block Diagram

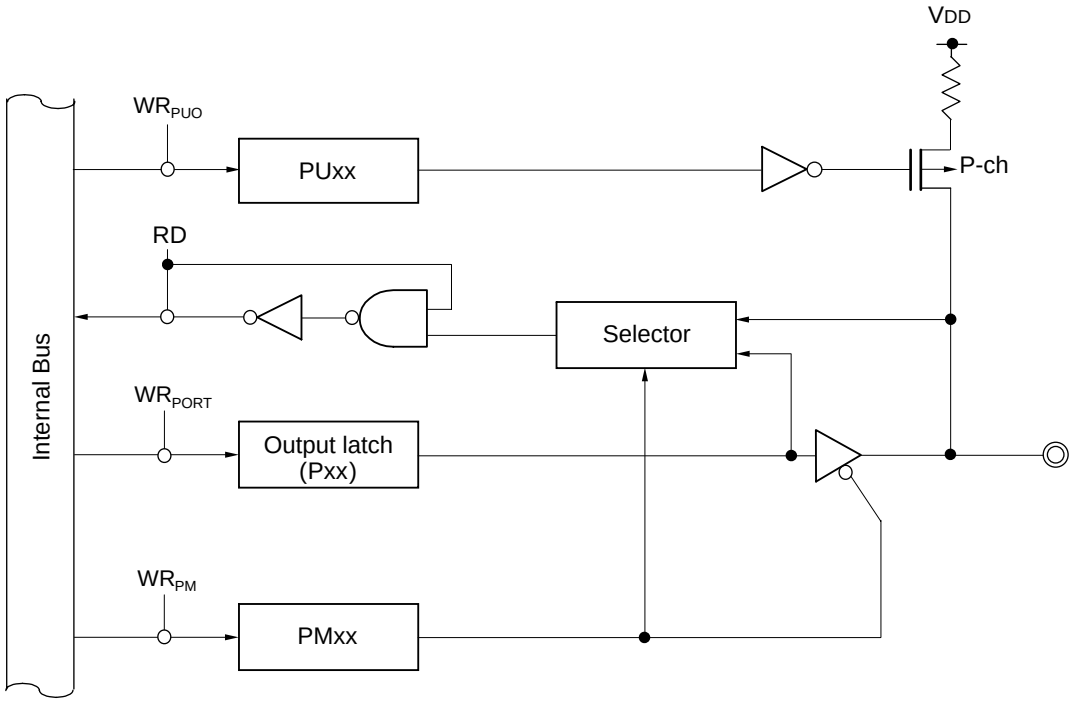


Figure 14-3: Type B Block Diagram

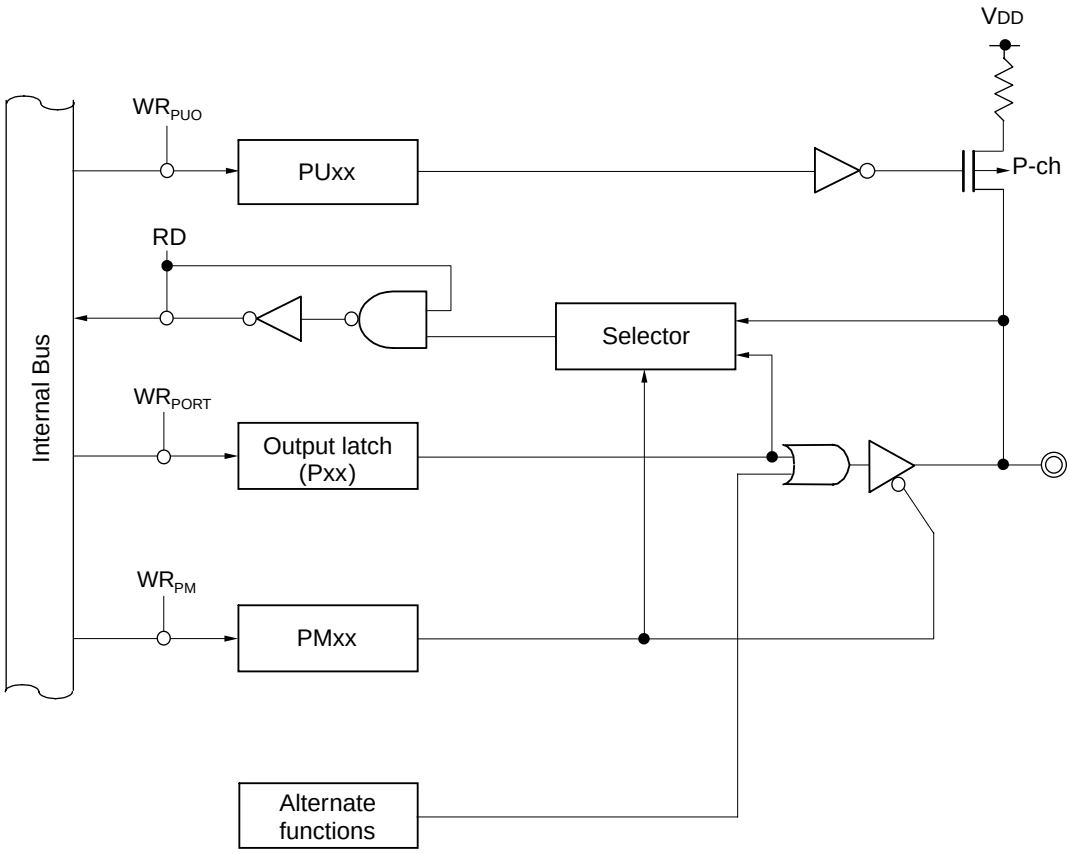


Figure 14-4: Type C Block Diagram

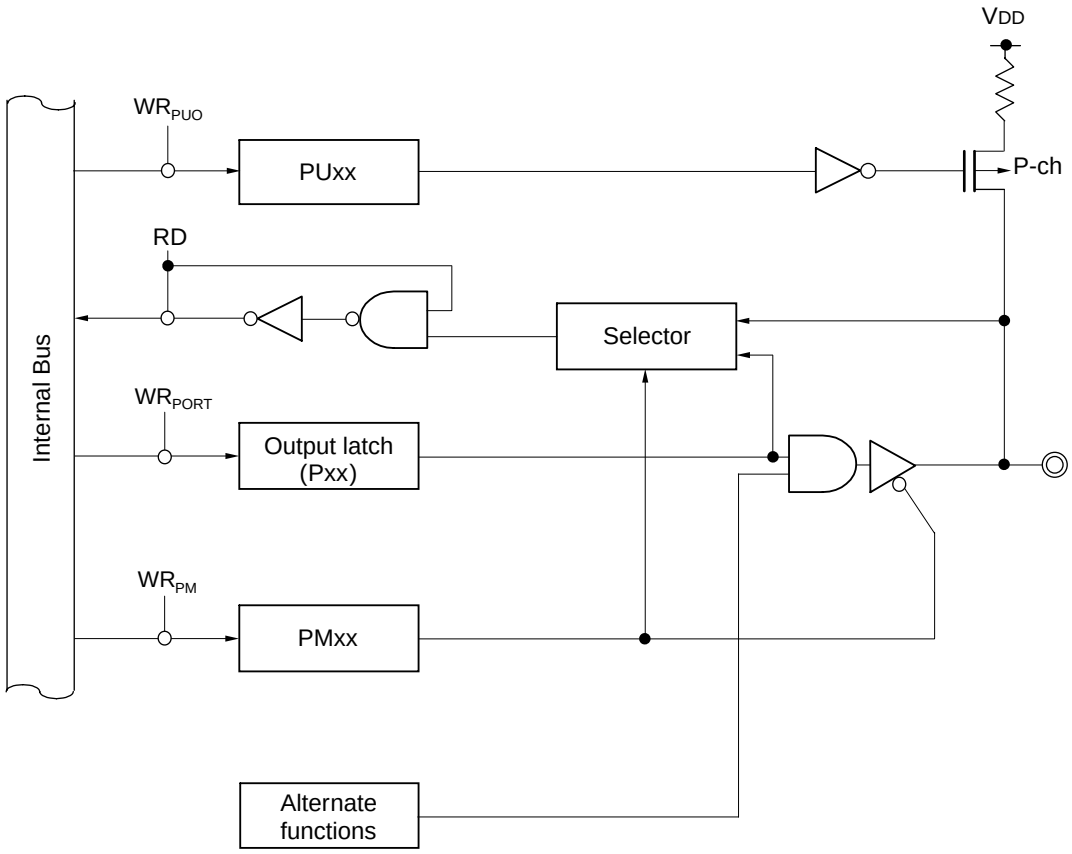


Figure 14-5: Type D Block Diagram

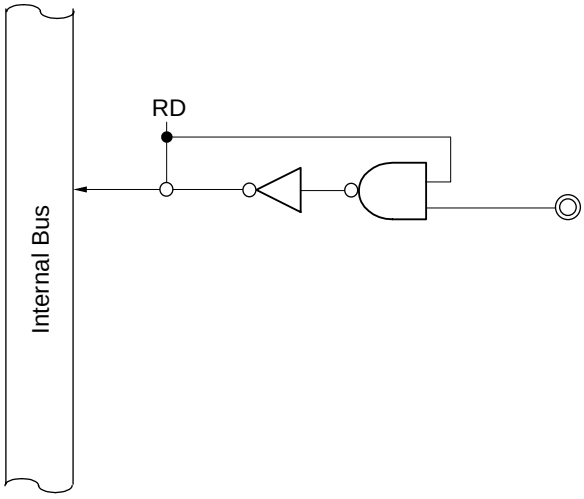


Figure 14-6: Type E Block Diagram

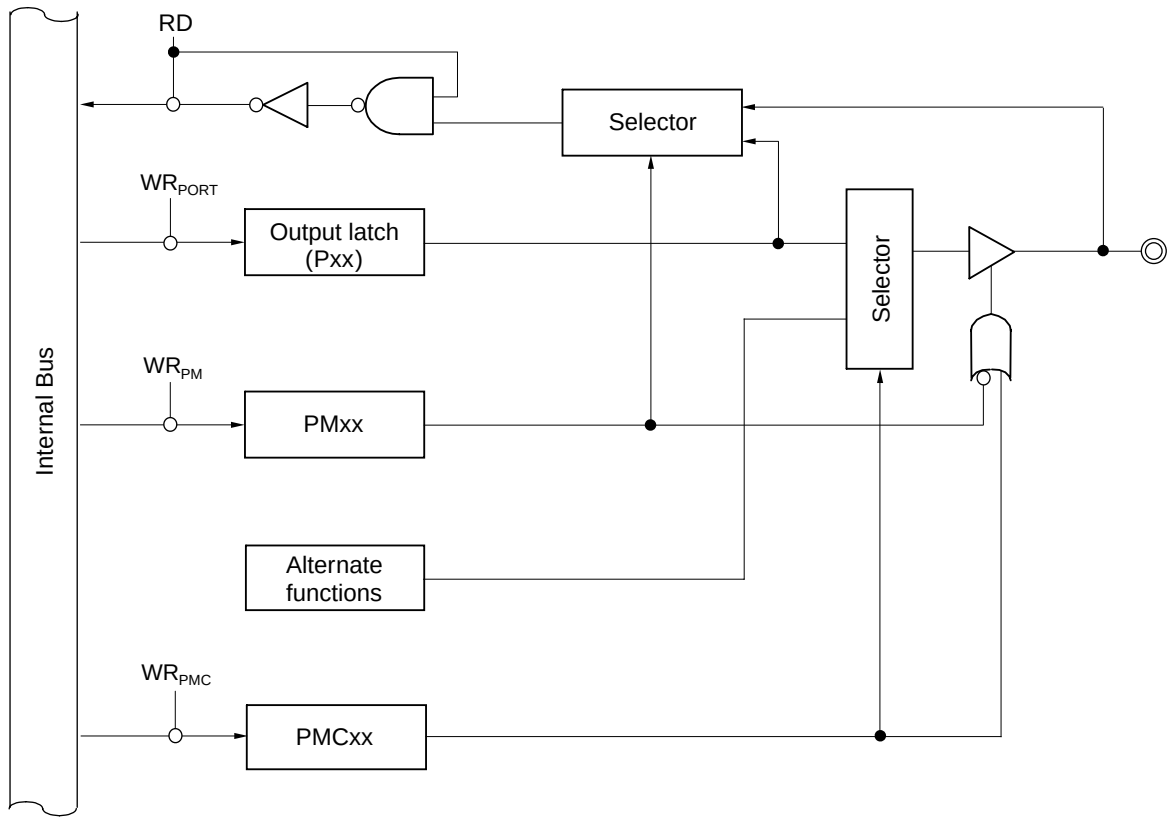


Figure 14-7: Type F Block Diagram

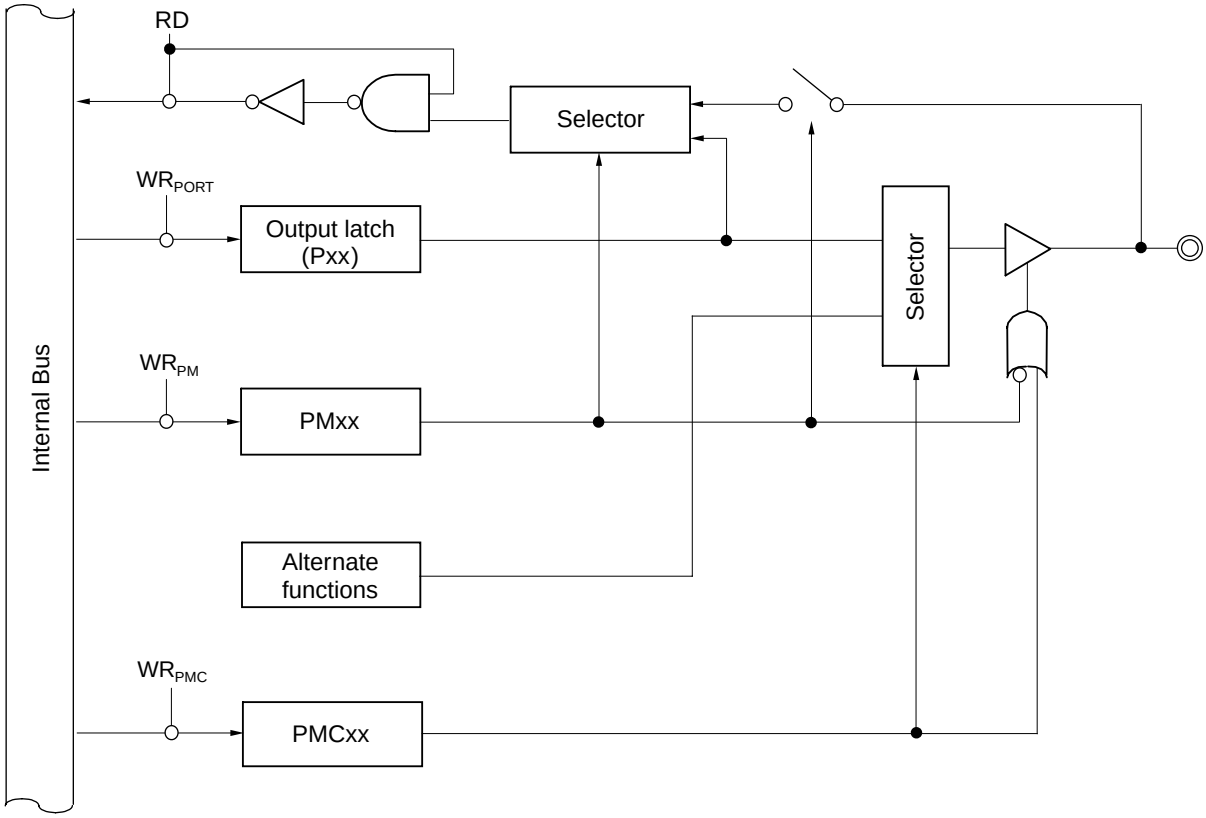


Figure 14-8: Type G Block Diagram

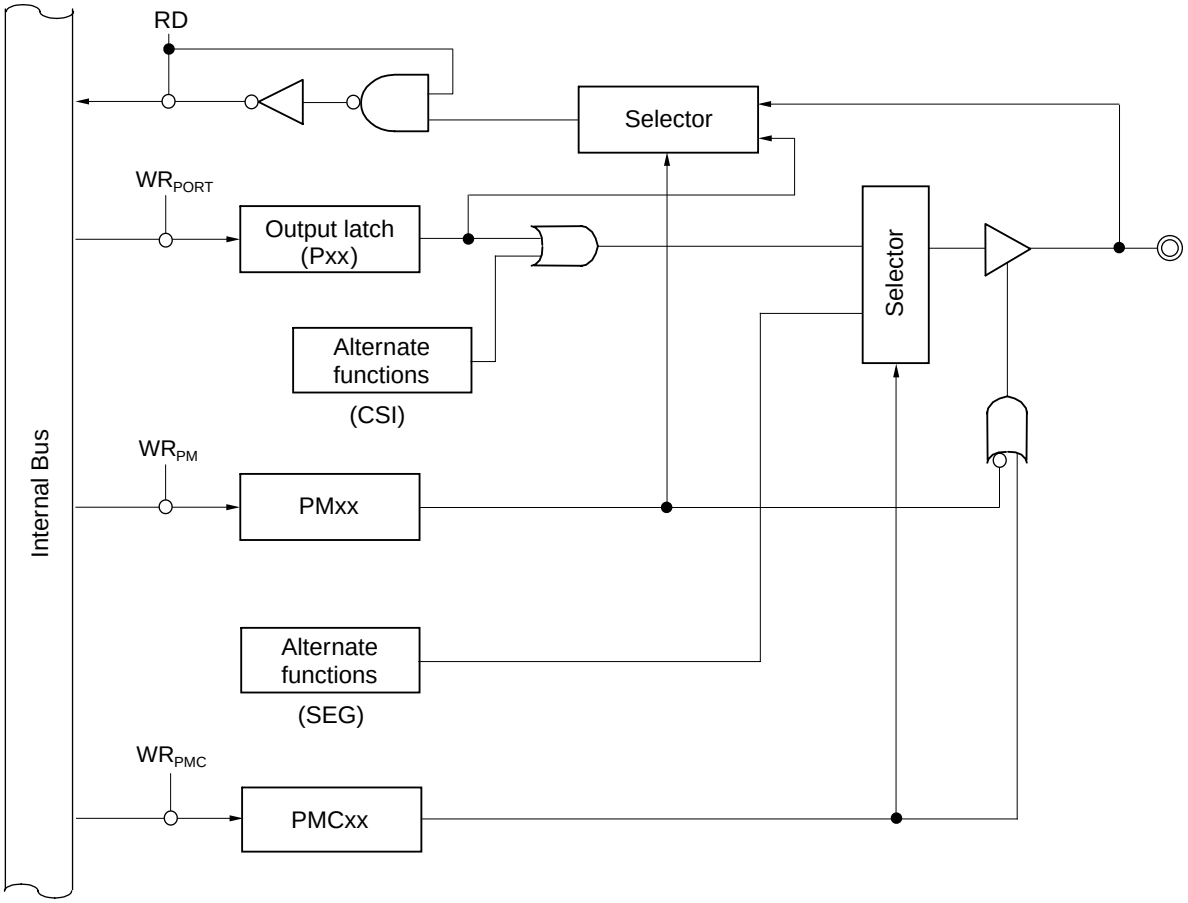


Figure 14-9: Type H Block Diagram

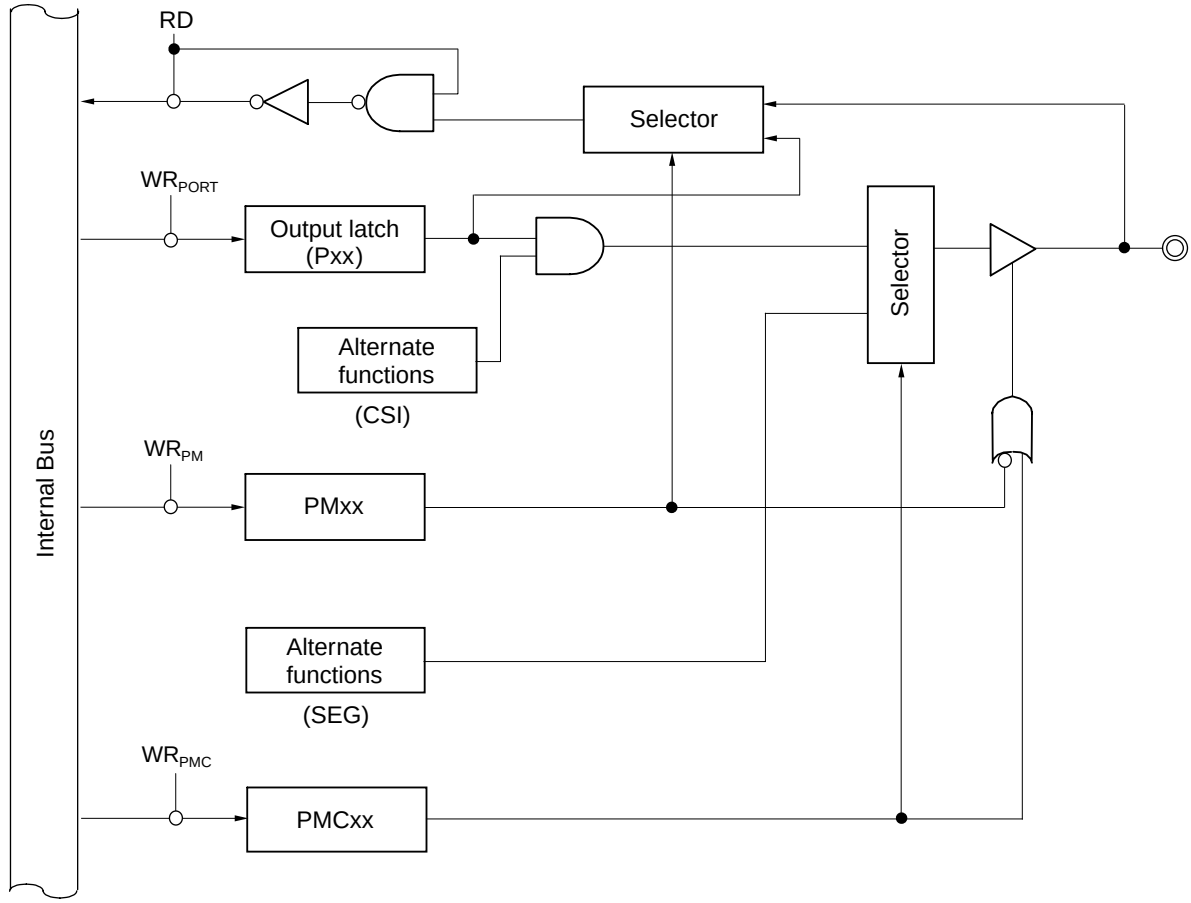
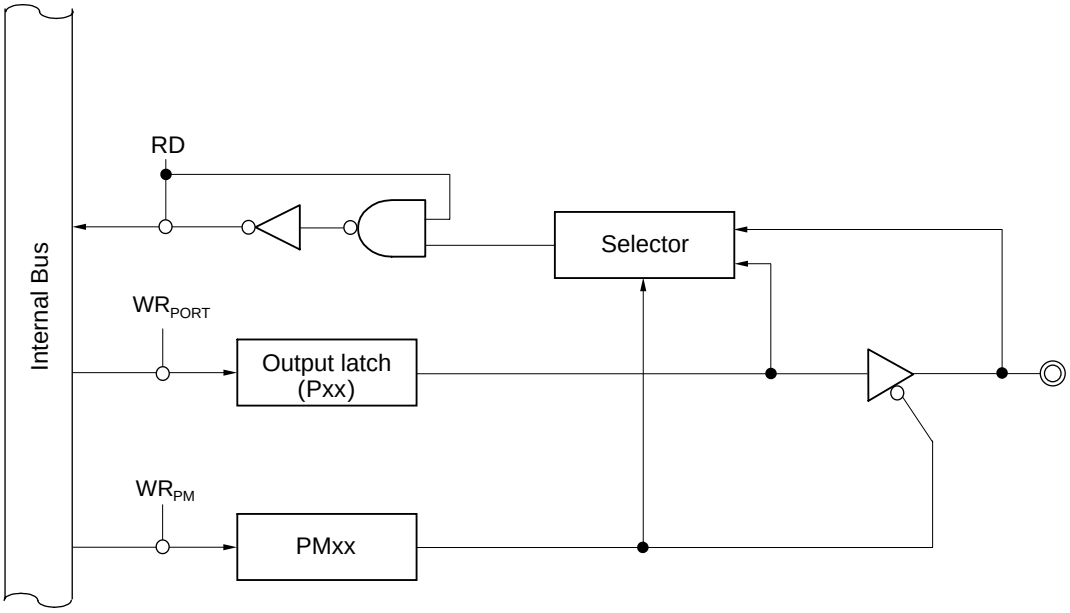


Figure 14-10: Type I Block Diagram



14.3 Pin Functions of Each Port

14.3.1 Port 0

Port 0 is a 5-bit input/output port in which input or output can be specified in 1-bit units.

Figure 14-11: Port 0 (P0)

	7	6	5	4	3	2	1	0	Address	At Reset
P0	0	0	0	P04	P03	P02	P01	P00	FFFF F000H	00H

Bit position	Bit name	Function
4 to 0	P0n (n = 4 to 0)	Input/output port

Besides functioning as a port, in control mode, it also can operate as external interrupt request inputs.

(1) Operation in alternative mode

Port		Alternate Pin Name	Remarks	Block Type
Port 1	P00	NMI	external interrupt request input	A
	P01	INTP0		
	P02	INTP1		
	P03	INTP2		
	P04	INPT3		

(2) Setting in input/output mode

Port 0 is set in input/output mode using the port 0 mode register (PM0).

(a) Port 0 mode register (PM0)

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

Figure 14-12: Port 0 Mode Register (PM0)

	7	6	5	4	3	2	1	0	Address	At Reset
PM0	0	0	0	PM04	PM03	PM02	PM00	PM00	FFFF F020H	1FH

Bit Position	Bit Name	Function
6 to 0	PM0n (n = 6 to 0)	Specifies input/output mode of P0n pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)

(b) Port 0 Pull-up Resistor Option register (PU0)

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

Figure 14-13: Port 0 Pull-up Resistor Option register (PU0)

	7	6	5	4	3	2	1	0	Address	At Reset
PU0	0	0	0	PU04	PU03	PU02	PU00	PU00	FFFF F080H	00H

Bit Position	Bit Name	On-chip Pull-up Resistor Connection Control
4 to 0	PU0n (n = 4 to 0)	Specifies input/output mode of P0n pin. 0: Pull-up Resistor is not connected 1: Pull-up Resistor is connected

14.3.2 Port 1

Port 1 is a 7-bit input/output port in which input or output can be specified in 1-bit units.

Figure 14-14: Port 1 (P1)

	7	6	5	4	3	2	1	0	Address	At Reset
P1	0	P16	P15	P14	P13	P12	P11	P10	FFFF F002H	00H

Bit position	Bit name	Function
6 to 0	P1n (n = 6 to 0)	Input/output port

Besides functioning as a port, in control mode, it also can operate as Serial interface (CSI00, UART50, UART51) input/output and External interrupt input.

(1) Operation in alternative mode

Port		Alternate Pin Name	Remarks	Block Type
Port 1	P10	SI00	Serial interface (CSI00, UART50, UART51) inputs/outputs, External interrupt request input	A
	P11	SO00		B
	P12	$\overline{\text{SCK00}}$		C
	P13	RXD50/INTP4		A
	P14	TXD50		C
	P15	RXD51/INTP5		A
	P16	TXD51		C

(2) Setting in input/output mode

Port 1 is set in input/output mode using the port 1 mode register (PM1). In control mode, it is set using the port 1 mode control register (PMC1).

(a) Port 1 mode register (PM1)

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

Figure 14-15: Port 1 Mode Register (PM1)

	7	6	5	4	3	2	1	0	Address	At Reset
PM1	0	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFFF F022H	7FH

Bit Position	Bit Name	Function
5, 3, 0	PM15, PM13, PM10	Specifies input/output mode of P15, P13, P10 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
6	PM26	Specifies input/output mode of P16 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
4	PM24	Specifies input/output mode of P14 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
2	PM22	Specifies input/output mode of P12 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
1	PM21	Specifies input/output mode of P11 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)

(b) Port 1 Pull-up Resistor Option register (PU1)

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

Figure 14-16: Port 1 Pull-up Resistor Option register (PU1)

	7	6	5	4	3	2	1	0	Address	At Reset
PU1	0	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FFFF F082H	00H

Bit Position	Bit Name	On-chip Pull-up Resistor Connection Control
6 to 0	PU1n (n = 6 to 0)	Specifies input/output mode of P0n pin. 0: Pull-up Resistor is not connected 1: Pull-up Resistor is connected

14.3.3 Port 2

Port 2 is a 8-bit input/output port in which input or output can be specified in 1-bit units.

Figure 14-17: Port 2 (P2)

	7	6	5	4	3	2	1	0	Address	At Reset
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFFF F004H	00H

Bit position	Bit name	Function
7 to 0	P2n (n = 7 to 0)	Input/output port

Besides functioning as a port, in control mode, it also can operate as 16 Bit Timer G inputs/outputs and 8 Bit Timer 50/51 inputs/outputs.

(1) Operation in alternative mode

Port		Alternate Pin Name	Remarks	Block Type
Port 2	P20	TIG0	16 Bit Timer G inputs/outputs, 8 Bit Timer 50/51 inputs/outputs	A
	P21	TIG1/TOG1		B
	P22	TIG2/TOG2		B
	P23	TIG3/TOG3		B
	P24	TIG4/TOG4		B
	P25	TIG5		A
	P26	TI50/TO50		B
	P27	TI51/TO51		B

(2) Setting in input/output mode and control mode

Port 2 is set in input/output mode using the port 2 mode register (PM2).

(a) Port 2 mode register (PM2)

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

Figure 14-18: Port 2 Mode Register (PM2)

	7	6	5	4	3	2	1	0	Address	At Reset
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFFF F024H	FFH

Bit Position	Bit Name	Function
5, 0	PM25, PM20	Specifies input/output mode of P25, P20 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
7	PM27	Specifies input/output mode of P27 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
6	PM26	Specifies input/output mode of P26 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
4	PM24	Specifies input/output mode of P24 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
3	PM23	Specifies input/output mode of P23 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
2	PM22	Specifies input/output mode of P22 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
1	PM21	Specifies input/output mode of P21 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)

(b) Port 2 Pull-up Resistor Option register (PU2)

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

Figure 14-19: Port 2 Pull-up Resistor Option register (PU2)

	7	6	5	4	3	2	1	0	Address	At Reset
PU2	PU27	PU26	PU25	PU24	PU23	PU22	PU21	PU20	FFFF F084H	00H

Bit Position	Bit Name	On-chip Pull-up Resistor Connection Control
7 to 0	PU2n (n = 7 to 0)	Specifies input/output mode of P0n pin. 0: Pull-up Resistor is not connected 1: Pull-up Resistor is connected

14.3.4 Port 3

Port 3 is a 6-bit input/output port in which input or output can be specified in 1-bit units.

Figure 14-20: Port 3 (P3)

	7	6	5	4	3	2	1	0	Address	At Reset
P3	0	0	P35	P34	P33	P32	P31	P30	FFFF F006H	00H

Bit position	Bit name	Function
5 to 0	P3n (n = 5 to 0)	Input/output port

Besides functioning as a port, in control mode, it also can operate as the Processor Clock output or 16 Bit Timer C0/C1 inputs/outputs.

(1) Operation in alternative mode

Port		Alternate Pin Name	Remarks	Block Type
Port 3	P30	PCL	Processor Clock output, 16 Bit Timer C0/C1 inputs/outputs	B
	P31	TIC00/TOC0		B
	P32	TIC10		A
	P33	TIC01/TOC1		B
	P34	TIC11		A
	P35	-		A

(2) Setting in input/output mode and control mode

Port 3 is set in input/output mode using the port 3 mode register (PM3).

(a) Port 3 mode register (PM3)

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

Figure 14-21: Port 3 Mode Register (PM3)

	7	6	5	4	3	2	1	0	Address	At Reset
PM3	0	0	PM35	PM34	PM33	PM32	PM31	PM30	FFFF F026H	3FH

Bit Position	Bit Name	Function
5, 4, 2	PM35, PM34, PM32	Specifies input/output mode of P35, P34, P32 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
3	PM33	Specifies input/output mode of P81 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
1	PM31	Specifies input/output mode of P81 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
0	PM30	Specifies input/output mode of P81 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)

(b) Port 3 Pull-up Resistor Option register (PU3)

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

Figure 14-22: Port 3 Pull-up Resistor Option register (PU3)

	7	6	5	4	3	2	1	0	Address	At Reset
PU3	0	0	PU35	PU34	PU33	PU32	PU31	PU30	FFFF F086H	00H

Bit Position	Bit Name	On-chip Pull-up Resistor Connection Control
5 to 0	PU3n (n = 5 to 0)	Specifies input/output mode of P0n pin. 0: Pull-up Resistor is not connected 1: Pull-up Resistor is connected

14.3.5 Port 4

Port 4 is a 8-bit input/output port in which input or output can be specified in 1-bit units.

Figure 14-23: Port 4 (P4)

	7	6	5	4	3	2	1	0	Address	At Reset
P4	P47	P46	P45	P44	P43	P42	P41	P40	FFFF F008H	00H

Bit position	Bit name	Function
5 to 0	P4n (n = 5 to 0)	Input/output port

Besides functioning as a port, in control mode, it also can operate as the LCD Segment output.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 4	P40	COM0	LCD Segment output	E
	P41	COM1		
	P42	COM2		
	P43	COM3		
	P44	SEG0		
	P45	SEG1		
	P46	SEG2		
	P47	SEG3		

(2) Setting in input/output mode and control mode

Port 4 is set in input/output mode using the port 4 mode register (PM4). In control mode, it is set using the port 4 mode control register (PMC4).

(a) Port 4 mode register (PM4)

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

Figure 14-24: Port 4 Mode Register (PM4)

	7	6	5	4	3	2	1	0	Address	At Reset
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFFF F028H	FFH

Bit Position	Bit Name	Function
5 to 0	PM4n (n = 5 to 0)	Specifies input/output mode of P4n pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)

(b) Port 4 mode control register (PMC4)

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

Figure 14-25: Port 4 Mode Control Register (PMC4)

	7	6	5	4	3	2	1	0	Address	At Reset
PMC4	PMC47	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40	FFFF F0E8H	00H

Bit Position	Bit Name	Function
7	PMC47	Specifies operation mode of P47 pin 0: Input/output port mode 1: LCD SEG3 output mode
6	PMC46	Specifies operation mode of P46 pin 0: Input/output port mode 1: LCD SEG2 output mode
5	PMC45	Specifies operation mode of P45 pin 0: Input/output port mode 1: LCD SEG1 output mode
4	PMC44	Specifies operation mode of P44 pin 0: Input/output port mode 1: LCD SEG0 output mode
3	PMC43	Specifies operation mode of P43 pin 0: Input/output port mode 1: LCD COM3 output mode
2	PMC42	Specifies operation mode of P42 pin 0: Input/output port mode 1: LCD COM2 output mode
1	PMC41	Specifies operation mode of P41 pin 0: Input/output port mode 1: LCD COM1 output mode
0	PMC40	Specifies operation mode of P40 pin 0: Input/output port mode 1: LCD COM0 output mode

14.3.6 Port 5

Port 5 is a 8-bit input/output port in which input or output can be specified in 1-bit units.

Figure 14-26: Port 5 (P5)

	7	6	5	4	3	2	1	0	Address	At Reset
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFFF F00AH	00H

Bit position	Bit name	Function
7 to 0	P5n (n = 7 to 0)	Input/output port

Besides functioning as a port, in control mode, it also can operate as the LCD Segment output.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 5	P50	SEG4	LCD Segment output	E
	P51	SEG5		
	P52	SEG6		
	P53	SEG7		
	P54	SEG8		
	P55	SEG9		
	P56	SEG10		
	P57	SEG11		

(2) Setting in input/output mode and control mode

Port 5 is set in input/output mode using the port 5 mode register (PM5). In control mode, it is set using the port 5 mode control register (PMC5).

(a) Port 5 mode register (PM5)

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

Figure 14-27: Port 5 Mode Register (PM5)

	7	6	5	4	3	2	1	0	Address	At Reset
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFFF F02AH	FFH

Bit Position	Bit Name	Function
7 to 0	PM5n (n = 7 to 0)	Specifies input/output mode of P5n pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)

(b) Port 5 mode control register (PMC5)

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

Figure 14-28: Port 5 Mode Control Register (PMC5)

	7	6	5	4	3	2	1	0	Address	At Reset
PMC5	PMC57	PMC56	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50	FFFF F0EAH	00H

Bit Position	Bit Name	Function
7	PMC57	Specifies operation mode of P57 pin 0: Input/output port mode 1: LCD SEG11 output mode
6	PMC56	Specifies operation mode of P56 pin 0: Input/output port mode 1: LCD SEG10 output mode
5	PMC55	Specifies operation mode of P55 pin 0: Input/output port mode 1: LCD SEG9 output mode
4	PMC54	Specifies operation mode of P54 pin 0: Input/output port mode 1: LCD SEG8 output mode
3	PMC53	Specifies operation mode of P53 pin 0: Input/output port mode 1: LCD SEG7 output mode
2	PMC52	Specifies operation mode of P52 pin 0: Input/output port mode 1: LCD SEG6 output mode
1	PMC51	Specifies operation mode of P51 pin 0: Input/output port mode 1: LCD SEG5 output mode
0	PMC50	Specifies operation mode of P50 pin 0: Input/output port mode 1: LCD SEG4 output mode

14.3.7 Port 6

Port 6 is a 6-bit input/output port in which input or output can be specified in 1-bit units.

Figure 14-29: Port 6 (P6)

	7	6	5	4	3	2	1	0	Address	At Reset
P6	0	0	P65	P64	P63	P62	P61	P60	FFFF F00CH	00H

Bit position	Bit name	Function
5 to 0	P6n (n = 5 to 0)	Input/output port

Besides functioning as a port, in control mode, it also can operate as the LCD Segment output.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 6	P60	SEG12	LCD Segment output	E
	P61	SEG13		
	P62	SEG14		
	P63	SEG15		
	P64	SEG16		
	P65	SEG17		

(2) Setting in input/output mode and control mode

Port 6 is set in input/output mode using the port 6 mode register (PM6). In control mode, it is set using the port 6 mode control register (PMC6).

(a) Port 6 mode register (PM6)

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

Figure 14-30: Port 6 Mode Register (PM6)

	7	6	5	4	3	2	1	0	Address	At Reset
PM6	0	0	PM65	PM64	PM63	PM62	PM61	PM60	FFFF F02CH	3FH

Bit Position	Bit Name	Function
5 to 0	PM6n (n = 5 to 0)	Specifies input/output mode of P6n pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)

(b) Port 6 mode control register (PMC6)

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

Figure 14-31: Port 6 Mode Control Register (PMC6)

	7	6	5	4	3	2	1	0	Address	At Reset
PMC6	0	0	PMC65	PMC64	PMC63	PMC62	PMC61	PMC60	FFFF F0ECH	00H

Bit Position	Bit Name	Function
5	PMC65	Specifies operation mode of P65 pin 0: Input/output port mode 1: LCD SEG17 output mode
4	PMC64	Specifies operation mode of P64 pin 0: Input/output port mode 1: LCD SEG16 output mode
3	PMC63	Specifies operation mode of P63 pin 0: Input/output port mode 1: LCD SEG15 output mode
2	PMC62	Specifies operation mode of P62 pin 0: Input/output port mode 1: LCD SEG14 output mode
1	PMC61	Specifies operation mode of P61 pin 0: Input/output port mode 1: LCD SEG13 output mode
0	PMC60	Specifies operation mode of P60 pin 0: Input/output port mode 1: LCD SEG12 output mode

14.3.8 Port 7

Port 7 is a 8-bit input port in which input can be specified in 1-bit units.

Figure 14-32: Port 7 (P7)

	7	6	5	4	3	2	1	0	Address	At Reset
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFFF F00EH	00H

Bit position	Bit name	Function
7 to 0	P7n (n = 5 to 0)	Input port

Besides functioning as a input port, in control mode, it also can operate as the Analog/Digital Converter inputs.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 7	P70	ANI0	Analog/Digital Converter inputs	D
	P71	ANI1		
	P72	ANI2		
	P73	ANI3		
	P74	ANI4		
	P75	ANI5		
	P76	ANI6		
	P77	ANI7		

(2) Setting in input/output mode and control mode

Port 7 is set in input mode at any time.

14.3.9 Port 8

Port 8 is a 8-bit input/output port in which input or output can be specified in 1-bit units.

Figure 14-33: Port 8 (P8)

	7	6	5	4	3	2	1	0	Address	At Reset
P8	P87	P86	P85	P84	P83	P82	P81	P80	FFFF F010H	00H

Bit position	Bit name	Function
7 to 0	P8n (n = 7 to 0)	Input/output port

Besides functioning as a port, in control mode, it also can operate as the serial interface (CSI00, CSI01, UART50) input/output.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 8	P80	SI01	Serial interface (CSI01, DCAN0, DCAN1) input/output.	A
	P81	SO01		B
	P82	$\overline{\text{SCK01}}$		C
	P83	CRXD10		A
	P84	CRTD10		C
	P85	CRXD11		A
	P86	CRTD11		C
	P87	-		A

(2) Setting in input/output mode and control mode

Port 8 is set in input/output mode using the port 8 mode register (PM8).

(a) Port 8 mode register (PM8)

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

Figure 14-34: Port 8 Mode Register (PM8)

	7	6	5	4	3	2	1	0	Address	At Reset
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FFFF F030H	FFH

Bit Position	Bit Name	Function
7, 5, 3, 0	PM87, PM85, PM83, PM80	Specifies input/output mode of P87, P85, P83, P80 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
6	PM86	Specifies input/output mode of P86 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
4	PM84	Specifies input/output mode of P84 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
2	PM82	Specifies input/output mode of P82 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
1	PM81	Specifies input/output mode of P81 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)

(b) Port 8 Pull-up Resistor Option register (PU8)

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

Figure 14-35: Port 8 Pull-up Resistor Option register (PU8)

	7	6	5	4	3	2	1	0	Address	At Reset
PU8	PU87	PU86	PU85	PU84	PU83	PU82	PU81	PU80	FFFF F090H	00H

Bit Position	Bit Name	On-chip Pull-up Resistor Connection Control
7 to 0	PU8n (n = 7 to 0)	Specifies input/output mode of P0n pin. 0: Pull-up Resistor is not connected 1: Pull-up Resistor is connected

14.3.10 Port 9

Port 9 is a 7-bit input/output port in which input or output can be specified in 1-bit units.

Figure 14-36: Port 9 (P9)

	7	6	5	4	3	2	1	0	Address	At Reset
P9	0	P96	P99	P94	P93	P92	P91	P90	FFFF F012H	00H

Bit position	Bit name	Function
6 to 0	P9n (n = 7 to 0)	Input/output port

Besides functioning as a port, in control mode, it also can operate as the LCD Segment output.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 9	P90	SEG18	LCD Segment output	E
	P91	SEG19		
	P92	SEG20		
	P93	SEG21		
	P94	SEG22		
	P95	SEG23		
	P96	SEG24		

(2) Setting in input/output mode and control mode

Port 9 is set in output/output-disable mode using the port 9 mode register (PM9). In control mode, it is set using the port 9 mode control register (PMC9).

(a) Port 9 mode register (PM9)

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

Figure 14-37: Port 9 Mode Register (PM9)

	7	6	5	4	3	2	1	0	Address	At Reset
PM9	0	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FFFF F032H	7FH

Bit Position	Bit Name	Function
6 to 0	PM9n (n = 6 to 0)	Specifies input/output mode of P9n pin. 0: Output enable (Output buffer on) 1: Output disable (Output buffer off, high Z)

(b) Port 9 mode control register (PMC9)

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

Figure 14-38: Port 9 Mode Control Register (PMC9)

	7	6	5	4	3	2	1	0	Address	At Reset
PMC9	0	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90	FFFF F0F2H	00H

Bit Position	Bit Name	Function
6	PMC96	Specifies operation mode of P96 pin 0: Input/output port mode 1: LCD SEG24 output mode
5	PMC95	Specifies operation mode of P95 pin 0: Input/output port mode 1: LCD SEG23 output mode
4	PMC94	Specifies operation mode of P94 pin 0: Input/output port mode 1: LCD SEG22 output mode
3	PMC93	Specifies operation mode of P93 pin 0: Input/output port mode 1: LCD SEG21 output mode
2	PMC92	Specifies operation mode of P92 pin 0: Input/output port mode 1: LCD SEG20 output mode
1	PMC91	Specifies operation mode of P91 pin 0: Input/output port mode 1: LCD SEG19 output mode
0	PMC90	Specifies operation mode of P90 pin 0: Input/output port mode 1: LCD SEG18 output mode

14.3.11 Port 10

Port 10 is a 8-bit output port in which the output mode can be specified in 1-bit units.

Figure 14-39: Port 10 (P10)

	7	6	5	4	3	2	1	0	Address	At Reset
P10	P107	P106	P105	P104	P103	P102	P101	P100	FFFF F014H	00H

Bit position	Bit name	Function
7 to 0	P10n (n = 7 to 0)	output port

Besides functioning as a port, in control mode, it also can operate as the Meter-PWM output.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 10	P100	SM11	Meter-PWM output	F
	P101	SM12		
	P102	SM13		
	P103	SM14		
	P104	SM21		
	P105	SM22		
	P106	SM23		
	P107	SM24		

(2) Setting output mode and control mode

Port 10 is set in output mode using the port 10 mode register (PM10). In control mode, it is set using the port 10 mode control register (PMC10).

(a) Port 10 mode register (PM10)

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

Figure 14-40: Port 10 Mode Register (PM10)

	7	6	5	4	3	2	1	0	Address	At Reset
PM10	P107	PM106	PM105	PM104	PM103	PM102	PM101	PM100	FFFF F034H	FFH

Bit Position	Bit Name	Function
7 to 0	PM10n (n = 7 to 0)	Specifies output mode of P10n pin. 0: Output enable (Output buffer on) 1: Output disable (Output buffer off, high Z)

(b) Port 10 mode control register (PMC10)

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

Figure 14-41: Port 10 Mode Control Register (PMC10)

	7	6	5	4	3	2	1	0	Address	At Reset
PMC10	PMC107	PMC106	PMC105	PMC104	PMC103	PMC102	PMC101	PMC100	FFFF F0F4H	00H

Bit Position	Bit Name	Function
7	PMC107	Specifies operation mode of P107 pin 0: Input/output port mode 1: Meter SM24 output mode
6	PMC106	Specifies operation mode of P106 pin 0: Input/output port mode 1: Meter SM23 output mode
5	PMC105	Specifies operation mode of P105 pin 0: Input/output port mode 1: Meter SM22 output mode
4	PMC104	Specifies operation mode of P104 pin 0: Input/output port mode 1: Meter SM21 output mode
3	PMC103	Specifies operation mode of P103 pin 0: Input/output port mode 1: Meter SM14 output mode
2	PMC102	Specifies operation mode of P102 pin 0: Input/output port mode 1: Meter SM13 output mode
1	PMC101	Specifies operation mode of P101 pin 0: Input/output port mode 1: Meter SM12 output mode
0	PMC100	Specifies operation mode of P100 pin 0: Input/output port mode 1: Meter SM11 output mode

14.3.12 Port 11

Port 11 is a 8-bit output port in which the output mode can be specified in 1-bit units.

Figure 14-42: Port 11 (P11)

	7	6	5	4	3	2	1	0	Address	At Reset
P11	P117	P116	P115	P114	P113	P112	P111	P110	FFFF F016H	00H

Bit position	Bit name	Function
7 to 0	P11n (n = 7 to 0)	output port

Besides functioning as a port, in control mode, it also can operate as the Meter-PWM output.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 11	P110	SM31	Meter-PWM output	F
	P111	SM32		
	P112	SM33		
	P113	SM34		
	P114	SM41		
	P115	SM42		
	P116	SM43		
	P117	SM44		

(2) Setting in output mode and control mode

Port 11 is set in output enable/disable mode using the port 11 mode register (PM11). In control mode, it is set using the port 11 mode control register (PMC11).

(a) Port 11 mode register (PM11)

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

Figure 14-43: Port 11 Mode Register (PM11)

	7	6	5	4	3	2	1	0	Address	At Reset
PM11	P117	PM116	PM115	PM114	PM113	PM112	PM111	PM110	FFFF F036H	FFH

Bit Position	Bit Name	Function
7 to 0	PM11n (n = 7 to 0)	Specifies output mode of P11n pin. 0: Output enable (Output buffer on) 1: Output disable (Output buffer off, high Z)

(b) Port 11 mode control register (PMC11)

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

Figure 14-44: Port 11 Mode Control Register (PMC11)

	7	6	5	4	3	2	1	0	Address	At Reset
PMC11	PMC117	PMC116	PMC115	PMC114	PMC113	PMC112	PMC111	PMC110	FFFF F0F6H	00H

Bit Position	Bit Name	Function
7	PMC117	Specifies operation mode of P117 pin 0: Input/output port mode 1: Meter SM44 output mode
6	PMC116	Specifies operation mode of P116 pin 0: Input/output port mode 1: Meter SM43 output mode
5	PMC115	Specifies operation mode of P115 pin 0: Input/output port mode 1: Meter SM42 output mode
4	PMC114	Specifies operation mode of P114 pin 0: Input/output port mode 1: Meter SM41 output mode
3	PMC113	Specifies operation mode of P113 pin 0: Input/output port mode 1: Meter SM34 output mode
2	PMC112	Specifies operation mode of P112 pin 0: Input/output port mode 1: Meter SM33 output mode
1	PMC111	Specifies operation mode of P111 pin 0: Input/output port mode 1: Meter SM32 output mode
0	PMC110	Specifies operation mode of P110 pin 0: Input/output port mode 1: Meter SM31 output mode

14.3.13 Port 12

Port 12 is a 8-bit input/output port in which input or output can be specified in 1-bit units.

Figure 14-45: Port 12 (P12)

	7	6	5	4	3	2	1	0	Address	At Reset
P12	P127	P126	P125	P124	P123	P122	P121	P120	FFFF F018H	00H

Bit position	Bit name	Function
7 to 0	P12n (n = 7 to 0)	Input/output port

Besides functioning as a port, in control mode, it also can operate as the Meter-PWM output.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 12	P120	SM51	Meter-PWM output	E
	P121	SM52		
	P122	SM53		
	P123	SM54		
	P124	SM61		
	P125	SM62		
	P126	SM63		
	P127	SM64		

(2) Setting in input/output mode and control mode

Port 12 is set in input/output mode using the port 12 mode register (PM12). In control mode, it is set using the port 12 mode control register (PMC12).

(a) Port 12 mode register (PM12)

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

Figure 14-46: Port 12 Mode Register (PM12)

	7	6	5	4	3	2	1	0	Address	At Reset
PM12	P127	PM126	PM125	PM124	PM123	PM122	PM121	PM120	FFFF F038H	FFH

Bit Position	Bit Name	Function
7 to 0	PM12n (n = 7 to 0)	Specifies input/output mode of P12n pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)

(b) Port 12 mode control register (PMC12)

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

Figure 14-47: Port 12 Mode Control Register (PMC12)

	7	6	5	4	3	2	1	0	Address	At Reset
PMC12	PMC127	PMC126	PMC125	PMC124	PMC123	PMC122	PMC121	PMC120	FFFF F0F8H	00H

Bit Position	Bit Name	Function
7	PMC127	Specifies operation mode of P127 pin 0: Input/output port mode 1: Meter SM64 output mode
6	PMC126	Specifies operation mode of P126 pin 0: Input/output port mode 1: Meter SM63 output mode
5	PMC125	Specifies operation mode of P125 pin 0: Input/output port mode 1: Meter SM62 output mode
4	PMC124	Specifies operation mode of P124 pin 0: Input/output port mode 1: Meter SM61 output mode
3	PMC123	Specifies operation mode of P123 pin 0: Input/output port mode 1: Meter SM54 output mode
2	PMC122	Specifies operation mode of P122 pin 0: Input/output port mode 1: Meter SM53 output mode
1	PMC121	Specifies operation mode of P121 pin 0: Input/output port mode 1: Meter SM52 output mode
0	PMC120	Specifies operation mode of P120 pin 0: Input/output port mode 1: Meter SM51 output mode

14.3.14 Port 13

Port 13 is a 8-bit input/output port in which input or output can be specified in 1-bit units.

Figure 14-48: Port 13 (P13)

	7	6	5	4	3	2	1	0	Address	At Reset
P13	P137	P136	P135	P134	P133	P132	P131	P130	FFFF F01AH	00H

Bit position	Bit name	Function
7 to 0	P13n (n = 7 to 0)	Input/output port

Besides functioning as a port, in control mode, it also can operate as the LCD Segment output.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 13	P130	SEG25	LCD Segment output	E
	P131	SEG26		
	P132	SEG27		
	P133	SEG28		
	P134	SEG29		
	P135	SEG30		
	P136	SEG31		
	P137	SEG32		

(2) Setting in input/output mode and control mode

Port 13 is set in input/output mode using the port 13 mode register (PM13). In control mode, it is set using the port 13 mode control register (PMC13).

(a) Port 13 mode register (PM13)

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

Figure 14-49: Port 13 Mode Register (PM13)

	7	6	5	4	3	2	1	0	Address	At Reset
PM13	P137	PM136	PM135	PM134	PM133	PM132	PM131	PM130	FFFF F03AH	FFH

Bit Position	Bit Name	Function
7 to 0	PM13n (n = 7 to 0)	Specifies input/output mode of P13n pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)

(b) Port 13 mode control register (PMC13)

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

Figure 14-50: Port 13 Mode Control Register (PMC13)

	7	6	5	4	3	2	1	0	Address	At Reset
PMC13	PMC137	PMC136	PMC135	PMC134	PMC133	PMC132	PMC131	PMC130	FFFF F0FAH	00H

Bit Position	Bit Name	Function
7	PMC137	Specifies operation mode of P137 pin 0: Input/output port mode 1: LCD SEG32 output mode
6	PMC136	Specifies operation mode of P136 pin 0: Input/output port mode 1: LCD SEG31 output mode
5	PMC135	Specifies operation mode of P135 pin 0: Input/output port mode 1: LCD SEG30 output mode
4	PMC134	Specifies operation mode of P134 pin 0: Input/output port mode 1: LCD SEG29 output mode
3	PMC133	Specifies operation mode of P133 pin 0: Input/output port mode 1: LCD SEG28 output mode
2	PMC132	Specifies operation mode of P132 pin 0: Input/output port mode 1: LCD SEG27 output mode
1	PMC131	Specifies operation mode of P131 pin 0: Input/output port mode 1: LCD SEG26 output mode
0	PMC130	Specifies operation mode of P130 pin 0: Input/output port mode 1: LCD SEG25 output mode

14.3.15 Port 14

Port 14 is a 4-bit input/output port in which input or output can be specified in 1-bit units.

Figure 14-51: Port 14 (P14)

	7	6	5	4	3	2	1	0	Address	At Reset
P14	0	0	0	0	P143	P142	P141	P140	FFFF F01CH	00H

Bit position	Bit name	Function
7 to 0	P14n (n = 3 to 0)	Input/output port

Besides functioning as a port, in control mode, it also can operate as the Serial interface input/output (CSI02) or LCD Segment output.

(1) Operation in control mode

Port		Alternate Pin Name	Remarks	Block Type
Port 14	P140	SI02/SEG33	Serial interface input/output (CSI02), LCD Segment output	E
	P141	SO02/SEG34		G
	P142	$\overline{\text{SCK02}}$/SEG35		H
	P143	-		I

(2) Setting in input/output mode and control mode

Port 14, bit 0 to 3 are set in input/output mode using the port 14 mode register (PM14). In control mode, only the bits 0 to 2 are set using the port 14 mode control register (PMC14).

(a) Port 14 mode register (PM14)

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

Figure 14-52: Port 14 Mode Register (PM14)

	7	6	5	4	3	2	1	0	Address	At Reset
PM14	0	0	0	0	PM143	PM142	PM141	PM140	FFFF F03CH	0FH

Bit Position	Bit Name	Function
0,3	PM140, PM143	Specifies input/output mode of P14n pin (n = 0, 3). 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
1	PM141	Specifies input/output mode of P141 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)
2	PM142	Specifies input/output mode of P141 pin. 0: Output mode (Output buffer on) 1: Input mode (Output buffer off)

(b) Port 14 mode control register (PMC14)

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

Figure 14-53: Port 14 Mode Control Register (PMC14)

	7	6	5	4	3	2	1	0	Address	At Reset
PMC14	0	0	0	0	0	PMC142	PMC141	PMC140	FFFF F0FCH	00H

Bit Position	Bit Name	Function
2	PMC142	Specifies operation mode of P142 pin 0: Input/output port mode 1: LCD SEG35 output mode
1	PMC141	Specifies operation mode of P141 pin 0: Input/output port mode 1: LCD SEG34 output mode
0	PMC140	Specifies operation mode of P140 pin 0: Input/output port mode 1: LCD SEG33 output mode

[MEMO]

Chapter 15 RESET Function

When a low level is input to the RESET pin, there is a system reset and each hardware item of the V850/DB1 is initialized to its initial status.
When the RESET pin changes from low level to high level, reset status is released and the CPU starts program execution. The user has to initialize the contents of various registers as needed within the program.

15.1 Features

- Noise elimination of reset pin (RESET) using analog delay (approximately 500 ns)

15.2 Pin Functions

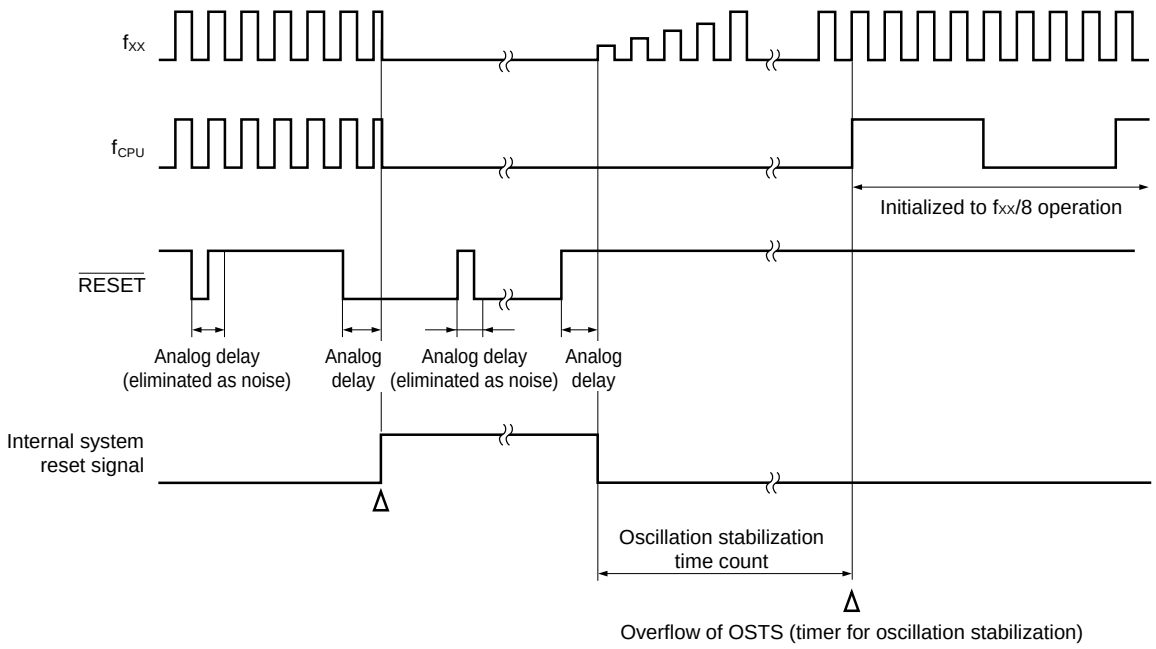
During a system reset period, the operation status of each Port-pin during a reset period is High Impedance.

(Except PCL, $\overline{\text{RESET}}$, X1, X2, V_{DDn}, V_{SSn}, V_{LCD}, SMV_{DDm}, SMV_{SSm}, REGC, V_{PP}/IC, AV_{DD}, AV_{REF}, and AV_{SS}).

Similarly, perform pin processing so that on-chip peripheral I/O function signal output and output ports are not affected.

(1) Reset signal acknowledgment

Figure 15-1: Reset signal acknowledgment

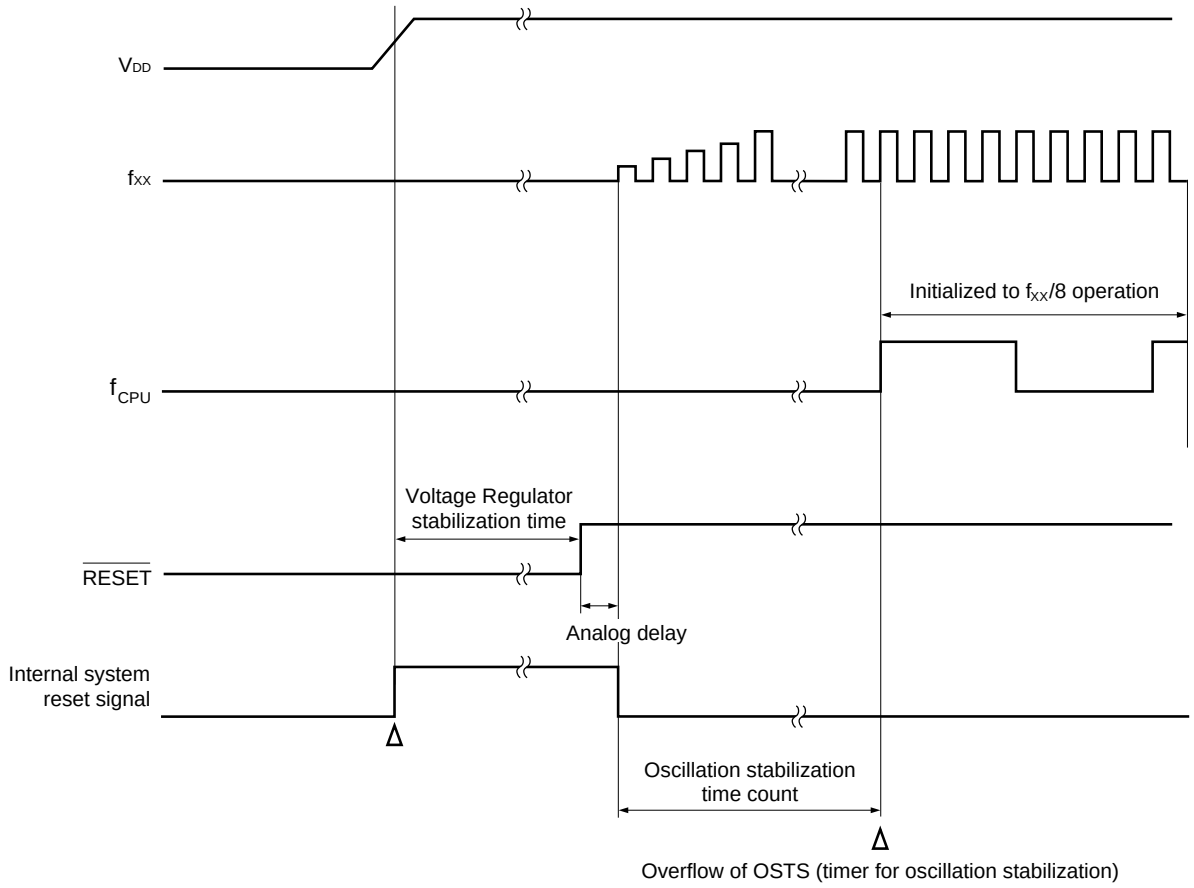


Remark: The internal system reset signal continues in active status for a period of at least 4 system clocks after the timing of a reset release by the $\overline{\text{RESET}}$ signal.

(2) Reset at power-on

The device guarantees the oscillation stabilization time from power-on until reset acknowledgment. Due to the RESET setting (0x04) of the OSTS register the oscillation stabilization time is 8.19 ms.
Nevertheless an external $\overline{\text{RESET}}$ high delay after V_{DD} has been applied, has to be performed by the user. This time is used for stabilization of the internal Voltage Regulator, to guarantee a right CPU performance.

Figure 15-2: Reset at power-on



Caution: The external $\overline{\text{RESET}}$ high delay has to be performed by the user.

15.3 Initialization

Initialize the contents of each register as needed within a program.
Table 15-1 shows the initial values of the CPU and internal RAM after reset. The initial values of on-chip peripheral I/O's after reset can be found in Table 15-1.

Table 15-1: Initial Values of CPU and Internal RAM After Reset

On-Chip Hardware		Register Name	Initial Value After Reset
CPU	Program registers	General-purpose register (r0)	00000000H
		General-purpose registers (r1 to r31)	Undefined
		Program counter (PC)	00000000H
	System registers	Status save registers during interrupt (EIPC, EIPSW)	Undefined
		Status save registers during NMI (FEPC, FEPSW)	Undefined
		Exception cause register (ECR): FECC EICC	 0000H 0000H
		Program status word (PSW)	00000020H
Internal RAM		-	Undefined

Caution: In the table above, "Undefined" means either undefined at the time of a power-on reset or undefined due to data destruction when $\overline{\text{RESET}}$ ↓ input and data write timing are synchronized. On a $\overline{\text{RESET}}$ ↓ other than this, data is maintained in its previous status.

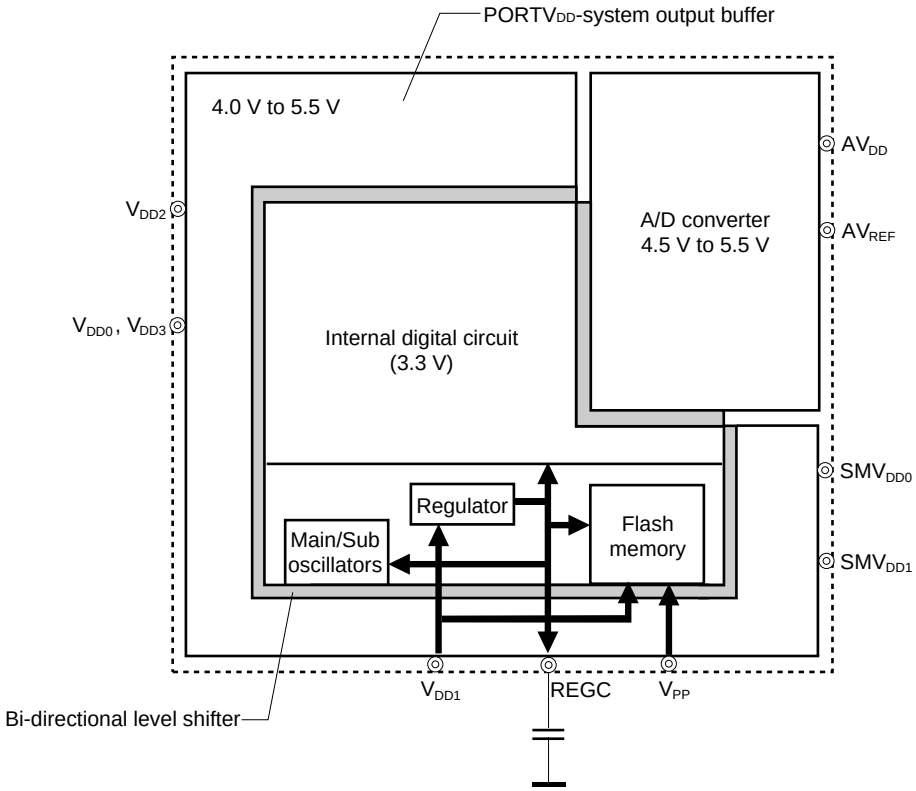
[MEMO]

Chapter 16 Regulator

16.1 Outline

The V850/DB1 incorporates a regulator to realize a 5 V single power supply, low power consumption, and to reduce noise.
This regulator supplies a voltage obtained by stepping down V_{DD1} power supply voltage to oscillation blocks and on-chip logic circuits (excluding the A/D converter and output buffers). The regulator output voltage is set to 3.3 V.
Refer to Chapter 2.3 “Types of Pin I/O Circuit and Connection of Unused Pin” on page 55 for the power supply corresponding to each pin.

Figure 16-1: Regulator



16.2 Operation

The regulator of V850/DB1operates in every mode (HALT, WATCH, STOP).
For stabilization of regulator outputs, connect an capacitor to the REGC pin.
For the value of the capacitor please refer to the Electrical Data Sheet.

[MEMO]

Chapter 17 Flash Memory

The V850/DB1 provides a 128 K flash memory. An instruction fetch from the flash memory takes one clock.
The flash memory can be programmed using a dedicated flash writer. Furthermore this product has a Selfprogramming mode, which allows to program the flash memory by control of the application without any dedicated writer.

The following can be considered as the development environment and the application using a flash memory:

- Software can be altered after the µPD70F3080 is solder mounted on the target system.
- Small scale production of various models is made easier by differentiating software.
- Data adjustment in starting mass production is made easier.
- Alter the software in the field using the Selfprogramming option.

17.1 Features

- 4-byte (1-word) access in 1 clock (instruction fetch access)
- Entire flash memory is divided into 2 areas, which can be erased separately
  - Area 0: 64 K
  - Area 1: 64 K
- Communication through serial interfaces (CSI00 and UART50) from the dedicated flash writer
- Erase/write voltage: V_{PP} = 7.8 V
- On-board programming using flash writer
- Selfprogramming mode
- After erase flash memory becomes FFFF FFFFH

17.2 Writing by Flash writer

Writing can be performed either on-board or off-board by the dedicated flash writer.

(1) On-board programming

The contents of the flash memory is rewritten after the µPD70F3080 is mounted on the target system. It has to be ensured that the signals required for programming are made available to the flash writer.

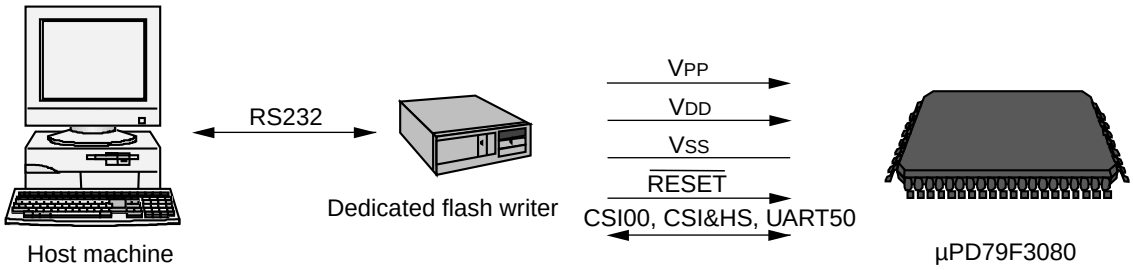
(2) Off-board programming

Writing to a flash memory is performed using a dedicated programming adapter (PA board), etc., before mounting the µPD70F3080 onto the target system.

17.3 Programming Environment

The following diagram shows the environment required for writing programs to the flash memory.

Figure 17-1: Programming Environment in Conjunction with External Flash Writer



A host machine can be used to control the flash writer.
CSI00 is used as the interface between the flash writer and the µPD70F3080 to perform writing, erasing, etc. A programming adapter board is required for off-board writing.

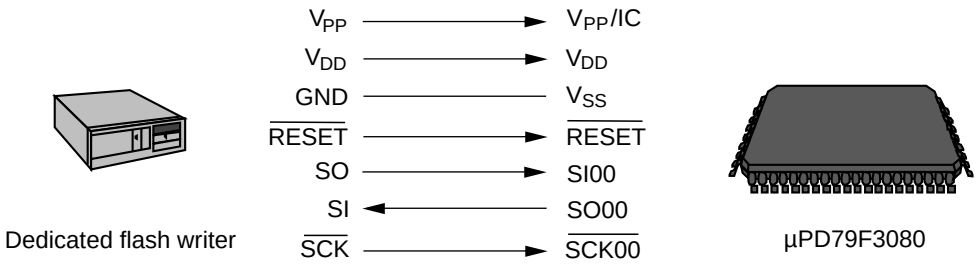
17.4 Communication System

The communication between the dedicated flash writer and the µPD70F3080 is performed by serial communication using CSI.

(1) CSI

Transfer rate: up to 1.0 MHz (MSB first)

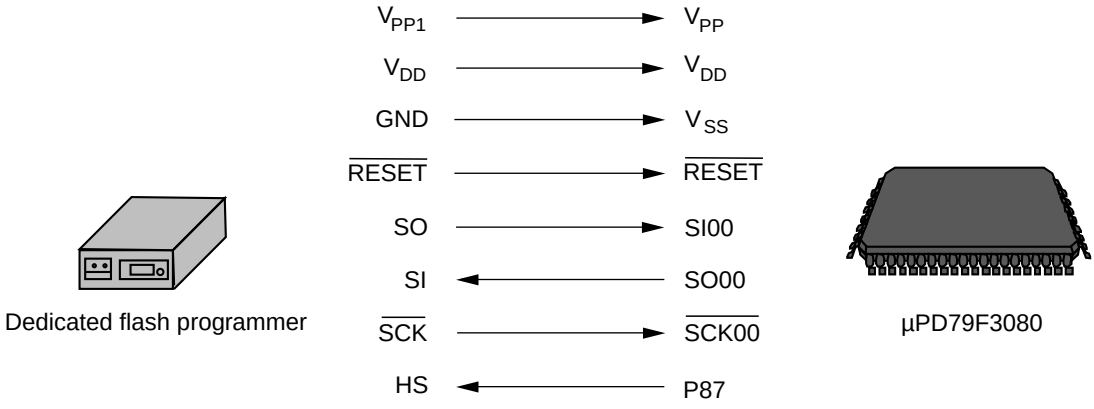
Figure 17-2: Flash Writer Communication via CSI00



(2) Handshake-supported CSI communication

Transfer rate: up to 2 MHz (MSB first)

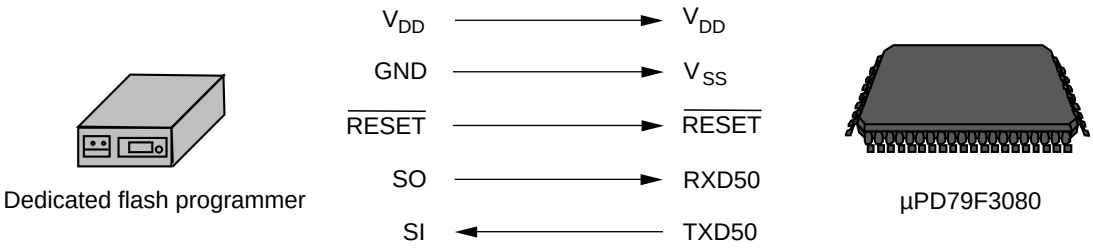
Figure 17-3: Handshake-Supported CSI Communication



(3) UART

Transfer rate: 4,800 bps to 76,800 bps (LSB first)

Figure 17-4: Flash Writer Communication via UART50

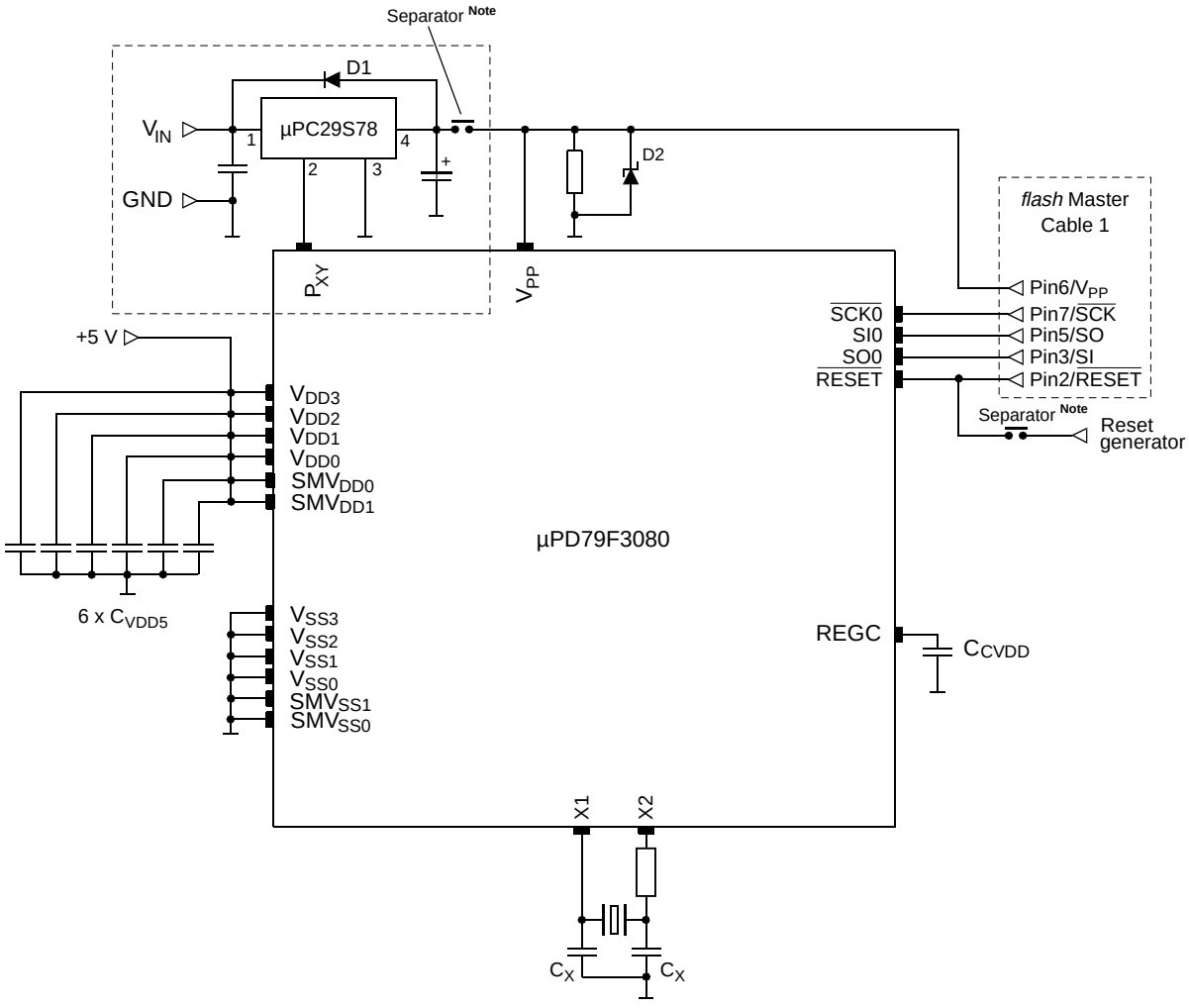


17.5 Flash Programming Circuitry

The following schematic shows the minimal circuitry. The circuitry incorporates a low-dropout voltage regulator (µPC29S78) as well as flash writer support. If the device is not used for Selfprogramming the V_{PP} pin have to be connected via a pull down resistor of 10 K to ground and the voltage regulator (µPC29S78) can be removed.
As the dedicated flash writer the *flash*MASTER is used. A further flash writer that can be used is the PG-FP3.

Remark: For unused pins the recommended connection of the “Table for recommended connection for unused pins Chapter 2” has to be used.

Figure 17-5: Minimal Circuitry for Flash Selfprogramming via CSI00



Note: The separators (e.g. jumpers) make possible that the restrictions of an external flash programmer (e.g. capacitive load of the selfprogramming voltage regulator) can be fulfilled. See “Pin Handling” on page 486.

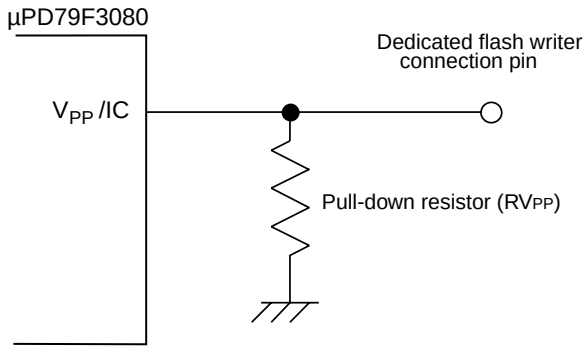
17.6 Pin Handling

When performing on-board writing, all required signals on the target system have to be made accessible to the dedicated flash writer. Also, it has to be ensured that the modes are set correctly and the V_{PP} signal, which is required to enter the programming mode can be controlled by the flash writer. In flash memory programming mode, all pins not required for the flash memory programming, remain in the same status as immediately after reset.

17.6.1 V_{PP} pin

In the normal operation mode, 0 V is input to V_{PP} pin. In the flash memory programming mode, 7.8 V writing voltage is supplied to V_{PP} pin. The following figure shows an example of the connection of the V_{PP} pin.

Figure 17-6: Pin Handling of V_{PP} pin



Remark: As pull-down resistor for using flashMASTER 10K are recommended.

17.6.2 Serial interface pins

The following shows the pins used by the serial interface.

Table 17-1: Serial interface pins

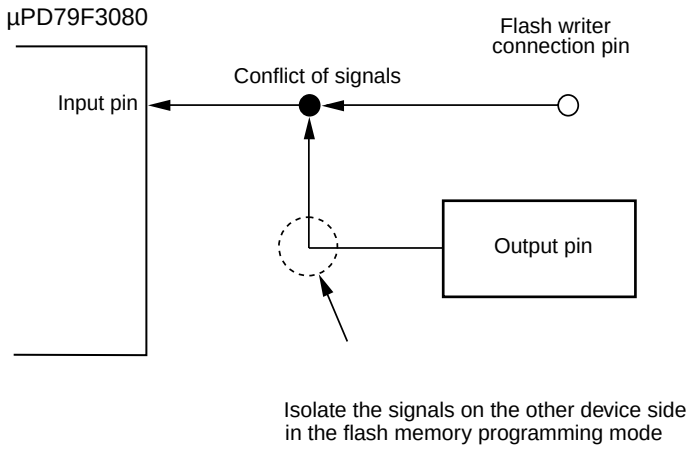
Serial Interface	Pins Used
CSI00	SO00, SI00, $\overline{\text{SCK00}}$
CSI00 with HW-Handshake	SO00, SI00, $\overline{\text{SCK00}}$, P87
UART50	TXD50, RXD50

When connecting a dedicated flash writer to a serial interface pin, which is connected to other devices on-board, care should be taken to avoid the conflict of signals and the malfunction of other devices, etc.

(1) Conflict of signals

When connecting a flash writer (output) to a serial interface pin (input) which is connected to another device (output), conflict of signals may happen. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the high-impedance status.

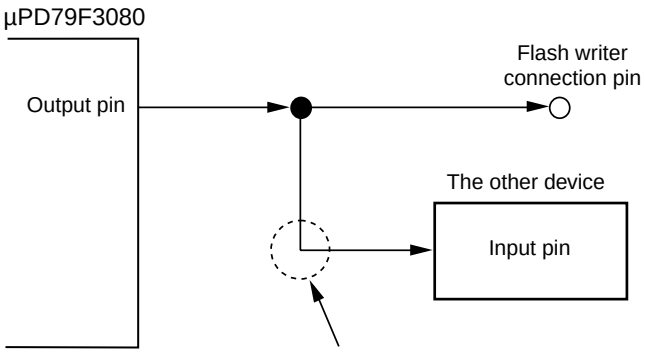
Figure 17-7: Conflict between Flash Writer and Other Output Pin



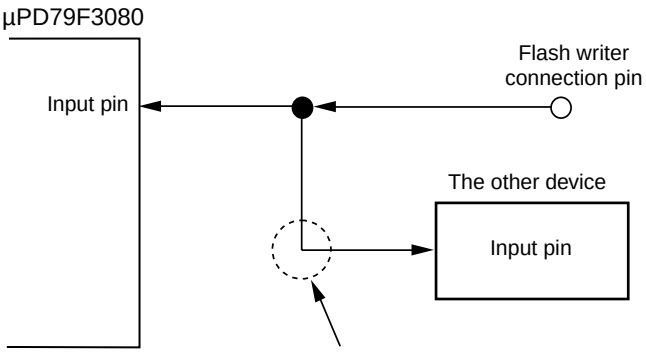
(2) Malfunction of the other device

When connecting a flash writer (output or input) to a serial interface pin (input or output) connected to another device (input), the signal output to the other device may cause the device to malfunction. To avoid this, isolate the connection to the other device or make the setting so that the input signal to the other device is ignored.

Figure 17-8: Malfunction of Other Input Pins



Isolate the signal on the other device input side in flash memory programming mode in case the µPD79F3080 output signal affects the other device input

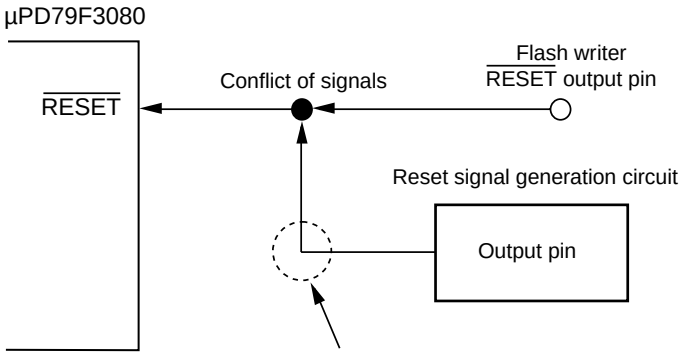


Isolate the signal on the other device input side in flash memory programming mode in case the flash writer output signal affects the other device input

17.6.3 $\overline{\text{RESET}}$ pin

When connecting the reset signals of the dedicated flash writer to the $\overline{\text{RESET}}$ pin which is connected to the reset signal generation circuit on-board, conflict of signals may happen. To avoid the conflict of signals, isolate the connection to the reset signal generation circuit.
When reset signal is input from the user system during the flash memory programming mode, programming operation will not be performed correctly. Therefore, do not input signals other than the reset signals from the dedicated flash writer.

Figure 17-9: Conflict between Flash Writer Reset Line and Reset Signal Generation Circuit



In the flash memory programming mode, the signal that the reset signal generation circuit outputs conflicts with the signal that the flash writer outputs. Therefore, isolate the signals on the reset signal generation circuit side.

17.6.4 NMI pin

Do not change the input signal to the NMI pin during the flash memory programming mode. If the NMI pin is changed during the flash memory programming mode, the programming may not be performed correctly.

17.6.5 Flash memory programming mode

To switch to the flash memory programming mode, apply writing voltage to V_{PP} pin, and release the reset.

17.6.6 Port pins

When the flash memory programming mode is set, all the port pins except the pins which communicate with the dedicated flash writer become high-impedance status. The treatment of these port pins is not necessary.

17.6.7 Other signal pins

Connect X1, X2 and AV_{REF} to the same status as that in the normal operation mode.

17.6.8 Power supply

Provide the same power supply (V_{DD0} to V_{DD3}, V_{SS0} to V_{SS3}, SMV_{DD0}, SMV_{DD1}, SMV_{SS0}, SMV_{SS1}, AV_{DD}, AV_{SS}, V_{LCD}) as that in normal operation mode.

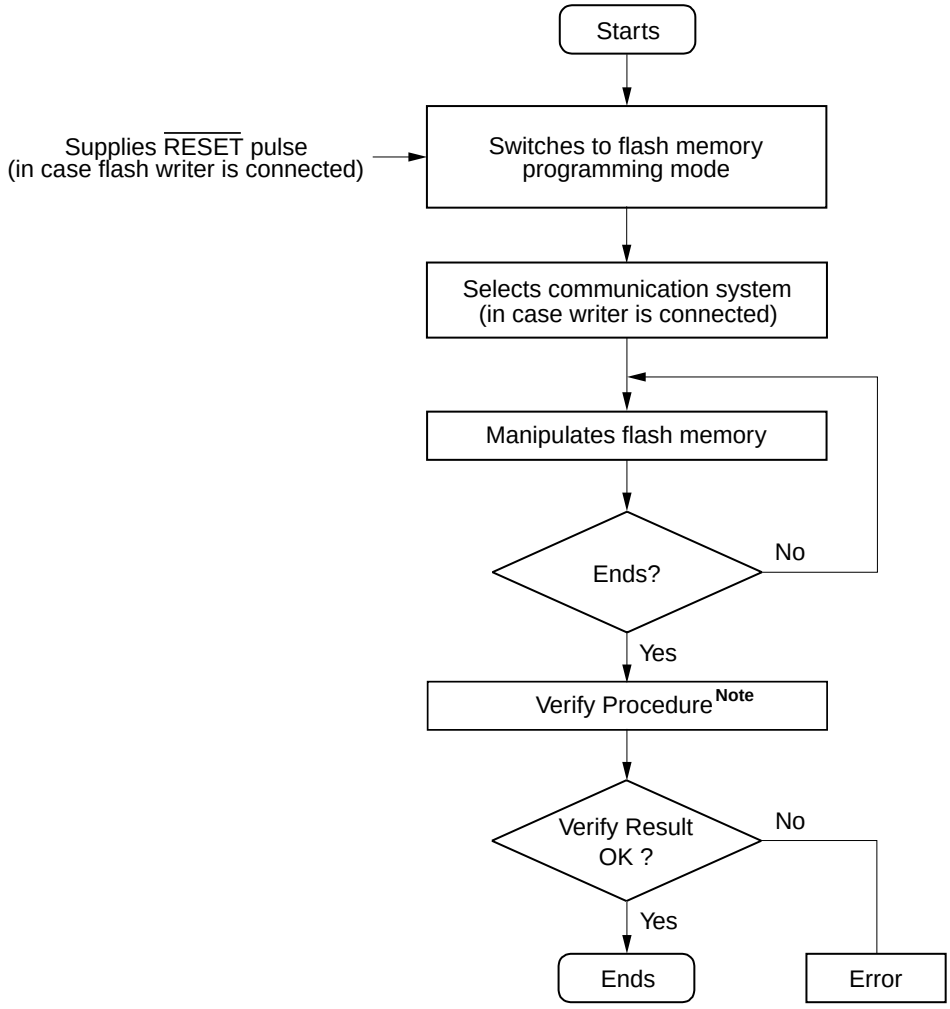
17.7 Programming Method

17.7.1 Flash memory control

To manipulate the flash memory the µPD70F3080 has to operate in a special flash memory programming mode. This mode can be entered either by applying the programming voltage of 7.8 V to the V_{PP} before the reset is release or by entering the Selfprogramming mode.

The following figure shows the procedure for manipulating the flash memory.

Figure 17-10: Flow Chart of Flash Memory Manipulation



Note: For further details please refer to the following document:

- Application Note - Selfprogramming
  32-/16-bit Single-Chip Microcontroller -- Selfprogramming Library
  (Doc.No.: U15352EE2V0AN00)

17.7.2 Selection of communication mode

In the µPD70F3080 as well as for other V850 family devices, a communication system is selected by inputting pulses (16 pulses maximal) to V_{PP} pin after switching to the flash memory programming mode. The V_{PP} pulses are generated by the dedicated flash writer.
The following table shows the relation between the number of pulses and the communication systems.

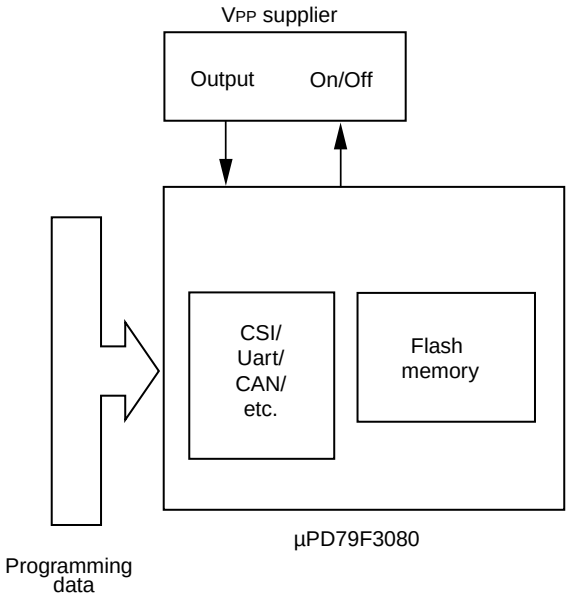
Table 17-2: List of Communication Systems

V_{PP} pulse	Communication System	Remarks
0	CSI00	µPD70F3080 performs slave operation, MSB first
3	CSI00 Handshake mode	P87 is used for HW-Handshake
8	UART50 Communication	rate: 9600 bps (after reset), LSB first
Others	(reserved)	Setting prohibited

17.8 Selfprogramming Mode

The flash Selfprogramming feature allows user to reprogram the flash contents by a user application program, without the necessity of an external flash writer.
This feature allows an update of the application with only on-board resources and a user defined communication interface.

Figure 17-11: Configuration in Selfprogramming Mode



In order to operate flash Selfprogramming, flash Selfprogramming libraries are prepared for user.

Following operations to the flash memory are supported by libraries.

- Initialize
- Blank Check
- Erase
- Write
- Verify
- Blank check
- V_{PP} Voltage Check
- Create Signature
- Check Signature
- Swap Area
- Check Area

For further details please refer to the following document:

- Application Note - Selfprogramming
  32-/16-bit Single-Chip Microcontroller -- Selfprogramming Library
  (Doc.No.: U15352EE2V0AN00)

17.9 Secure Selfprogramming

17.9.1 General description

A flash memory area can only be erased as a whole. If parts of the lower flash area have to be updated, the complete flash has to be erased. This bears the risk that a problem during Selfprogramming, particularly a power failure, leaves the device without any valid program for start-up.
To overcome those limitations µPD70F3080 features a method which is called “secure Selfprogramming”. By using “secure Selfprogramming”, it is always ensured that a valid boot program is available in the flash memory. This is achieved by enabling the user to select which of the two flash areas is mapped at address 0 and therefore accessed after reset, thus ensuring that the boot program located in this area is executed.

This selection is done by creating a signature at address 0000 0030H or 0001 0030H, depending on which area should become the one located at address 0H. Directly after reset, the device determines which area contains a valid signature and maps this area to address 0H.

17.9.2 Signature structure

The library provides a function to create a signature in either one of the two areas. It is located within the user address space of the flash memory. The signature structure was chosen in a way to ensure that no user data can be mistakenly interpreted as a signature. This is achieved by a different usage of internal structures of the flash memory.

17.9.3 Secure selfprogramming flow

A reprogramming of the flash memory starts with an erase of the upper area. After a successful erase, the boot program that is located in the lower area has to be copied (modification possible) to the upper area. Afterwards a signature is created in the upper area, indicating that this area contains a valid boot program. The signature which is found in the lower area is destroyed by writing a 00000000H to this address, and the areas are swapped, ensuring that the copied boot program is now located at address 0. This is followed by an erase of the area, which became the upper one still containing the old boot program and an invalid signature. After completion of the erase operation, the flash memory contains now only the boot program, so that the new application program can be written.

The flow looks as follows:

- Erase upper area
- Copy boot program from lower area to upper area
- Create signature in upper area
- Kill signature in lower area
- Swap area so that the lower area becomes the upper and vice versa
- Erase the upper area (which was formerly the lower)
- Write new application program to the flash memory
- After the last writing perform an internal verify to check the correct flash programming procedure

Figure 17-12: Secure Selfprogramming Flow (1/2)

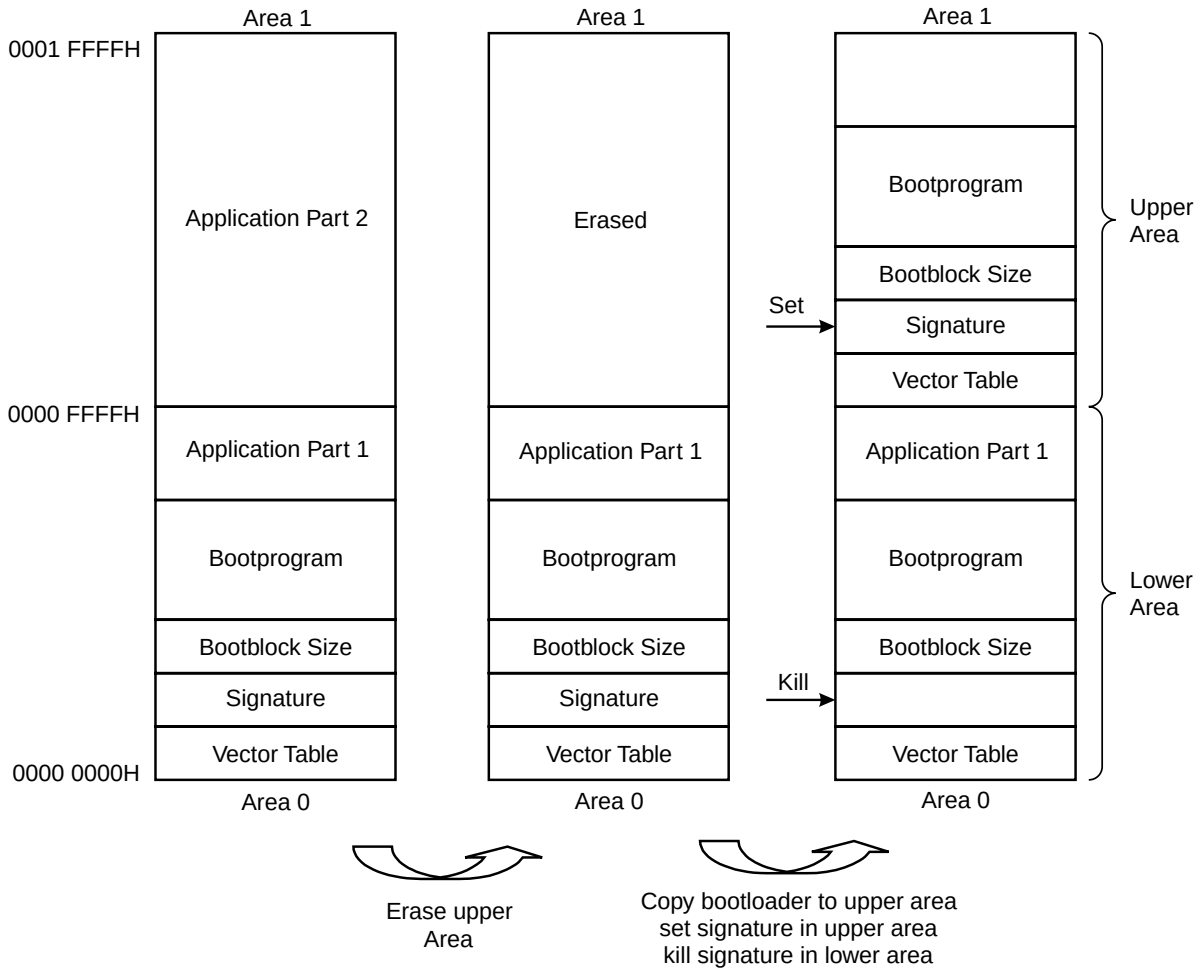
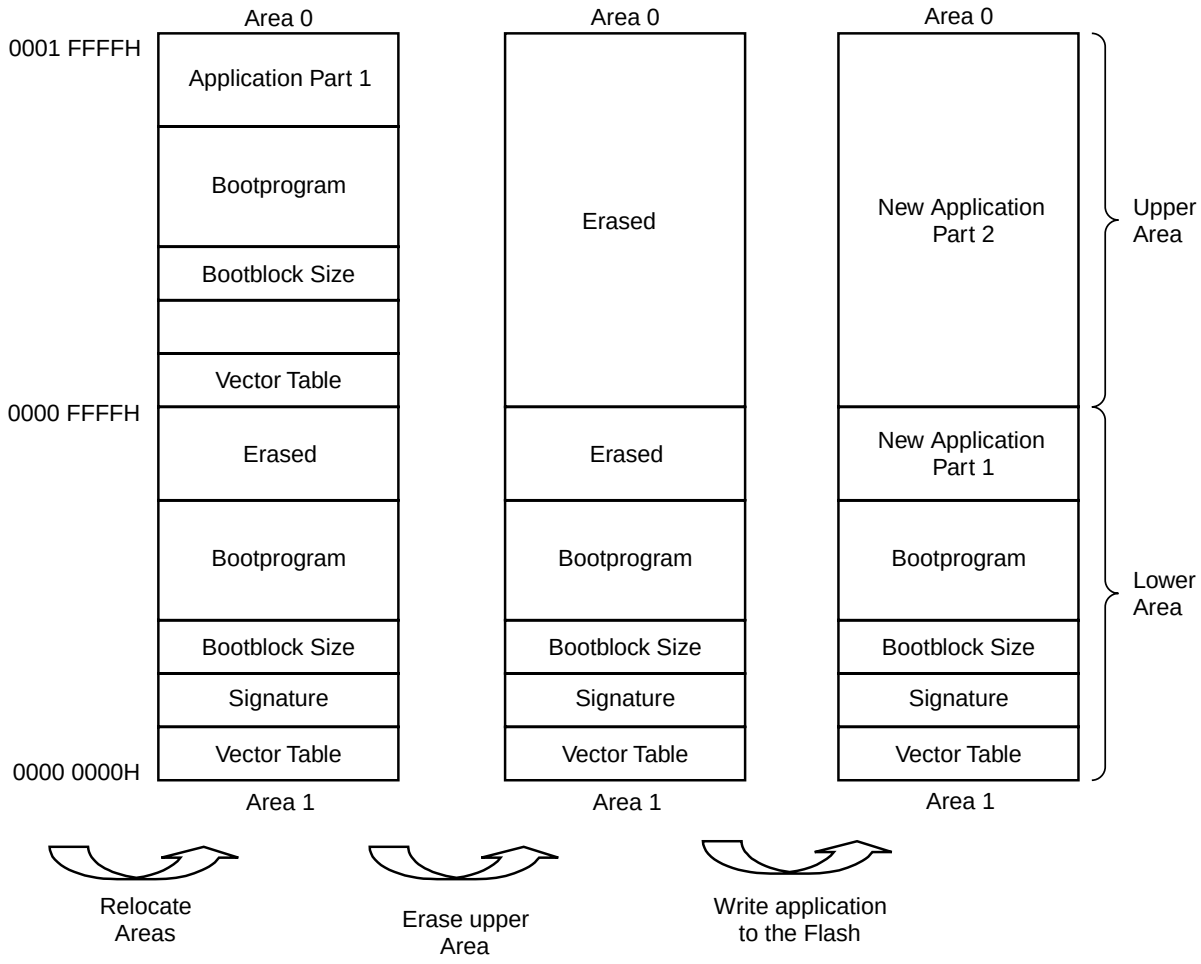


Figure 17-12: Secure Selfprogramming Flow (2/2)



17.9.4 Advantages of Secure Selfprogramming

- A boot program is always available, thus ensuring that the device can always be reprogrammed.
- The size of the boot block is not fixed. All sizes up to 64 K are supported.
- It is possible to update the boot program using the secure mechanisms.

[MEMO]

Appendix A List of Instruction Sets

Figure A-1: How to Read Instruction Set List

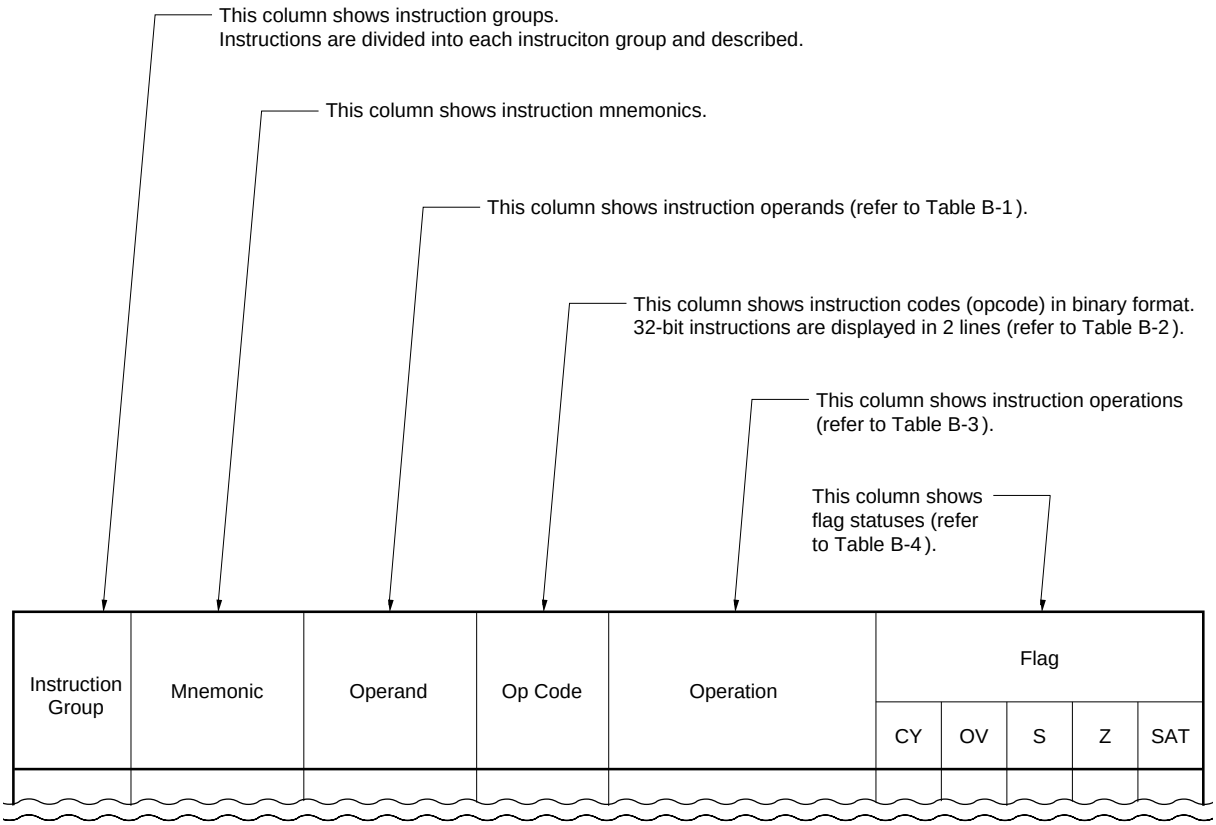


Table A-1: Symbols in Operand Description

Symbol	Description
reg1	General register (r0 to r31): Used as source register
reg2	General register (r0 to r31): Mainly used as destination register
ep	Element pointer (r30)
bit#3	3-bit data for bit number specification
imm×	×-bit immediate data
disp×	×-bit displacement
regID	System register number
vector	5-bit data that specifies trap vector number (00H to 1FH)
cccc	4-bit data that indicates condition code

Table A-2: Symbols Used for Op Code

Symbol	Description
R	1-bit data of code that specifies reg1 or regID
r	1-bit data of code that specifies reg2
d	1-bit data of displacement
i	1-bit data of immediate data
cccc	4-bit data that indicates condition code
bbb	3-bit data that specifies bit number

Table A-3: Symbols Used for Operation Description

Symbol	Description
←	Assignment
GR[ ]	General register
SR[ ]	System register
zero-extend (n)	Zero-extends n to word length.
sign-extend (n)	Sign-extends n to word length.
load-memory (a,b)	Reads data of size b from address a.
store-memory (a,b,c)	Writes data b of size c to address a.
load-memory-bit (a,b)	Reads bit b from address a.
store-memory-bit (a,b,c)	Writes c to bit b of address a
saturated (n)	Performs saturated processing of n. (n is 2ís complements). Result of calculation of n: If n is n ≥ 7FFFFFFFH as result of calculation, 7FFFFFFFH. If n is n ≤ 80000000H as result of calculation, 80000000H.
result	Reflects result to a flag.
Byte	Byte (8 bits)
Halfword	Half-word (16 bits)
Word	Word (32 bits)
+	Add
-	Subtract
||	Bit concatenation
×	Multiply
÷	Divide
AND	Logical product
OR	Logical sum
XOR	Exclusive logical sum
NOT	Logical negate
logically shift left by	Logical left shift
logically shift right by	Logical right shift
arithmetically shift right by	Arithmetic right shift

Table A-4: Symbols Used for Flag Operation

Symbol	Description
(blank)	Not affected
0	Cleared to 0
×	Set of cleared according to result
R	Previously saved value is restored

Table A-5: Condition Codes

Condition Name (cond)	Condition Code (cccc)	Conditional Expression	Description
V	0000	OV = 1	Overflow
NV	1000	OV = 0	No overflow
C/L	0001	CY = 1	Carry Lower (Less than)
NC/NL	1001	CY = 0	No carry No lower (Greater than or equal)
Z/E	0010	Z = 1	Zero Equal
NZ/NE	1010	Z = 0	Not zero Not equal
NH	0011	(CY OR Z) = 1	Not higher (Less than or equal)
H	1011	(CY OR Z) = 0	Higher (Greater than)
N	0100	S = 1	Negative
P	1100	S = 0	Positive
T	0101	-	Always (unconditional)
SA	1101	SAT = 1	Saturated
LT	0110	(S XOR OV) = 1	Less than signed
GE	1110	(S XOR OV) = 0	Greater than or equal signed
LE	0111	((S XOR OV) OR Z) = 1	Less than or equal signed
GT	1111	((S XOR OV) OR Z) = 0	Greater than signed

Table A-6: Instruction Set List (1/7)

Instruction Group	Mne-monic	Operand	Opcode	Operation	Flag				
					CY	OV	S	Z	SAT
Load/store	SLD.B	disp7 [ep], reg2	rrrrr0110 ddddddd	adr ← ep + zero-extend (disp7) GR [reg2] ← sign-extend (Load-memory (adr, Byte))					
	SLD.H	disp8 [ep], reg2	rrrrr1000 ddddddd **Note 1**	adr ← ep + zero-extend (disp8) GR [reg2] ← sign-extend (Load-memory (adr, Halfword))					
	SLD.W	disp8 [ep], reg2	rrrrr1010 dddddd0 **Note 2**	adr ← ep + zero-extend (disp8) GR [reg2] ← Load-memory (adr, Word)					
	LD.B	disp16 [reg1], reg2	rrrrr111000 RRRRR ddddddddd ddddddd	adr ← GR [reg1] + sign-extend (disp16) GR [reg2] ← sign-extend (Load-memory (adr, Byte))					
	LD.H	disp16 [reg1], reg2	rrrrr1110 01RRRRR ddddddddd dddddd0 **Note 3**	adr ← GR [reg1] + sign-extend (disp16) GR [reg2] ← sign-extend (Load-memory (adr, Halfword))					
	LD.W	disp16 [reg1], reg2	rrrrr1110 01RRRRR ddddddddd dddddd1 **Note 3**	adr ← GR [reg1] + sign-extend (disp16) GR [reg2] ← Load-memory (adr, Word))					
	SST.B	reg2, disp7 [ep]	rrrrr0111 ddddddd	adr ← ep + zero-extend (disp7) Store-memory (adr, GR [reg2], Byte)					
	SST.H	reg2, disp8 [ep]	rrrrr1001 ddddddd **Note 1**	adr ← ep + zero-extend (disp8) Store-memory (adr, GR [reg2], Halfword)					
	SST.W	reg2, disp8 [ep]	rrrrr1010 dddddd1 **Note 2**	adr ← ep + zero-extend (disp8) Store-memory (adr, GR [reg2], Word)					
	ST.B	reg2, disp16 [reg1]	rrrrr1110 10RRRRR ddddddddd ddddddd	adr ← GR [reg1] + sign-extend (disp16) Store-memory (adr, GR [reg2], Byte)					

Notes: 1. ddddddd is the higher 7 bits of disp8.

2. dddddd is the higher 6 bits of disp8.

3. ddddddddddddddd is the higher 15 bits of disp16.

4. Only the lower half-word data is valid.

5. ddddddddddddddddddddd is the higher 21 bits of dip22.

6. ddddddddd is the higher 8 bits of disp9.

7. The op code of this instruction uses the field of reg1 through the source register is shown as reg2 in the above table. Therefore, the meaning of register specification for mnemonic description and op code is different from that of the other instructions
rrr = regID specification
RRRRR = reg2 specification

Table A-6: Instruction Set List (2/7)

Instruction Group	Mnemonic	Operand	Opcode	Operation	Flag				
					CY	OV	S	Z	SAT
Load/store	ST.H	reg2, disp16 [reg1]	rrrrr1110 11RRRRR ddddddddd dddddd0 **Note 3**	adr ← GR [reg1] + sign-extend (disp16) Store-memory (adr, GR [reg2], Halfword)					
	ST.W	reg2, disp16 [reg1]	rrrrr1110 11RRRRR ddddddddd dddddd1 **Note 3**	adr ← GR [reg1] + sign-extend (disp16) Store-memory (adr, GR [reg2], Word)					
Arithmetic operation	MOV	reg1, reg2	rrrrr0000 00RRRRR	GR [reg2] ← GR [reg1]					
	MOV	imm5, reg2	rrrrr0100 00iiiii	GR [reg2] ← sign-extend (imm5)					
	MOVHI	imm16, reg1, reg2	rrrrr1100 10RRRRR iiiiiiii-iiiiiiii	GR [reg2] ← GR [reg1] + (imm16 || 0^{16})					
	MOVEA	imm16, reg1, reg2	rrrrr1100 01RRRRR iiiiiiii-iiiiiiii	GR [reg2] ← GR [reg1] + sign-extend (imm16)					
	LD.H	disp16 [reg1], reg2	rrrrr1110 01RRRRR ddddddddd dddddd0 **Note 3**	adr ← GR [reg1] + sign-extend (disp16) GR [reg2] ← sign-extend (Load-memory (adr, Halfword))					
	LD.W	disp16 [reg1], reg2	rrrrr1110 01RRRRR ddddddddd dddddd1 **Note 3**	adr ← GR [reg1] + sign-extend (disp16) GR [reg2] ← Load-memory (adr, Word))					
	SST.B	reg2, disp7 [ep]	rrrrr0111 ddddddd	adr ← ep + zero-extend (disp7) Store-memory (adr, GR [reg2], Byte)					
	ADD	reg1, reg2	rrrrr001110 RRRRR	GR [reg2] ← GR [reg2] + GR [reg1]					
	ADD	imm5, reg2	rrrrr010010i iiii	GR [reg2] ← GR [reg2] + sign-extend (imm5)	×	×	×	×	

Notes: 1. ddddddd is the higher 7 bits of disp8.

2. dddddd is the higher 6 bits of disp8.

3. ddddddddddddddd is the higher 15 bits of disp16.

4. Only the lower half-word data is valid.

5. ddddddddddddddddddddd is the higher 21 bits of dip22.

6. ddddddddd is the higher 8 bits of disp9.

7. The op code of this instruction uses the field of reg1 through the source register is shown as reg2 in the above table. Therefore, the meaning of register specification for mnemonic description and op code is different from that of the other instructions
rrr = regID specification
RRRRR = reg2 specification

Table A-6: Instruction Set List (3/7)

Instruction Group	Mnemonic	Operand	Opcode	Operation	Flag				
					CY	OV	S	Z	SAT
Arithmetic operation	ADDI	imm16, reg1, reg2	rrrrr110 000RRRRR iiiiiiii iiiiiiii	GR [reg2] ← GR [reg1] + sign-extend (imm16)	×	×	×	×	
	SUB	reg1, reg2	rrrrr001 101RRRRR	GR [reg2] ← GR [reg2] - GR [reg1]	×	×	×	×	
	SUBR	reg1, reg2	rrrrr001 100RRRRR	GR [reg2] ← GR [reg1] - GR [reg2]	×	×	×	×	
	MULH	reg1,reg2	rrrrr000 111RRRRR	GR [reg2] ← GR [reg2] **Note 4** × GR [reg1] **Note 4** (Signed multiplication)	×	×	×	×	
	MULH	imm5, reg2	rrrrr010 111iiiii	GR [reg2] ← GR [reg2] **Note 4** × sign-extend (imm5) (Signed multiplication)					
	MULHI	imm16, reg1, reg2	rrrrr110 111RRRRR iiiiiiii iiiiiiii	GR [reg2] ← GR [reg1] **Note 4** × imm16 (signed multiplication)					
	DIVH	reg1, reg2	rrrrr000 010RRRRR	GR [reg2] ← GR [reg2] ÷ GR [reg2] **Note 4** (Signed division)					
	CMP	reg1, reg2	rrrrr001 111RRRRR	result ← GR [reg2] - GR [reg1]		×	×	×	
	CMP	imm5, reg2	rrrrr010 011iiiii	result ← GR [reg2] - sign-extend (imm5)	×	×	×	×	
	SETF	cccc, reg2	rrrrr111 1110cccc 00000000 00000000	if conditions are satisfied then GR [reg2] ← 00000001H else GR [reg2] ← 00000000H	×	×	×	×	
Saturated operation	SAT-ADD	reg1, reg2	rrrrr000 110RRRRR	GR [reg2] ← saturated (GR [reg2] + GR [reg1])					
	SAT-ADD	imm5, reg2	rrrrr010 001iiiii	GR [reg2] ← saturated (GR [reg2] + sign-extend (imm5))	×	×	×	×	×
	SAT-SUB	reg1, reg2	rrrrr000 101RRRRR	GR [reg2] ← saturated (GR [reg2] - GR [reg1])	×	×	×	×	×

Notes: 1. ddddddd is the higher 7 bits of disp8.

2. dddddd is the higher 6 bits of disp8.

3. dddddddddddddddd is the higher 15 bits of disp16.

4. Only the lower half-word data is valid.

5. ddddddddddddddddddddd is the higher 21 bits of dip22.

6. dddddddd is the higher 8 bits of disp9.

7. The op code of this instruction uses the field of reg1 through the source register is shown as reg2 in the above table. Therefore, the meaning of register specification for mnemonic description and op code is different from that of the other instructions
rrr = regID specification
RRRRR = reg2 specification

Table A-6: Instruction Set List (4/7)

Instruction Group	Mnemonic	Operand	Opcode	Operation	Flag				
					CY	OV	S	Z	SAT
Saturated operation	SATSUBI	imm16, reg1, reg2	rrrrr110011RRRRR iiiiiiii iiiiiiii	GR [reg2] ← saturated (GR [reg1] - sign-extend (imm16))	×	×	×	×	×
	SATSUBR	reg1, reg2	rrrrr000100RRRRR	GR [reg2] ← saturated (GR [reg1] - GR [reg2])	×	×	×	×	×
Logic operation	TST	reg1, reg2	rrrrr001011RRRRR	result ← GR [reg2] AND GR [reg1]	×	×	×	×	×
	OR	reg1, reg2	rrrrr001000RRRRR	GR [reg2] ← GR [reg2] OR GR [reg1]		0	×	×	
	ORI	imm16, reg1, reg2	rrrrr110100RRRRR iiiiiiii iiiiiiii	GR [reg2] ← GR [reg1] OR zero-extend (imm16)		0	×	×	
	AND	reg1, reg2	rrrrr001010RRRRR	GR [reg2] ← GR [reg2] AND GR [reg1]		0	×	×	
	ANDI	imm16, reg1, reg2	rrrrr110110RRRRR iiiiiiii iiiiiiii	GR [reg2] ← GR [reg1] AND zero-extend (imm16)		0	×	×	
	XOR	reg1, reg2	rrrrr001001RRRRR	GR [reg2] ← GR [reg2] XOR GR [reg1]		0	×	×	
	XORI	imm16, reg1, reg2	rrrrr110101RRRRR iiiiiiii- iiiiiiii	GR [reg2] ← GR [reg1] XOR zero-extend (imm16)		0	×	×	
	NOT	reg1, reg2	rrrrr000001RRRRR	GR [reg2] ← NOT (GR [reg1])		0	×	×	
	SHL	reg1, reg2	rrrrr111111RRRRR 000000001 1000000	GR [reg2] ← GR [reg2] logically shift left by GR [reg1])	×	0	×	×	
	SHL	imm5, reg2	rrrrr010110iiiii	GR [reg2] ← GR [reg2] logically shift left by zero-extend (imm5)	×	0	×	×	

Notes: 1. ddddddd is the higher 7 bits of disp8.

2. dddddd is the higher 6 bits of disp8.

3. ddddddddddddddd is the higher 15 bits of disp16.

4. Only the lower half-word data is valid.

5. ddddddddddddddddddddd is the higher 21 bits of dip22.

6. ddddddddd is the higher 8 bits of disp9.

7. The op code of this instruction uses the field of reg1 through the source register is shown as reg2 in the above table. Therefore, the meaning of register specification for mnemonic description and op code is different from that of the other instructions
rrr = regID specification
RRRRR = reg2 specification

Table A-6: Instruction Set List (5/7)

Instruction Group	Mnemonic	Operand	Opcode	Operation	Flag				
					CY	OV	S	Z	SAT
Logic operation	SHR	reg1, reg2	rrrrr1111 111cccc 000000001 0000000	GR [reg2] ← GR [reg2] logically shift right by GR [reg1]	×	0	×	×	
	SHR	imm5, reg2	rrrrr0101 00iiiii	GR [reg2] ← GR [reg2] logically shift right by zero-extend (imm5)	×	0	×	×	
	SAR	reg1, reg2	rrrrr1111 11RRRRR 000000001 0100000	GR [reg2] ← GR [reg2] arithmetically shift right by GR [reg1]	×	0	×	×	
	SAR	imm5, reg2	rrrrr0101 01iiiii	GR [reg2] ← GR [reg2] arithmetically shift right by zero-extend (imm5)	×	0	×	×	
Jump	JMP	[reg1]	000000000 11RRRRR	PC ← GR [reg1]					
	JR	disp22	000001111 0dddddd dddddddddd dddddd0 **Note 5**	PC ← PC + sign-extend (disp22)					
	JARL	disp22, reg2	rrrrr1111 0dddddd dddddddddd dddddd0 **Note 5**	GR [reg2] ← PC + 4 PC ← PC + sign-extend (disp22)					
	Bcond	disp9	ddddd1011 dddcccc **Note 6**	if conditions are satisfied then PC ← PC + sign-extend (disp9)					
Bit manipulate	SET1	bit#3, disp16 [reg1]	00bbb1111 10RRRRR ddddddddd ddddddd	adr ← GR [reg1] + sign-extend (disp16) Z flag ← Not (Load-memory-bit (adr, bit#3) Store memory-bit (adr, bit#3, 1)				×	
	CLR1	bit#3, disp16 [reg1]	10bbb1111 10RRRRR ddddddddd ddddddd	adr ← GR [reg1] + sign-extend (disp16) Z flag ← Not (Load-memory-bit (adr, bit#3)) Store memory-bit (adr, bit#3, 0)				×	

Notes: 1. ddddddd is the higher 7 bits of disp8.

2. dddddd is the higher 6 bits of disp8.

3. ddddddddddddddd is the higher 15 bits of disp16.

4. Only the lower half-word data is valid.

5. dddddddddddddddddddddd is the higher 21 bits of dip22.

6. dddddddd is the higher 8 bits of disp9.

7. The op code of this instruction uses the field of reg1 through the source register is shown as reg2 in the above table. Therefore, the meaning of register specification for mnemonic description and op code is different from that of the other instructions
rrr = regID specification
RRRRR = reg2 specification

Table A-6: Instruction Set List (6/7)

Instruction Group	Mnemonic	Operand	Opcode	Operation	Flag				
					CY	OV	S	Z	SAT
Bit manipulate	NOT1	bit#3, disp16 [reg1]	01bbb1111 10RRRRR ddddddddd ddddddd	adr ← GR [reg1] + sign-extend (disp16) Z flag ← Not (Load-memory-bit (adr, bit#3)) Store-memory-bit (adr, bit#3, Z flag)				×	
	TST1	bit#3, disp16 [reg1]	11bbb1111 10RRRRR ddddddddd ddddddd	adr ← GR [reg1] + sign-extend (disp16) Z flag ← Not (Load-memory-bit (adr, bit#3))				×	
Special	LDSR	reg2, regID	rrrrr1111 11RRRRR 000000000 0100000 **Note 7**	SR [regID] ←GR [reg2]					
				regID = EIPSW, FEPSW					
				regID = PSW	×	×	×	×	×
	STSR	regID, reg2	rrrrr1111 11RRRRR 000000000 1000000	GR [reg2] ← SR [regID]					
	TRAP	vector	000001111 11iiiii 000000010 0000000	EIPC ← PC + 4 (Restored PC) EIPSW ← PSW ECR.EICC ← Interrupt code PSW.EP ← 1 PSW.ID ← 1 PC ← 00000040H (vector = 00H to 0FH) 00000050H (vector = 10H to 1FH)					
	RETI		000001111 1100000 000000010 1000000	if PSW.EP = 1 then PC ← EIPC PSW ← EIPSW else if PSW.NP = 1 then PC ← FEPC PSW ← FEPSW else PC ← EIPC PSW ← EIPSW	R	R	R	R	R
	HALT		000001111 1100000 000000010 0100000	Stops					

Notes: 1. ddddddd is the higher 7 bits of disp8.

2. dddddd is the higher 6 bits of disp8.

3. ddddddddddddddd is the higher 15 bits of disp16.

4. Only the lower half-word data is valid.

5. dddddddddddddddddddddd is the higher 21 bits of dip22.

6. ddddddddd is the higher 8 bits of disp9.

7. The op code of this instruction uses the field of reg1 through the source register is shown as reg2 in the above table. Therefore, the meaning of register specification for mnemonic description and op code is different from that of the other instructions
rrr = regID specification
RRRRR = reg2 specification

Table A-6: Instruction Set List (7/7)

Instruction Group	Mnemonic	Operand	Opcode	Operation	Flag				
					CY	OV	S	Z	SAT
	DI		000001111 1100000 000000010 1100000	PSW.ID ← 1 (Maskable interrupt disabled)					
	EI		100001111 1100000 000000010 1100000	PSW.ID ← 0 (Maskable interrupt enabled)					
	NOP		000000000 0000000	Uses 1 clock cycle without doing anything					

Notes: 1. ddddddd is the higher 7 bits of disp8.

2. dddddd is the higher 6 bits of disp8.

3. ddddddddddddddd is the higher 15 bits of disp16.

4. Only the lower half-word data is valid.

5. ddddddddddddddddddddd is the higher 21 bits of dip22.

6. ddddddddd is the higher 8 bits of disp9.

7. The op code of this instruction uses the field of reg1 through the source register is shown as reg2 in the above table. Therefore, the meaning of register specification for mnemonic description and op code is different from that of the other instructions
rrr = regID specification
RRRRR = reg2 specification

[MEMO]

Appendix B Index

W

X

Appendix C Revision History

The following shows the revision history.

Table C-1: List of Revisions (1/4)

Edition No.	Pages	Chapter	Main revised contents from old edition	
			where	what
EE2	cover			added µPD703081 (Mask Version)
		Preface		added chapter Preface
	22	01-Introduction	1.2 Device Features • CPU • Internal memory • Power supply voltage range • internal BCU	• changed 65 ns value to 62.5 ns • added µPD703081 (Mask Version) internal memory table • changed 4.5 V value to 4.0 V • deleted internal BCU
	23		1.4 Ordering Information	added values for µPD703081 (Mask Version)
	24		Figure 1-1	added Notes
	25		Pin Identification	added pin functions
	26		Figure 1-2	added notes for 703081 (Mask Version)
	27		1.6.2 Pin-chip units	added Timer, WT,WDT, MTCD, LCD
	54	02- Pin Functions	Table 2-4	added "has to be used" pin remarks (especial SMV_{DDn}, SMV_{SSn})
	55			added Remark: Please aware, that the port pins are in input mode after RESET release
	70	03- CPU Functions	Figure 3-9: Memory Map	corrected DCAN (Ch0) Base address (xx3F FB00H -> xx3F FE00H)
	71		Figure 3-10: Internal ROM/Flash Memory Area	corrected addresses at the begin of the image area
	72		(3) Internal RAM area	improved explanation
	76		Figure 3-14: Recommended Memory Map (Flash Memory Version)	improved
	94		Table 3-6: Non-Peripheral I/O Registers for DCAN0 (1/10)	separated tables of the 2 DCAN channels
	109	04- Interrupt/ Exception Processing Function	4.1 Features • Interrupts	added RESET interrupt correction of NMI interrupt
	111		Table 4-1: Interrupt/Exception Sources	renamed "INTGOVF0" to “INTTMG0”
	138		4.7 Multiple Interrupt Processing Control	new explanation with new flow charts and new timings
	140		4.8 Interrupt Response Time	new explanation with new flow charts and new timings

Table C-1: List of Revisions (2/4)

Edition No.	Pages	Chapter	Main revised contents from old edition	
			where	what
EE2	167	06-Timer	(1) Timer G Mode Register High (TMGMH)	added Remark for POWER bit
			Figure 6-7: Timer G Mode Register High (TMGMH)	new cautions for OLDE bit, CSExn bits, CCSG1/0 bits
	169		(3) Timer G Channel Mode Register High and Low (TMGCMH, TMGCML)	new Caution for TMGCMH/L
	170		Figure 6-10: Timer G Output Control Register (OCTLGH, OCTLGL)	corrected Cautions for SWFGm and CCSGm bits
	173		6.1.6 Explanation of Basic Operation	added remark for PWM modes of Timer G outputs
	180		Figure 6-17: Timing when GCC1 is rewritten during operation (free run)	renamed "Clock" -> "f_{XX}"
	204		(2) Capture/compare registers (CCC0n and CCC1n)	deleted Caution for rewriting CCC0n and CCC1n registers
	212		(2) Overflow	deleted explanation for the OST bit
	221		6.2.6 Precautions Timer C	deleted Caution for rewriting CCC0n and CCC1n registers
	223		6.3.1 Timer 5 (TM50, TM51) Figure 6-48	changed pin symbol of f_{XX} from external pin to internal signal
	243	07- Watch Timer	7.1 Function	added of further explanation
	243		Figure 7-1: Block Diagram of Watch Timer	renamed “Clear” to “Start/Stop and Clear”
	243		7.1 Function (1) Watch Timer	changed cross reference to “Watch Timer Clock Generator”
	245		7.3 Watch Timer Control Register (1) Watch Timer mode control register Figure 7-2	change explanation of WTM1, WTM0
	246		7.4 Watch Timer Clock Generator	added explanations to the Watch Timer Clock Generator
	248		7.5. Operation 7.5.1 Operation as watch timer	changed explanation for the Watch Timer function especial the WTM1, WTM0 bits
	290	09-Serial Interface Function	Figure 9-22: Clocked Serial Interface Mode Registers (CSIM00, CSIM01, CSIM02) (2/2)	changed explanation of CSIE bit 0: Enable CSIn operation. 1: Disable CSIn operation.
	309		(b) Usage (transmission/reception)	added explanation for the case of the repeat transfer mode

Table C-1: List of Revisions (3/4)

Edition No.	Pages	Chapter	Main revised contents from old edition	
			where	what
EE2		10-DCAN	10.11 Operation of DCAN Controller 10.11.2 CAN Control Register	added explanations for the SLEEP bit of CANCn
	318		10.2 CAN memory	added figure for DCAN memory map
	340		10.12.2 CAN Control Register (CANCn)	• better explanation of INIT bit function • added caution for SLEEP bit • better explanation for STOP bit
	342		Figure 10-17: DCAN Time Stamp Support	better explanation for time stamp
	344		Figure 10-21: Transmission/Reception Flag	moved to better position
	349		Figure 10-25: Message Count Register (MCNTn) (n = 0, 1)	removed position for TLRES bit function
	355		Limits on defining the bit timing	correction of the explanation for the timing
	357		Figure 10-29: Transmit Control Register (TCRn) (n = 0, 1) (1/3)	improved explanation for the TXCn bit
	362		Table 10-3: Mask Operation Buffers	added example
	364		Figure 10-32: Redefinition Control Register (2/2)	improved explanation of Dn flag
	368		10.16.4 DCAN Sleep Mode	improved explanation of the Sleep mode with Code example
	375		10.18.5 Data New Flag behaviour Figure 10-38: Receive with Interrupt in case of Data New Flag behaviour	corrected interrupt receiving
	376		10.18.5 Data New Flag behaviour Figure 10-39: Receive, Software Polling in case of Data New Flag Limitation	corrected software polling flow

Table C-1: List of Revisions (4/4)

Edition No.	Pages	Chapter	Main revised contents from old edition	
			where	what
EE2	380	11- A/D Converter	11.2 A/D Converter Configuration (7) AV_{DD} pin	changed explanation: AV_{DD} path is not shared with AV_{REF}
	380		11.2 A/D Converter Configuration (8) AV_{REF} pin	added explanation for AV_{REF} pin
	381		11.3 A/D Converter Control Register 11.3.1 Register format of A/D Converter Control Register Table 11-2 Register format of A/D Converter Control Register	deleted 8-bit access for ADCR register
	388		11.4.2 Input voltage and conversion results	corrected calculation
	391		Figure 11-10: A/D Conversion	corrected timing
	392		11.5 A/D Converter Precautions (1) Current consumption in standby mode	added caution for power save modes
	392		11.5 A/D Converter Precautions Figure 11-11	deleted AV_{DD} path (not shared with AV_{REF})
	393		11.5 A/D Converter Precautions Figure 11-12	changed "AV_{DD}" to "AV_{REF}"
	393		(6) AV_{REF} pin input impedance	corrected explanation of AV_{REF} pin
	394		11.5 A/D Converter Precautions Figure 11-13	corrected timing
	395 to 405	12-Meter Controller/ Driver	whole Chapter	changed "MCNTn" register tag to "TMCNTn"
	402		12.3 Meter Controller/Driver Control Register (2) Compare Control register (MCMPCn) Figure 12-5	changed Bit table DIR1, DIR0: SMn1 column: "1" to "0"
	407	13-LCD Controller/ Driver	Figure 13-1: LCD Controller/Driver Block Diagram	improved voltage pin circuitry
	408		Figure 13-2: LCD Display Mode Register Format	improved explanation for LIPS bit
	460 to 465	14-Port Functions	14.3.11 Port 10 14.3.12 Port 11	changed any "input/output" function to "output"
	475	15-RESET Function	Figure 15-1: Reset signal acknowledgment	corrected timing
	476		Figure 15-2: Reset at power-on	corrected timing
	485	17-Flash Memory	17.5 Flash Programming Circuitry Figure 17-5 Minimal Circuitry for Flash Selfprogramming via CSI00	• changed title • modified Figure: clearer separated selfprogramming circuit from the flashMASTER connections

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Market Communication Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6465-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: 02-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-6250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	☐	☐	☐	☐
Technical Accuracy	☐	☐	☐	☐
Organization	☐	☐	☐	☐